# Abstract and Concrete Model Checking

Ichiro Hasuo

National Institute of Informatics
SOKENDAI (The Graduate University for Advanced Studies)
Tokyo, Japan

`http://group-mmm.org/~ichiro`
`i.hasuo[at]acm.org`

Version: November 25, 2024
Draft—your comments are appreciated!

# Contents

## Reading Guide

## Prerequisites

## Acknowledgments

# Chapter 1

# Introduction

A lot to write here

## 1.1 Notations

The sets of natural numbers (by which we mean nonnegative integers in this book), positive integers, integers, rationals, are reals are denoted by $\mathbb{N}, \mathbb{Z}_{>0}, \mathbb{Z}, \mathbb{Q}$, and $\mathbb{R}$, respectively.

The class of all ordinals is denoted by **Ord**. For set-theoretic preliminaries on ordinals etc., see e.g. [7].

The application of a function $f\colon X \to Y$ to its argument $x \in X$ is denoted usually by $f(x)$, but sometimes by $fx$, omitting parentheses. The set of functions of the type $X \to Y$ is denoted by $Y^X$ and called the *function space* from $X$ to $Y$.

## 1.2 Examples I: Model Checking Transition Systems

We start with some examples that motivate our study of model checking in abstract lattice-theoretic terms.

### 1.2.1 Transition Systems

**Definition 1.2.1** (transition system)**.** A *transition system* is a pair $\mathcal{S} = (X, R)$ of a set $X$ of *states* and a binary relation $R \subseteq X \times X$ called a *transition relation*. Each pair $(x, x') \in R$ is called an *edge*, and we often write $x \to x'$ if there is such an edge. In this case, we say that $x'$ is a *successor* of $x$.

A (directed) *path* in $\mathcal{S}$ is a (finite or infinite) sequence $x_0x_1\dots$ such that $x_i \to x_{i+1}$ for each $i$. A path is also denoted by $x_0 \to x_1 \to \cdots$.

For a finite path $x_0x_1\dots x_n$, its *length* is $n+1$.

In the literature, the term *transition system* can be used as a general term and refer to a much wider class of mathematical models that have states and transitions. In this case, labeled transition systems, Markov chains, quantum

Markov decision processes, etc. are all examples of "transition systems." Instead, in this book, we use the term for the specific meaning of Definition 1.2.1.

**Remark 1.2.2.** Note that Definition 1.2.1 is nothing but the common notion of *(directed) graph*. We adopt the name with the model checking applications in mind.

**Example 1.2.3.** An example of a transition system is in **??**. It happens to be a finite graph ($X$ is a finite set), but it has infinite paths, such as $x^{(2)}x^{(3)}x^{(2)}x^{(3)}\dots$. Note that, in general, we do not restrict to finite graphs.

### 1.2.2 Model Checking Problems and Algorithms

We shall exhibit three typical examples of problems in model checking. We discuss algorithms too.

#### 1.2.2.1 What Is Model Checking?

In the literature, the term *model checking* refers to a problem of the following format.

**Input:**
- A *system model* $\mathcal{M}$, typically given by a state-based dynamical system such as a transition system (Definition 1.2.1) or some symbolic expression that denotes such a system, and
- a *specification* $\varphi$, i.e. a property desired of $\mathcal{M}$, typically given in some symbolic formalism. Typical examples are temporal logic formulas, automata, etc.

**Output:** Whether $\mathcal{M}$ satisfies $\varphi$ (often denoted by $\mathcal{M} \models \varphi$), or not. If it does, we would like a formal proof for the satisfaction; if it does not, a *counterexample*—a concrete evidence for violation, such as an execution trace of $\mathcal{M}$—is desired.

The "model checking" problems in this section look different from the above, and the reader may wonder if they are not too simple. They are so-called safety and reachability problems, and there is no symbolic formalism involved on the specification side.

We justify our study of simpler safety and reachability problems as follows.

Firstly, safety and reachability specifications are typical specifications of real-world significance. Many symbolic formalisms for specification accommodate those.

Secondly, algorithms for these problems serve as a foundation for more general model checking problems. In fact, model checking for more complicated specifications—such as recurrence ("reach $P$ infinitely often"), persistence ("reach $P$ and stay there"), and reach-avoid ("reach $P$ while avoiding $Q$")—gets reduced to checking safety and reachability, either in terms of algorithms or in terms of theoretical foundations.

#### 1.2.2.2 Problem I-I: Demonic Safety

Let us consider the following *demonic safety* problem:

**Input:**
- A transition system $\mathcal{S} = (X, R)$

**Algorithm 1** A "find the perpetually safe states" algorithm for demonic safety. When the $n$-th iteration starts, $Z$ is the set of states from which all paths of length $\leq n$ stay in $P$

$Y \leftarrow X$
$Z \leftarrow P$
**while** $Z \neq Y$ **do**
    $Y \leftarrow Z$
    $Z \leftarrow P \cap \{x \in X \mid \forall y \in X.\, (x \to y \text{ implies } y \in Y)\}$
**return** if $I \subseteq Y$

**Algorithm 2** A "find the states that can eventually go unsafe" algorithm for demonic safety. When the $n$-th iteration starts, $Z$ is the set of states from which there is a path to $X \setminus P$ of length $\leq n$

$Y \leftarrow \emptyset$
$Z \leftarrow X \setminus P$
**while** $Z \neq Y$ **do**
    $Y \leftarrow Z$
    $Z \leftarrow (X \setminus P) \cup \{x \in X \mid \exists y \in X.\, (x \to y \text{ and } y \in Y)\}$
**return** if $Y \subseteq X \setminus I$

- A set $I \subseteq X$ of *initial states*
- A set $P \subseteq X$ of *safe states*

**Output:** Whether the following holds or not:

for an arbitrary initial state $x \in I$, any path from $x$ stays in $P$ all the time.

We say "demonic" safety since we require safety for *any* path—we assume that the choice of successors is by the demonic environment and it will make choices in harm's way.

This problem can be solved by many algorithms. Some of them are shown in Algorithms 1 to 3. They are different algorithms—we expect that they have different performance on different transition problem instances. It is, however, also clear that they share a similar structure.

Later in Chapter 3 we present an abstract framework for model checking safety properties. There we will present a series of theoretical results that derive

**Algorithm 3** A "find all the states reachable from $I$" algorithm for demonic safety. When the $n$-th iteration starts, $Z$ is the set of states from which there is a path from some $x \in I$ of length $\leq n$

$Y \leftarrow \emptyset$
$Z \leftarrow I$
**while** $Z \neq Y$ **do**
    $Y \leftarrow Z$
    $Z \leftarrow I \cup \{y \in X \mid \exists x \in X.\, (x \to y \text{ and } x \in Y)\}$
**return** $Y \subseteq P$

**Algorithm 4** A "find the states that can reach $P$" algorithm for angelic reachability. When the $n$-th iteration starts, $Z$ is the set of states from which there is a path of length $\leq n$ that reaches $P$

$Y \leftarrow \emptyset$
$Z \leftarrow P$
**while** $Z \neq Y$ **do**
  $Y \leftarrow Z$
  $Z \leftarrow P \cup \{x \in X \mid \exists y \in X.\, (x \to y \text{ and } y \in Y)\}$
**return** if $I \subseteq Y$

**Algorithm 5** A "find the states that that never reaches $P$" algorithm for angelic reachability. When the $n$-th iteration starts, $Z$ is the set of states from which all paths of length $\leq n$ stay in $X \setminus P$

$Y \leftarrow X$
$Z \leftarrow X \setminus P$
**while** $Z \neq Y$ **do**
  $Y \leftarrow Z$
  $Z \leftarrow (X \setminus P) \cap \{x \in X \mid \forall y \in X.\, (x \to y \text{ implies } y \in Y)\}$
**return** if $Y \subseteq X \setminus I$

the variations in Algorithms 1 to 3.

#### 1.2.2.3 Problem I-II: Angelic Reachability

Now let us consider the following *angelic reachability* problem:

**Input:**
- A transition system $\mathcal{S} = (X, R)$
- A set $I \subseteq X$ of *initial states*
- A set $P \subseteq X$ of *target states*

**Output:** Whether the following holds or not:

> for an arbitrary initial state $x \in I$, there is a path from $x$ that reaches $P$ eventually.

We say "angelic" reachability since we need only one path to reach $P$. Intuitively, the choice of successors is made by us and we can steer the choices in a way we desire.

It is important to note that the angelic reachability problem is *not* the negation of demonic safety. Indeed many parts of the definitions look dual to their counterparts: "exists a path" vs. "for all paths," "stay in $P$" vs. "reach $P$ eventually," etc. This seeming duality breaks once we look at the treatment of initial states: in both problems, we require that *all* initial states $x \in I$ possess the desired property.

Much like for demonic safety (Section 1.2.2.2), there are a few conceivable algorithms for this problem. See Algorithms 4 and 5.

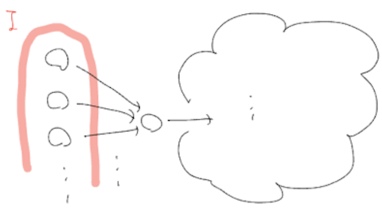
Another possible algorithm for angelic reachability is shown in Algorithm 6. It is different from the previous two (Algorithms 4 and 5) in that it operates in a forward manner: starting from an initial states, it enumerates all the reachable

**Algorithm 6** A "pointwise forward" algorithm for angelic reachability. It checks the reachability to $P$ for each $x_0 \in I$, and takes the conjunction.

```
b ← true
for all x_0 ∈ I do
    Y ← ∅
    Z ← {x_0}                                   ▷ reachable states discovered so far
    while Z ≠ Y do
        Y ← Z
        Z ← {x_0} ∪ {y ∈ X | ∃x ∈ X. (x → y and x ∈ Y)}
    b' ← whether Y ∩ P ≠ ∅
    b ← b ∧ b'
return b
```

states from it. In fact, Algorithm 6 may be the first algorithm many would come up with.

However, Algorithm 6 does not seem optimal. It seems to do many redundant tasks, for example in the following transition system.




By running the for-all loop, the task of analyzing the main part of the system (on the right) is repeated many times.

worst-case complexity analysis to show that this algorithm is indeed suboptimal

In the abstract framework we present later in Chapter 3, we will

– use the same theoretical results as we use for demonic safety (Section 1.2.2.2) to derive the variations of algorithms, and
– shed a theoretical light on why the pointwise algorithm (Algorithm 6) is suboptimal. We will observe that the pointwise algorithm is derived in a theoretically awkward manner.

Make a remark on "don't have to touch these guys any more" optimization.

## 1.3 Examples II: Model Checking Markov Chains

### 1.3.1 Markov Chains

Markov chains (MCs) are the simplest class of probabilistic systems. We restrict to those with discrete transition kernels, in order to avoid measure-theoretic complications.

**Definition 1.3.1** (the distribution construction $\mathcal{D}$)**.** Let $X$ be a countable set. The set $\mathcal{D}(X)$ denotes the set of *discrete probability distributions* over $X$, that is,

$$\mathcal{D}(X) \quad \stackrel{\text{def}}{=} \quad \Big\{\, \delta\colon X \to [0,1] \;\Big|\; \sum_{x \in X} \delta(x) = 1 \,\Big\}.$$

As a transition system is a set with nondeterministic transitions (Definition 1.2.1), a Markov chain is a set with probabilistic transitions.

**Definition 1.3.2** (Markov chain)**.** A *Markov chain (MC)* is a pair $\mathcal{S} = (X, \delta)$ of a counbable set $X$ of *states* and a function $\delta\colon X \to \mathcal{D}(X)$ called a *transition kernel*.

The condition says In a path $x_0 \to x_1 \to \cdots \to x_n$,

**Definition 1.3.3.** In the setting of Definition 1.3.2, a *path* in $\mathcal{S}$ is a finite sequence $x_0 \to x_1 \to \cdots \to x_n$ such that $\delta(x_i)(x_{i+1}) > 0$ for each $i \in [0, n-1]$, that is, such that each transition happens with a non-zero probability. The *length* of a path is defined similarly to Definition 1.2.1.

Each path $x_0 \to x_1 \to \cdots \to x_n$ comes with its *(path) probability* defined in the following natural manner:

$$\delta(x_0 \to x_1 \to \cdots \to x_n) \stackrel{\text{def}}{=} \delta(x_0)(x_1) \cdot \delta(x_1)(x_2) \cdot \cdots \cdot \delta(x_{n-1})(x_n).$$

We restrict ourselves to finite paths since, once we consider infinite paths, the set of paths is uncountable and we need measure theory. For our examples regarding safety and reachability, it suffices to deal with finite paths.

### 1.3.2 Safety and Reachability

The problems we are interested in are best formulated using the following quantitative notion of predicate.

**Definition 1.3.4** (fuzzy predicate)**.** A *fuzzy predicate* over a set $X$ is a function $p\colon X \to [0, 1]$, i.e. an element $p \in [0, 1]^X$ of the function space.

**Remark 1.3.5.** When $X$ is an uncountable set, it is natural to assume some measure-theoretic structure on $X$ and to require measurability of $p$. We do not do so for simplicity; we are interested in countable $X$ anyway.

... to be continued. Show an iteration algorithm and show that it may not terminate. Fixed point characterization, solving it, value iteration. Hint optimistic value iteration.

# Chapter 2

# Fixed Points in Complete Lattices

(Some leader here)

We assume the familiarity with basic order theory. See e.g. [6].

## 2.1 Complete Lattices

### 2.1.1 Posets

A *partially ordered set* (*poset* in short) is, as usual, a set $X$ equipped with a binary relation $\sqsubseteq \subseteq X \times X$ that is subject to the following axioms:

- reflexivity ($x \sqsubseteq x$ for each $x \in X$),
- transitivity ($x \sqsubseteq y$ and $y \sqsubseteq z$ imply $x \sqsubseteq z$, for each $x, y, z \in X$), and
- anti-symmetry ($x \sqsubseteq y$ and $y \sqsubseteq x$ imply $x = x$, for each $x, y \in X$).

Such a relation $\sqsubseteq$ is called a *partial order* or simply an *order*. When the axioms other than anti-symmetry are satisfied, it is called a *preorder*.

**Notation 2.1.1** ($\sqsubseteq$ for partial orders)**.** We use the symbol $\sqsubseteq$ as the metavariable for partial orders. Accordingly, we will be using $\bigsqcap$ and $\bigsqcup$ for inf's and sup's, etc.

Many references use $\leq$ as a metavariable for partial orders. We do not do so, since it causes confusion with some common concrete partial orders, such as the one between real numbers. See **??**.

Monotone functions are those which preserve order.

**Definition 2.1.2** (monotone function)**.** Let $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ be posets, and $f\colon X \to Y$ be a function. We say $f$ is *monotone* if, for each $x, x' \in X$, $x \sqsubseteq_X x'$ implies $f(x) \sqsubseteq_Y f(x')$.

### 2.1.2 Infimums and Supremums

**Definition 2.1.3** (infimum $\bigsqcap S$, supremum $\bigsqcup S$)**.** Let $(X, \sqsubseteq)$ be a poset, and $S \subseteq X$ be a subset of $X$.

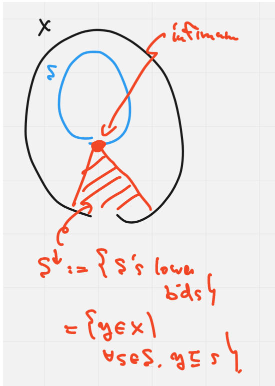



Figure 2.1: the infimum $\bigsqcap S$ of $S \subseteq X$

We say that $x \in X$ is the *infimum* (or *inf* in shoft) of $S$ if 1) $x$ is a *lower bound* of $S$ (meaning $x \sqsubseteq s$ for each $s \in S$), and 2) $x$ is the greatest among such, meaning, for each $y \in X$,

$$\big(y \sqsubseteq s \text{ for each } s \in S\big) \quad \text{implies} \quad y \sqsubseteq x.$$

An infimum of $S$, if it exists, is necessarily unique in a poset $(X, \sqsubseteq)$. The infimum of $S$ is denoted by $\bigsqcap S$.

Symmetrically, we say that $x \in X$ is the *supremum* (*sup* in shoft) of $S$ if 1) $x$ is an *upper bound* of $S$ (meaning $s \sqsubseteq x$ for each $s \in S$), and 2) $x$ is the least among such, meaning, for each $y \in X$,

$$\big(s \sqsubseteq y \text{ for each } s \in S\big) \quad \text{implies} \quad x \sqsubseteq y.$$

A supremum of $S$, if it exists, is necessarily unique in a poset $(X, \sqsubseteq)$ (Exercise 2.1). The supremum of $S$ is denoted by $\bigsqcup S$.

The notion of infimum is illustrated in Figure 2.1.

The above definition of infimums and supremums can be equivalently described by the following "universal properties" (also called "universalities"). The double lines here denote two-way implications: the top condition implies the bottom one; and vice versa.

$$\frac{y \sqsubseteq s \quad \text{for each } s \in S}{y \sqsubseteq \bigsqcap S} \tag{2.1}$$

$$\frac{s \sqsubseteq y \quad \text{for each } s \in S}{\bigsqcup S \sqsubseteq y} \tag{2.2}$$

One can derive from the universal property, for example, that $\bigsqcup S$ is indeed an upper bound of $S$:

$$\begin{array}{ll} \bigsqcup S \sqsubseteq \bigsqcup S & \text{by reflexivity, thus} \\ s \sqsubseteq \bigsqcup S \quad \text{for each } s \in S, & \text{using (2.2) upwards, taking } \bigsqcup S \text{ as } y. \end{array}$$

The characterizations in (2.1) and (2.2) are important: they naturally pave the way to the categorical generalization of inf's and sup's (namely limits and colimits); moreover, one often finds them to be the most useful form of the definition when it comes to proving properties. See e.g. Propositions 2.1.8 and 2.1.18.

We formally state the characterizations in (2.1) and (2.2) for the record.

**Proposition 2.1.4.** *Let $(X, \sqsubseteq)$ be a poset, $S \subseteq X$, and $x \in X$.*

1. *The element $x$ is the infimum $\bigsqcap S$ if and only if the following hold:*

$$\frac{y \sqsubseteq s \quad \text{for each } s \in S}{y \sqsubseteq x} \qquad \text{for each } y \in X.$$

2. *The element $x$ is the supremum $\bigsqcup S$ if and only if the following hold:*

$$\frac{s \sqsubseteq y \quad \text{for each } s \in S}{x \sqsubseteq y} \qquad \text{for each } y \in X. \qquad \square$$

**Proposition 2.1.5** ($\top, \bot$ in a complete lattice)**.** *A complete lattice $(L, \sqsubseteq)$ has the greatest and least elements, commonly denoted by $\top$ and $\bot$. The greatest element $\top$ is described as both $\bigsqcap \emptyset$ and $\bigsqcup L$; the least element $\bot$ is described as both $\bigsqcup \emptyset$ and $\bigsqcap L$.*

**Notation 2.1.6** (binary infimum and supermum $\sqcap, \sqcup$)**.** Let $(X, \sqsubseteq)$ be a poset, and $x, y \in X$. We write $x \sqcap y$ for $\bigsqcap\{x, y\}$. Similarly, we write $x \sqcup y$ for $\bigsqcup\{x, y\}$.

### 2.1.3 Complete (Semi)lattices

**Definition 2.1.7** (complete (semi)lattice)**.** A poset $(L, \sqsubseteq)$ is called a *complete lattice* if, for each subset $S \subseteq L$, its infimum $\bigsqcap S$ and its supremum $\bigsqcup S$ exist.

A poset $(L, \sqsubseteq)$ is called a *complete meet-semilattice* (or *complete $\bigsqcap$-semilattice*) if $\bigsqcap S$ exists for each $S \subseteq L$. Similarly, $(L, \sqsubseteq)$ is called a *complete join-semilattice* (or *complete $\bigsqcup$-semilattice*) if $\bigsqcup S$ exists for each $S \subseteq L$.

It turns out that a complete semilattice is necessarily a complete lattice—see the proposition below.

**Proposition 2.1.8** (complete semilattices are complete lattices)**.** *In a complete $\bigsqcup$-semilattice $(L, \sqsubseteq)$, each subset $S \subseteq L$ has its infimum $\bigsqcap S$.*

*Symmetrically, in a complete $\bigsqcap$-semilattice $(L, \sqsubseteq)$, each subset $S \subseteq L$ has its supremum $\bigsqcup S$.*

*Proof.* We prove the first statement. The rest is shown symmetrically.

Let $S \subseteq L$ be an arbitrary subset. We let $S^{\downarrow}$ be the set of its lower bounds, that is,

$$S^{\downarrow} \;:=\; \{y \in L \mid y \sqsubseteq s \text{ for each } s \in S\}.$$

Since $S^{\downarrow} \subseteq L$ is a subset of $L$, it has its supremum $\bigsqcup S^{\downarrow}$ in the complete $\bigsqcup$-semilattice $(L, \sqsubseteq)$. We claim that $\bigsqcup S^{\downarrow}$ is the infimum of $S$.

To prove the claim, it suffices to show the two-way implications in the characterization in (2.1), that is, we need to show

$$\frac{y \sqsubseteq s \quad \text{for each } s \in S}{y \sqsubseteq \bigsqcup S^{\downarrow}}. \tag{2.3}$$

For the downward implication in (2.3),

$$
\begin{array}{lll}
y \sqsubseteq s \quad \text{for each } s \in S & & \\
\Longrightarrow \quad y \in S^{\downarrow} & & \text{by def. of } S^{\downarrow} \\
\Longrightarrow \quad y \sqsubseteq \bigsqcup S^{\downarrow} & & \text{since } \bigsqcup S^{\downarrow} \text{ is an upper bound of } S^{\downarrow}.
\end{array}
$$

For the upward implication in (2.3), we first observe that

$$\bigsqcup S^{\downarrow} \sqsubseteq s \quad \text{for each } s \in S. \tag{2.4}$$

Indeed, $\bigsqcup S^{\downarrow} \sqsubseteq s$ is equivalent to

$$t \sqsubseteq s \quad \text{for each } t \in S^{\downarrow}$$

by (2.2), and the latter holds by the definition of $S^{\downarrow}$.

Now we have

$$
\begin{array}{lll}
y \sqsubseteq \bigsqcup S^{\downarrow} & & \\
\Longrightarrow \quad y \sqsubseteq s \quad \text{for each } s \in S & & \text{by (2.4) and transitivity,}
\end{array}
$$

which shows the upward implication in (2.3). □

We use the following construction of complete lattices in many examples. Its proof is easy (Exercise 2.5).

**Proposition 2.1.9** (complete lattice-valued function space)**.** *Let $L$ be a complete lattice, and $X$ be a set. The* function space

$$L^X \;\coloneqq\; \{k \colon X \to L\}$$

*is again a complete lattice, by the* pointwise order $\sqsubseteq_{L^X}$ *that is defined by*

$$k \sqsubseteq_{L^X} l \qquad \Longleftrightarrow \qquad k(x) \sqsubseteq_L l(x) \quad \textit{for each } x \in X.$$ □

**Example 2.1.10.** 1. The two-element set $2 \stackrel{\text{def}}{=} \{0, 1\}$, with the order $0 \sqsubseteq 1$, is a complete lattice.

2. The singleton $1 = \{0\}$, with the trivial order $0 \sqsubseteq 0$, is a complete lattice.

**Example 2.1.11.** 1. The *unit interval* $[0, 1] \coloneqq \{r \in \mathbb{R} \mid 0 \leq r \leq 1\}$ is the set of real numbers between 0 and 1. Taking its usual order $\leq$ as $\sqsubseteq$, the poset $([0, 1], \sqsubseteq)$ is a complete lattice.

2. Reversing the order in the last example, i.e. letting $\sqsubseteq' \coloneqq \geq$, the poset $([0, 1], \sqsubseteq') = ([0, 1], \geq)$ is a complete lattice, too. Note that, in general, the dual of a poset (obtained by reversing the order) is again a poset. Moreover, the dual of a complete lattice is again a complete lattice.

3. The set $\mathbb{R}$ with the usual order $\leq$ is *not* a complete lattice. For example, it does not have the greatest or least element (cf. Proposition 2.1.5).

4. Any (bounded) closed interval $[a, b] \coloneqq \{r \in \mathbb{R} \mid a \leq r \leq b\}$ is a complete lattice. It is *isomorphic* to the unit interval $[0, 1]$, via a suitable order-preserving bijection.

5. An interval, if it is not closed, is not a complete lattice with respect to the usual order $\leq$. Examples are $[a, b)$, $(a, b]$, $(-\infty, b]$, etc., where $a, b \in \mathbb{R}$.

**Example 2.1.12.** 1. For any set $X$, the *powerset* $\mathcal{P}(X) := \{S \subseteq X\}$ together with the inclusion order

$$S \sqsubseteq T \quad \overset{\text{def}}{\iff} \quad S \subseteq T,$$

is a complete lattice. Its supremum $\bigsqcup$ is computed by set-theoretic union $\bigcup$; its infimum $\bigsqcap$ is computed by set-theoretic intersection $\bigcap$.

2.

Introduce $2^X$ as the set of predicates. Say that it is isomorphic to $\mathcal{P}(X)$ but we do *not* use the notations interchangeably. They are instances of different constructs in a general setting, and they happen to coincide in this specific case of subsets and Boolean predicates. (Notational convention) Say that, however, we exploit the isomorphism when we describe an element of $2^X$. For example, letting $f\colon X \to Y$ be a function and $q\colon Y \to 2$ be a Boolean predicate over $Y$, the Boolean predicate $q \circ f\colon X \to 2$ is often denoted by the corresponding subset $\{x \in X \mid f(x) \in q\}$.

3. A *topological space* is a pair $(X, \mathcal{O})$ of a set $X$ and a system $\mathcal{O} \subseteq \mathcal{P}(X)$ of *open sets*, where the latter is subject to the following axioms:

   (a) $\emptyset \in \mathcal{O}, X \in \mathcal{O}$;

   (b) $\mathcal{O}$ is closed under *finite* intersection: for any $n \in \mathbb{N}$ and $S_1, \dots, S_n \in \mathcal{O}$, we have their (set-theoretic) intersection belonging to $\mathcal{O}$, that is, $S_1 \cap \cdots \cap S_n \in \mathcal{O}$.

   (c) $\mathcal{O}$ is closed under *arbitrary* union: for any index set $I$ (that can be infinite) and any $I$-indexed family $(S_i)_{i \in I}$ of open sets (i.e. $S_i \in \mathcal{O}$ for each $i \in I$), we have $\bigcup_{i \in I} S_i \in \mathcal{O}$.

   (This is a definition by systems of open sets. Equivalent definitions can be given by neighborhoods, closed sets, etc.)

   Topological spaces play important roles in many topics of theoretical computer science and logic; see e.g. [**?**, 14, 16]. One possible intuition is as follows: an open set (i.e. an element $T \in \mathcal{O}$) is an *event observable by finitary means*. Then the above three axioms admit natural interpretation:

   - the axiom 3a means that $\emptyset$ and the whole space $X$ are both observable ("always false" and "always true");
   - the axiom 3b means that the event $S_1 \cap \cdots \cap S_n \in \mathcal{O}$ is finitarily observable, by the finite combination of observations of $S_1, \dots, S_n$ (infinite intersections $\bigcap_{i \in I} S_i$ are prohibited since the combination becomes infinite); and
   - the axiom 3c means that the event $\bigcup_{i \in I} S_i$ by *getting lucky*, i.e. by a fortunate choice of a suitable choice of $i$ and then conducting the corresponding finitary observation of $S_i$. (Therefore the intuition here assumes angelic nondeterminism.)

Given a topological space $(X, \mathcal{O})$, the family $\mathcal{O}$ of open sets ordered by the inclusion order $\sqsubseteq \coloneqq \subseteq$, is a complete lattice. Indeed, $\mathcal{O}$ is closed under set-theoretic union $\bigcup$, which equips $\mathcal{O}$ with supermums. (One can use Exercise 2.8 for a precise argument, taking $L = \mathcal{P}(X)$ and $L' = \mathcal{O}$.) Then, by Proposition 2.1.8, $\mathcal{O}$ is a complete lattice.

The infimum $\bigsqcap_{i \in I} S_i$ in $\mathcal{O}$ is in general different from the set-theoretic intersection $\bigcap_{i \in I} S_i$: following the proof of Proposition 2.1.8, we have

$$\bigsqcap_{i \in I} S_i \;=\; \bigcup\{T \in \mathcal{O} \mid T \subseteq S_i \text{ for each } i \in I\} \;=\; \bigcup\{T \in \mathcal{O} \mid T \subseteq \bigcap_{i \in I} S_i\}.$$

See Exercise 2.6.

**Example 2.1.13.** Thanks to Proposition 2.1.9, the following function spaces (among others) are complete lattices with the pointwise order. Here $X$ is an arbitrary set.

- (From Example 2.1.10) $2^X$, with $2 = \{\bot \sqsubseteq \top\}$. This is isomorphic to the powerset lattice $\mathcal{P}(X)$ (Example 2.1.12).
- (From Example 2.1.11) $[0,1]^X$. An element of this set (a function $k \colon X \to [0,1]$) is called a (1-bounded) *fuzzy predicate*.
- (From Example 2.1.12) $\mathcal{O}^X$ for any topological space $(Y, \mathcal{O})$.

Some measure-theoretic examples? See [**?**, **?**]

### 2.1.4 Morphisms of Complete (Semi)lattices

By Proposition 2.1.8, a poset is a complete lattice if and only if it is a complete ($\bigsqcap$- or $\bigsqcup$-)semilattice. Their difference appears, however, when we talk about *morphisms* (i.e. structure-preserving maps) between them.

**Definition 2.1.14** (morphism of complete (semi)lattices)**.** Let $(L, \sqsubseteq_L)$ and $(M, \sqsubseteq_M)$ be complete lattices, and $f \colon L \to M$ be a function.

- We say $f$ is a $\bigsqcap$*-preserving map* (or a *morphism of complete* $\bigsqcap$*-semilattices*) if, for each $S \subseteq L$, $f(\bigsqcap_L S) = \bigsqcap_M f(S)$. Here $f(S) \coloneqq \{f(s) \mid s \in S\}$.
- Similarly, $f$ is a $\bigsqcup$*-preserving map* (or a *morphism of complete* $\bigsqcup$*-semilattices*) if, for each $S \subseteq L$, $f(\bigsqcup_L S) = \bigsqcup_M f(S)$.
- We say $f$ is a *morphism of complete lattices* if $f$ is both $\bigsqcap$-preserving and $\bigsqcup$-preserving.

Note that the notions of $\bigsqcap$- and $\bigsqcup$-preserving map can be defined more generally between posets (instead of between complete lattices). In this case, preservation of infimums/supremums means preservation of those which exist. See Proposition 2.1.18.

**Definition 2.1.15** (categories of complete lattices)**.** We define three categories $\mathbf{CLat}_{\bigsqcup}, \mathbf{CLat}_{\bigsqcap}, \mathbf{CLat}$ as follows. They all have complete lattices as objects.

- In $\mathbf{CLat}_{\bigsqcap}$, complete lattices are regarded as complete $\bigsqcap$-semilattices. Therefore the arrows are their morphisms, namely $\bigsqcap$-preserving maps.
- In $\mathbf{CLat}_{\bigsqcup}$, complete lattices are regarded as complete $\bigsqcup$-semilattices, and the arrows are $\bigsqcup$-preserving maps.

– In **CLat**, arrows are morphisms of complete lattices (i.e. both $\bigsqcap$- and $\bigsqcup$-preserving).

In our application to model checking, we often use $\bigsqcap$-preserving maps, $\bigsqcup$-preserving maps, and monotone maps. The use of morphism of complete lattices is rare.

The following basic property is worth noting. Its proof is somewhat similar to that for Proposition 2.1.8 and is a good exercise. Its categorical generalization is *Freyd's adjoint functor theorem* which we will discuss in Chapter 5. finer pointer

**Theorem 2.1.16** (adjunction between complete lattices)**.** *Let $(L, \sqsubseteq_L)$ and $(M, \sqsubseteq_M)$ be complete lattices.*

$$\begin{tikzcd} L \arrow[r, bend left, "f"] \arrow[r, phantom, "\perp"] & M \arrow[l, bend left, "g"] \end{tikzcd} \tag{2.5}$$

1. *Let $f\colon L \to M$ be a $\bigsqcup$-preserving map. Then there exists a function $g\colon M \to L$ that is $\bigsqcap$-preserving and satisfies*

$$\frac{x \sqsubseteq_L g(y)}{f(x) \sqsubseteq_M y} \quad \text{for each } x \in L, y \in M. \tag{2.6}$$

2. *Conversely, let $g\colon M \to L$ be a $\bigsqcap$-preserving map. Then there exists a function $f\colon L \to M$ that is $\bigsqcup$-preserving and satisfies the same property as* (2.6)*.*

*Proof.* We show the first statement; the second one is shown symmetrically.

Given such $f$, we let $g\colon M \to L$ be defined as follows: for each $y \in M$,

$$g(y) \;\coloneqq\; \bigsqcup\{x \in L \mid f(x) \sqsubseteq y\}, \tag{2.7}$$

that is, by first collecting all those $x$ that are carried by $f$ to some element below $y$, and then taking their supremum.

We first show that $g$ as defined in (2.7) satisfies the two-way implications in (2.6). The upward direction is easy:

$$\begin{aligned} & f(x) \sqsubseteq y \\ \Longrightarrow\;\; & x \in \{x \in L \mid f(x) \sqsubseteq y\} \\ \Longrightarrow\;\; & x \sqsubseteq \bigsqcup\{x \in L \mid f(x) \sqsubseteq y\} = g(y). \end{aligned}$$

For the downward direction, we first note that $f(g(y)) \sqsubseteq y$ holds for each $y \in Y$ (this is an important property—see Proposition 2.1.18.2). Indeed,

$$\begin{aligned} f(g(y)) &= f\bigl(\bigsqcup\{x \in L \mid f(x) \sqsubseteq y\}\bigr) && \text{by def. of } g \\ &= \bigsqcup\{f(x) \mid x \in L, f(x) \sqsubseteq y\} && f \text{ is } \bigsqcup\text{-preserving} \\ &\sqsubseteq y. \end{aligned}$$

This is used in the following.

$$\begin{aligned} & x \sqsubseteq g(y) \\ \Longrightarrow\;\; & f(x) \sqsubseteq f(g(y)) && \text{since } f \text{ is monotone (Exercise 2.4)} \end{aligned}$$

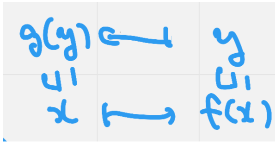



Figure 2.2: adjunction between posets

$$\Longrightarrow \quad f(x) \sqsubseteq y \qquad \text{by } f(g(y)) \sqsubseteq y \text{ and transitivity.}$$

This establishes the two-way implications in (2.6).

It remains to show that $g$ in (2.7) is $\bigsqcap$-preserving. In fact, this is a general property of a function $g$ satisfying (2.6)—see Definition 2.1.17 and Proposition 2.1.18.1a later. This conclude the proof. □

Situations such as (2.5) are commonly called *Galois connections*; the notion is formally defined below for the record. Its use is actively pursued in the field of *abstract interpretation* [4]. It is a special case of the categorical notion of *adjunction* [13, Chapter IV], restricted from categories to posets; see Chapter 5. In this book, therefore, Galois connections will be often called adjunctions, too.

better pointer

Here is some general theory of Galois connections. It can be thought of as an exercise of lattice-theoretic reasoning; it will also prepare readers for a fully category-theoretic treatment of adjunctions, found e.g. in [13, Chapter IV].

**Definition 2.1.17** (adjunction (Galois connection) between posets)**.** Let $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ be posets, and $f\colon X \to Y$ and $g\colon Y \to X$ be monotone maps. We say that $f$ is the *left adjoint* to $g$ if the following two-way implications hold for each $x \in X, y \in Y$.

$$\frac{x \sqsubseteq_X g(y)}{\overline{f(x) \sqsubseteq_Y y}} \tag{2.8}$$

We say the following for the same mathematical condition, too: $g$ is the *right adjoint* to $f$; $f$ and $g$ form an *adjunction* between posets; $f$ and $g$ form a *Galois connection*. All these conditions are denoted by $f \dashv g$, or

$$\begin{tikzcd} X \arrow[r, bend left, "f"] \arrow[r, phantom, "\perp"] & Y \arrow[l, bend left, "g"] \end{tikzcd}. \tag{2.9}$$

Implicit in the above description is the uniqueness of (left and right) adjoints. See Exercise 2.7.

One can think of an adjunction as a notion of pseudo-inverse: $f$ and $g$ go in the opposite directions; while they do not quite constitute proper inverses (being proper inverses would mean that $g \circ f = \mathrm{id}_X$ and $f \circ g = \mathrm{id}_Y$), they do respect order on both sides in the sense of (2.8). The last is illustrated in Figure 2.2.

Here are some general properties of adjunctions between posets. They generalize smoothly to adjunctions between categories (Chapter 5). Item 1a can be seen as a converse of Theorem 2.1.16. Items 1b and 2 are converse to each other.

**Proposition 2.1.18.** *1. Let $f\colon X \to Y$ and $g\colon Y \to X$ form an adjunction $f \dashv g$, as in Definition 2.1.17.*

(a) *The monotone function $f$ is $\bigsqcup$-preserving. That is, for each $S \subseteq X$, if $\bigsqcup_X S$ exists in $X$, then $\bigsqcup_Y f(S)$ exists too, and moreover $f(\bigsqcup_X S) = \bigsqcup_Y f(S)$.*
*Symmetrically, $g$ is $\bigsqcap$-preserving.*

(b) *We have $\mathrm{id}_X \sqsubseteq g \circ f$ with respect to the pointwise order between functions of the type $X \to X$. That is, for each $x \in X$, we have $x \sqsubseteq_X g(f(x))$.*
*Symmetrically, we have $f \circ g \sqsubseteq \mathrm{id}_Y$ with respect to the pointwise order between functions of the type $Y \to Y$. That is, for each $y \in Y$, we have $f(g(y)) \sqsubseteq_Y y$.*

2. *Let $f \colon X \to Y$ and $g \colon Y \to X$ be monotone maps between posets, and assume that $\mathrm{id}_X \sqsubseteq g \circ f$ and $f \circ g \sqsubseteq \mathrm{id}_Y$ hold with respect to the same pointwise order as in Item 1b. Then $f$ and $g$ form an adjunction ($f \dashv g$).*

*Proof.* For Item 1a, we prove its first statement. The second is shown symmetrically. Its proof exemplifies the power of the characterization (2.2) of supremums. The proof can be succinctly presented as follows.

$$\dfrac{\dfrac{\dfrac{\bigsqcup f(S) \sqsubseteq y}{f(s) \sqsubseteq y \quad \text{for each } s \in S}}{s \sqsubseteq g(y) \quad \text{for each } s \in S}\; f \dashv g}{\dfrac{\bigsqcup S \sqsubseteq g(y)}{f(\bigsqcup S) \sqsubseteq y}\; f \dashv g}$$

Let us spell out the details. Let $S \subseteq X$ be an arbitrary subset. It suffices to show the following:

$$\frac{f(s) \sqsubseteq y \quad \text{for each } s \in S}{f(\bigsqcup S) \sqsubseteq y}\,; \tag{2.10}$$

that is, that $f(\bigsqcup S)$ satisfies the universality of $\bigsqcup f(S)$ in (2.2) (see also Proposition 2.1.4). Here we have also used the definition $f(S) = \{f(s) \mid s \in S\}$.

The two-way implications (2.10) are shown as follows, crucially relying on (2.8).

$$\begin{array}{ll}
f(s) \sqsubseteq y \quad \text{for each } s \in S & \\
\iff \quad s \sqsubseteq g(y) \quad \text{for each } s \in S & \text{by } f \dashv g, \text{ see (2.8)} \\
\iff \quad \bigsqcup S \sqsubseteq g(y) & \text{by the universality of } \bigsqcup S \text{ in (2.2)} \\
\iff \quad f(\bigsqcup S) \sqsubseteq y & \text{by } f \dashv g, \text{ see (2.8).}
\end{array}$$

This proves $f(\bigsqcup S) = \bigsqcup f(S)$.

Item 1b is easy: in the bottom of (2.8), we have $f(x) \sqsubseteq_Y f(x)$ by reflexivity, thus by (2.8) we have $x \sqsubseteq_X g(f(x))$. The other half is similar.

For Item 2, we show the two-way implications in (2.8), assuming $\mathrm{id}_X \sqsubseteq g \circ f$ and $f \circ g \sqsubseteq \mathrm{id}_Y$. The downward implication is shown as follows.

$$\begin{array}{ll}
x \sqsubseteq_X g(y) & \\
\implies \quad f(x) \sqsubseteq_Y f(g(y)) & \text{since } f \text{ is monotone, cf. Exercise 2.4} \\
\implies \quad f(x) \sqsubseteq_Y y & f(g(y)) \sqsubseteq_Y y \text{ (by assumption), and by transitivity.}
\end{array}$$

The upward direction is shown symmetrically. □

## 2.2 Fixed Points in Complete Lattices

Make subsections sections

We have discussed, in **??**, 1) the significance of fixed points in theoretical computer science as a study of *infinitary* behaviors by *finitary* means, and 2) different settings that provide necessary fixed points. In this chapter, we will study the third setting in the list of **??**, namely the order-theoretic setting with complete lattices and monotone maps. We will exhibit two different characterizations of extremal fixed points in the setting: the *Knaster–Tarski* and *Cousot–Cousot* theorems. They yield different reasoning principles for those fixed points.

In what follows, we discuss these two characterizations. Their generalization to the categorical setting (the last in the list) will be discussed in Chapter 5.

### 2.2.1 The Knaster–Tarski Theorem

Let $(L, \sqsubseteq)$ be a complete lattice, and $f\colon L \to L$ be a function. There are many preservation properties of $f$ that one can think of—$f$ can be $\bigsqcap$-preserving, in which case we have $f(\top) = \top$ for the greatest element $\top = \bigsqcap \emptyset \in L$ (Proposition 2.1.5). This means that, for $\bigsqcap$-preserving $f$, its greatest fixed point is trivially $\top$. Similarly, for $\bigsqcup$-preserving $f$, its least fixed point is $\bot$.

It turns out that $f$ being (only) monotone is enough for ensuring existence of fixed points. This axiomatization covers many examples in which $f$ has nontrivial least and greatest fixed points.

**Theorem 2.2.1** (Knaster–Tarski)**.** *Let $(L, \sqsubseteq)$ be a complete lattice, and $f\colon L \to L$ be a monotone function. Then the following hold.*

1. *The set*
$$\mathrm{Pre}(f) \;\coloneqq\; \{x \in L \mid f(x) \sqsubseteq x\}$$
*of* prefixed points *of $f$ is a complete lattice with respect to the order $\sqsubseteq$ inherited from $L$. Moreover, its infimum $\bigsqcap_{\mathrm{Pre}(f)}$ coincides with the infimum $\bigsqcap_L$ of $L$ ($\bigsqcap_{\mathrm{Pre}(f)}$ is "computed in $L$").*

2. *The least prefixed point, which exists by the above, is a fixed point; it is therefore the least fixed point (lfp, denoted by $\mu f$). Thus we obtain*
$$\mu f \;=\; \bigsqcap\{x \in L \mid f(x) \sqsubseteq x\}.$$

3. *Symmetrically, the set*
$$\mathrm{Post}(f) \;\coloneqq\; \{x \in L \mid x \sqsubseteq f(x)\}$$
*of* postfixed points *of $f$ is a complete lattice with respect to the order $\sqsubseteq$ inherited from $L$. Moreover, its supremum $\bigsqcup_{\mathrm{Post}(f)}$ coincides with the supremum $\bigsqcup_L$ of $L$ ($\bigsqcup_{\mathrm{Post}(f)}$ is "computed in $L$").*

4. *The greatest postfixed point, which exists by the above, is a fixed point; it is therefore the greatest fixed point (denoted by $\nu f$). Thus we obtain*
$$\nu f \;=\; \bigsqcup\{x \in L \mid x \sqsubseteq f(x)\}.$$

5. *Furthermore, the set*

$$\mathrm{Fix}(f) \;:=\; \{x \in L \mid f(x) = x\}$$

*of all fixed points of* $f$ *is a complete lattice with respect to the order* $\sqsubseteq$ *inherited from* $L$.

In the following proof, we explicitly distinguish inf's and sup's taken in different posets (such as $\bigsqcup_L$ vs. $\bigsqcup_{\mathrm{Pre}(f)}$). Blurring the distinction is often a source of confusion.

*Proof.* For Item 1, we first show that the set $\mathrm{Pre}(f)$ of prefixed points are closed under infimum $\bigsqcap_L$ of $L$, that is for each $S \subseteq \mathrm{Pre}(f)$,

$$\textstyle\bigsqcap_L S \in \mathrm{Pre}(f).$$

Indeed,

$$\begin{aligned} f(\textstyle\bigsqcap_L S) \;&\sqsubseteq\; \textstyle\bigsqcap_L \{f(x) \mid x \in S\} && \text{since } f \text{ is monotone, Exercise 2.3} \\ &\sqsubseteq\; \textstyle\bigsqcap_L \{x \mid x \in S\} && S \subseteq \mathrm{Pre}(f) \text{ so } f(x) \sqsubseteq x \text{ for } x \in S \\ &=\; \textstyle\bigsqcap_L S. \end{aligned}$$

This infimum $\bigsqcap_L S$—it is computed in $L$ and it happens to be in $\mathrm{Pre}(f)$—is easily seen to be the infimum $\bigsqcap_{\mathrm{Pre}(f)} S$ in the poset $(\mathrm{Pre}(f), \sqsubseteq)$ (Exercise 2.8). Therefore $(\mathrm{Pre}(f), \sqsubseteq)$ is a complete $\bigsqcap$-semilattice, and thus is a complete lattice by Proposition 2.1.8. This concludes the proof of Item 1.

In Item 1, while we have $\bigsqcap_L = \bigsqcap_{\mathrm{Pre}(f)}$, supremums $\bigsqcup_L$ and $\bigsqcup_{\mathrm{Pre}(f)}$ may not coincide. The latter is described as in the proof of Proposition 2.1.8 using $\mathrm{Pre}(f)$ and $\bigsqcap_{\mathrm{Pre}(f)}$, and has little to do with $\bigsqcup_f$. For example, $\bigsqcup_L \emptyset = \bot$; if it were $\bigsqcup_L \emptyset = \bigsqcup_{\mathrm{Pre}(f)} \emptyset$, then we would have $f(\bot) = \bot$, forcing every monotone function to be $\bot$-preserving (which is not the case).

For Item 2, let $x_0 := \bigsqcap \mathrm{Pre}(f)$, the infimum of all prefixed points. (We have seen that it does not matter if $\bigsqcap$ denotes $\bigsqcap_L$ or $\bigsqcap_{\mathrm{Pre}(f)}$.) By Item 1, we have $x_0 \in \mathrm{Pre}(f)$; thus $x_0$ is the least prefixed point. We need to show that $x_0$ is in fact a fixed point. Now consider $f(x_0) \in L$; we have

$$\begin{aligned} & f(x_0) \sqsubseteq x_0 && \text{by } x_0 \in \mathrm{Pre}(f), \text{ and} \\ & f(f(x_0)) \sqsubseteq f(x_0) && \text{since } f \text{ is monotone.} \end{aligned}$$

Therefore $f(x_0) \in \mathrm{Pre}(f)$ is a prefixed point. By the definition of $x_0$ as the least element of $\mathrm{Pre}(f)$, we have $x_0 \sqsubseteq f(x_0)$. Combining with the fact that $x_0$ is a prefixed point, we obtain $x_0 = f(x_0)$. This fixed point $x_0$ is the least one: it is the least prefixed point; and all fixed points are prefixed points as well.

Items 3 & 4 are shown symmetrically to the above.

For Item 5, let $S \subseteq \mathrm{Fix}(f)$. It suffices, in view of Proposition 2.1.8, to show that the infimum $\bigsqcap_{\mathrm{Fix}(f)} S$ exists in $\mathrm{Fix}(f)$.

Towards the goal, we consider the subset

$$(\textstyle\bigsqcap_L S)^{\downarrow} \;:=\; \{x \in L \mid x \sqsubseteq \textstyle\bigsqcap_L S\}$$

of $L$. See Figure 2.3. Note that the infimum $\bigsqcap_L S$ exists in a complete lattice $L$; however it is not necessarily a fixed point of $f$. It is easily shown that $\bigsqcap_{\mathrm{Fix}(f)} S$

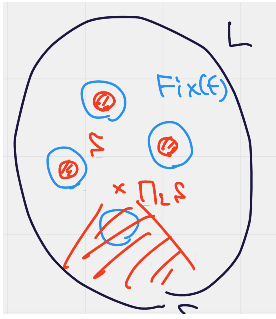



Figure 2.3: the set $(\bigsqcap_L S)^{\downarrow}$

must be below $\bigsqcap_L S$ if the former exists (Exercise 2.8); therefore we look for $\bigsqcap_{\mathrm{Fix}(f)} S$ in the set $(\bigsqcap_L S)^{\downarrow}$.

One can easily show that the set $(\bigsqcap_L S)^{\downarrow}$ is a complete lattice. Indeed, it is clear that all inf's and sup's of a subset $S' \subseteq (\bigsqcap_L S)^{\downarrow}$, computed in $L$, land in $(\bigsqcap_L S)^{\downarrow}$. Thus they are inf's and sup's in $(\bigsqcap_L S)^{\downarrow}$ as well (cf. Exercise 2.8).

We shall now show that $f\colon L \to L$ restricts to $(\bigsqcap_L S)^{\downarrow}$, that is, for any $x \in (\bigsqcap_L S)^{\downarrow}$, we have $f(x) \in (\bigsqcap_L S)^{\downarrow}$. Indeed, we have

$$
\begin{aligned}
&x \sqsubseteq \textstyle\bigsqcap_L S && \text{by } x \in (\textstyle\bigsqcap_L S)^{\downarrow},\\
&f(x) \sqsubseteq f(\textstyle\bigsqcap_L S) \sqsubseteq \textstyle\bigsqcap_L f(S) && \text{by } f\text{: monotone and Exercise 2.3, and}\\
&f(x) \sqsubseteq \textstyle\bigsqcap_L S && \text{since } S \subseteq \mathrm{Fix}(f) \text{ and thus } f(S) = \{f(s) \mid s \in S\} = S.
\end{aligned}
$$

Therefore we have established that 1) $(\bigsqcap_L S)^{\downarrow}$ is a complete lattice, and 2) $f\colon (\bigsqcap_L S)^{\downarrow} \to (\bigsqcap_L S)^{\downarrow}$ is a monotone function. We can apply Items 1–4 to this setting; towards the discovery of $\bigsqcap_{\mathrm{Fix}(f)} S$, we turn specifically to the greatest fixed point of $f\colon (\bigsqcap_L S)^{\downarrow} \to (\bigsqcap_L S)^{\downarrow}$. Let $x_0$ be the greatest fixed point.

We claim that $x_0 \in \mathrm{Fix}(f)$ is $\bigsqcap_{\mathrm{Fix}(f)} S$. It is a lower bound of $S$ in $\mathrm{Fix}(f)$, since $x_0$ is in $(\bigsqcap_L S)^{\downarrow}$. Let $y \in \mathrm{Fix}(f)$ be an arbitrary fixed point of $f$ below $S$. Then we have $y \sqsubseteq \bigsqcap_L S$, thus $y$ is a fixed point of $f$ in $(\bigsqcap_L S)^{\downarrow}$. By the definition of $x_0$ as the greatest, we conclude $y \sqsubseteq x_0$. $\square$

The above presentation of the Knaster–Tarski theorem slightly deviates from the usual presentation. It is adapted to the current context (especially towards the reasoning principles in Section 2.2.3), and thus puts more emphasis on Items 1–4 rather than on Item 5 (Item 5 is not very important for our purpose). Items 2 & 4 have a well-known categorical generalization called the *Lambek lemma*. See e.g. [10, 15].

To summarize, the Knaster–Tarski theorem (as presented in Theorem 2.2.1) characterizes

- the least fixed point $\mu f$ as the least *prefixed* point,
- and symmetrically, the greatest fixed point $\nu f$ as the greatest *postfixed* point.

The following notational convention, like in the modal $\mu$-calculus [2, 12], is useful especially when fixed point operators are nested.

**Notation 2.2.2.** The lfp $\mu f$ of a function $f$ is also denoted by $\mu u.\, f(u)$, using a variable $u$ that does not occur in $f$. Similarly, the gfp $\nu f$ is also denoted by $\nu u.\, f(u)$.

### 2.2.2 The Cousot–Cousot Theorem

Let us move on to the second characterization of the extremal (i.e. the least and greatest) fixed points of $f\colon L \to L$, where $L$ is a complete lattice and $f$ is a monotone map. The characterization uses sequences—that can be very long, over transfinite ordinals.

**Definition 2.2.3** (Cousot–Cousot sequence)**.** Let $(L, \sqsubseteq)$ be a complete lattice, and $f\colon L \to L$ be a monotone function. The *bottom-up Cousot–Cousot sequence* is the (transfinite) sequence

$$x_0 \ \sqsubseteq\ x_1 \ \sqsubseteq\ x_2 \ \sqsubseteq\ \cdots \ \sqsubseteq\ x_\omega \ \sqsubseteq\ x_{\omega+1} \ \sqsubseteq\ \cdots \ \sqsubseteq\ x_\alpha \ \sqsubseteq\ \cdots \tag{2.11}$$

defined inductively as follows. Here $\alpha$ is an arbitrary ordinal and $x_\alpha \in L$ for each $\alpha \in \mathbf{Ord}$,

- (Base case) We let $x_0 \coloneqq \bot$, where $\bot = \bigsqcup \emptyset$ is the minimum element of $L$.
- (Step case) For a successor ordinal $\alpha = \alpha' + 1$, we let $x_{\alpha'+1} \coloneqq f(x_{\alpha'})$.
- (Limit case) For a limit ordinal $\alpha$, we let $x_\alpha \coloneqq \bigsqcup\{x_\beta \mid \beta < \alpha\}$. Since $L$ is a complete lattice, such supremum exists.

The bottom-up Cousot–Cousot sequence (2.11) shall also be denoted as follows.

$$\bot \ \sqsubseteq\ f(\bot) \ \sqsubseteq\ f^2(\bot) \ \sqsubseteq\ \cdots \ \sqsubseteq\ f^\omega(\bot) \ \sqsubseteq\ f^{\omega+1}(\bot) \ \sqsubseteq\ \cdots \ \sqsubseteq\ f^\alpha(\bot) \ \sqsubseteq\ \cdots \tag{2.12}$$

Dually, the *top-down Cousot–Cousot sequence* is the sequence

$$\top \ \sqsupseteq\ f(\top) \ \sqsupseteq\ f^2(\top) \ \sqsupseteq\ \cdots \ \sqsupseteq\ f^\omega(\top) \ \sqsupseteq\ f^{\omega+1}(\top) \ \sqsupseteq\ \cdots \ \sqsupseteq\ f^\alpha(\top) \ \sqsupseteq\ \cdots . \tag{2.13}$$

Its precise definition is as follows: $x_0 \sqsupseteq x_1 \sqsupseteq \cdots$ with 1) $x_0 \coloneqq \top = \bigsqcap \emptyset$, 2) $x_{\alpha'+1} \coloneqq f(x_{\alpha'})$, and 3) $x_\alpha \coloneqq \bigsqcap\{x_\beta \mid \beta < \alpha\}$ for a limit ordinal $\alpha$.

**Lemma 2.2.4.** *In Definition 2.2.3, the bottom-up Cousot–Cousot sequence is indeed increasing: $\alpha \le \beta$ implies $f^\alpha(\bot) \sqsubseteq f^\beta(\bot)$. Dually, the top-down Cousot–Cousot sequence is indeed decreasing.*

*Proof.* By transfinite induction on the ordinal $\beta$. □

**Theorem 2.2.5** (Cousot–Cousot [5])**.** *Let $(L, \sqsubseteq)$ be a complete lattice, and $f\colon L \to L$ be a monotone function, as in Definition 2.2.3.*

1. *The bottom-up Cousot–Cousot sequence* (2.12) *stabilizes, that is, there exists an ordinal $\alpha_0$ such that $f^{\alpha_0}(\bot) = f^{\alpha_0+1}(\bot) = \cdots$. Moreover, its limit $f^{\alpha_0}(\bot) \in L$ is the least fixed point $\mu f$ of $f$.*
2. *Dually, the top-down Cousot–Cousot sequence* (2.13) *stabilizes, that is, there exists an ordinal $\alpha_0$ such that $f^{\alpha_0}(\top) = f^{\alpha_0+1}(\top) = \cdots$. Moreover, its limit $f^{\alpha_0}(\top) \in L$ is the greatest fixed point $\nu f$ of $f$.*

The theorem holds essentially because of the size limitation of $L$.

*Proof.* We prove Item 1; the proof of Item 2 is its dual.

We claim that there exists an ordinal $\alpha_0$ such that $f^{\alpha_0}(\bot) = f^{\alpha_0+1}(\bot)$. Assume not; then all the elements $f^{\alpha}(\bot)$ in the sequence (2.12) must be distinct. This yields an injection from the proper class **Ord** to a small set $L$, which should not exist.

Once $\alpha_0$ is chosen so that $f^{\alpha_0}(\bot) = f^{\alpha_0+1}(\bot)$ holds, it is easily shown that $f^{\alpha_0}(\bot) = f^{\beta}(\bot)$ holds for each $\beta$ such that $\alpha_0 \leq \beta$ (by induction on $\beta$). Thus we have shown the stabilization of the sequence 2.12.

The limit $f^{\alpha_0}(\bot)$ is a fixed point of $f$:

$$\begin{aligned} f\big(f^{\alpha_0}(\bot)\big) &= f^{\alpha_0+1}(\bot) \qquad \text{by def. of } f^{\alpha_0+1}(\bot) \\ &= f^{\alpha_0}(\bot) \qquad \text{by the choice of } \alpha_0. \end{aligned}$$

It remains to show that $f^{\alpha_0}(\bot)$ is the *least* fixed point. Let $y$ be an arbitrary fixed point, with $y = f(y)$. We claim $f^{\alpha}(\bot) \sqsubseteq y$ for each ordinal $\alpha \in \mathbf{Ord}$. This is easily proved by induction:

– (base case) $f^{0}(\bot) = \bot \sqsubseteq y$ since $\bot$ is the least element of $L$;
– (step case) assuming $f^{\alpha'}(\bot) \sqsubseteq y$, we have

$$f^{\alpha'+1}(\bot) \;=\; f\big(f^{\alpha'}(\bot)\big) \;\sqsubseteq\; f(y) \;=\; y,$$

where we crucially used the monotonicity of $f$;
– (limit case) assuing $f^{\beta}(\bot) \sqsubseteq y$ for each $\beta$ such that $\beta < \alpha$, we have

$$f^{\alpha}(\bot) \;=\; \bigsqcup\{f^{\beta}(\bot) \mid \beta < \alpha\} \;\sqsubseteq\; y.$$

Thus we have shown $f^{\alpha}(\bot) \sqsubseteq y$ for each $\alpha \in \mathbf{Ord}$; in particular, we have $f^{\alpha_0}(\bot) \sqsubseteq y$. □

We have now obtained two characterizations of extremal (i.e. the least and the greatest) fixed points. The Cousot–Cousot characterization (Theorem 2.2.5) is arguably more constructive: the construction of a Cousot–Cousot sequence is iterative (such as $\bot, f(\bot), f^2(\bot), \dots$); it eventually gives one an extremal fixed point. The sequence is infinitely long in general, so this "construction" is not really an algorithm. Nevertheless, in case the lattice $L$ is finite, the sequence is guaranteed to stabilize within finitely many steps. This is the case in some examples later; see e.g. Section 4.4.

The Kleene fixed-point theorem, another sequence-based characterization of fixed points, is probably better known in theoretical computer science (especially in domain theory [1]). In the Kleene case, the stabilization of sequences such as (2.12) is ensured not by the size of $L$ (as in the above proof) but by the "continuity," or the size, of $f$. The Kleene theorem will be discussed later in Section 2.2.4. The Kleene theorem plays an important role—more important than the Cousot–Cousot theorem—in the generalization from complete lattices to categories (Chapter 5).

### 2.2.3 Four Reasoning Principles

We have observed two different characterizations (Knaster–Tarski and Cousot–Cousot) for each of the two extremal fixed points (the least and the greatest).

Table 2.1: Fixed point approximation problems. "UA" is for underapproximation; "OA" is for overapproximation

| | |
|---|---|
| **(Common) Input** | A complete lattice $L$, a monotone function $f\colon L \to L$, and an element $x \in L$ |
| **Output** | (LFP-OA) $\mu f \sqsubseteq^{?} x$<br>(LFP-UA) $x \sqsubseteq^{?} \mu f$<br>(GFP-OA) $\nu f \sqsubseteq^{?} x$<br>(GFP-UA) $x \sqsubseteq^{?} \nu f$ |

Table 2.2: Four reasoning priciples, for least and greatest fixed points, by Knaster–Tarski and Cousot–Cousot

| | Knaster–Tarski | Cousot–Cousot |
|---|---|---|
| least fixed point $\mu f$ | **overapproximation**: $f(y) \sqsubseteq y$ implies $\mu f \sqsubseteq y$ | **underapproximation**: $f^{\alpha}(\bot) \sqsubseteq \mu f$ for each $\alpha \in \mathbf{Ord}$ |
| greatest fixed point $\nu f$ | **underapproximation**: $y \sqsubseteq f(y)$ implies $y \sqsubseteq \nu f$ | **overapproximation**: $\nu f \sqsubseteq f^{\alpha}(\top)$ for each $\alpha \in \mathbf{Ord}$ |

From these characterizations, we derive four $(= 2 \times 2)$ *reasoning principles* for extremal fixed points. They are summarized in Table 2.2.

The four reasoning principles are rarely mentioned explicitly in the formal verification literature; their systematic exposition, as we present below, is also rare. Nevertheless, it is surprising how many concrete formal verification techniques—whether they are for theorem proving or model checking—rely essentiallly on these reasoning principles.

We start with formalizing *approximation problems* of fixed points. Typically, fixed-point reasoning in formal verification aims at one of these problems.

**Definition 2.2.6** (approximation of fixed points)**.** The *least fixed point overapproximation (LFP-OA)* problem is formulated as follows.

| | |
|---|---|
| **Input** | A complete lattice $L$, a monotone function $f\colon L \to L$, and an element $x \in L$ |
| **Output** | $\mu f \sqsubseteq^{?} x$, that is, if $\mu f \sqsubseteq x$ holds or not. |

The *least fixed point underapproximation (LFP-UA)* problem is formulated similarly: its input is the same as above; its output is if $x \sqsubseteq \mu f$ holds or not.

The same problems are formulated for greatest fixed points, too, giving rise to the *GFP-OA* and *GFP-UA* problems. See Table 2.1 for a summary.

The following reasoning principles—summarized in Table 2.2—bridge the characterizations in Sections 2.2.1 and 2.2.2 and the approximation problems in Definition 2.2.6.

**Corollary 2.2.7** (the Knaster–Tarski and Cousot–Cousot reasoning principles)**.** *Let $L$ be a complete lattice, and $f\colon L \to L$ be a monotone function.*

1. *If $y \in L$ satisfies $f(y) \sqsubseteq y$, then $\mu f \sqsubseteq y$ holds.*
2. *For each ordinal $\alpha \in \mathbf{Ord}$, we have $f^{\alpha}(\bot) \sqsubseteq \mu f$.*

3. *If $y \in L$ satisfies $y \sqsubseteq f(y)$, then $y \sqsubseteq \nu f$ holds.*
4. *For each ordinal $\alpha \in \mathbf{Ord}$, we have $\nu f \sqsubseteq f^{\alpha}(\top)$.*

*Proof.* We prove the first two items; the rest is symmetric.

For Item 1, the statement follows immediately from Theorem 2.2.1.2, that is, that the least fixed point $\mu f$ is in fact the least *prefixed* point.

For Item 2, the statement follows from Theorem 2.2.5, that is, that the least fixed point $\mu f$ is the limit of the increasing (transfinite) chain

$$\bot \;\sqsubseteq\; f(\bot) \;\sqsubseteq\; f^2(\bot) \;\sqsubseteq\; \cdots$$

from (2.12). □

We present further consequences of Corollary 2.2.7. They directly connect to the approximation problems in Definition 2.2.6.

**Corollary 2.2.8** (verification and refutation by Knaster–Tarski and Cousot—Cousot)**.** *Let $L$ be a complete lattice, $f\colon L \to L$ be a monotone function, and $x \in L$.*

*Make this if and only if*

1. *(For LFP-OA)*
   (a) *(Verification) The existence of $y \in L$ such that $f(y) \sqsubseteq y$ and $y \sqsubseteq x$* verifies *the LFP-OA problem $\mu f \sqsubseteq^{?} x$:*

$$\frac{f(y) \sqsubseteq y \quad y \sqsubseteq x}{\mu f \sqsubseteq x}\ \text{(LFP-OA-V)}$$

   *Such $y$ is called a* Knaster–Tarski witness *for $\mu f \sqsubseteq x$.*

   (b) *(Refutation) The existence of $y \in L$ such that $y \sqsubseteq f^{\alpha}(\bot)$ (for some ordinal $\alpha \in \mathbf{Ord}$) and $y \not\sqsubseteq x$* refutes *the LFP-OA problem $\mu f \sqsubseteq^{?} x$:*

$$\frac{y \sqsubseteq f^{\alpha}(\bot) \quad y \not\sqsubseteq x}{\mu f \not\sqsubseteq x}\ \text{(LFP-OA-R)}$$

   *Such $y$ is called a* Cousot–Cousot witness *for $\mu f \not\sqsubseteq x$.*

*Similarly, we have the following implications.*

2. *(For LFP-UA)*

$$\frac{y \sqsubseteq f^{\alpha}(\bot) \quad x \sqsubseteq y}{x \sqsubseteq \mu f}\ \text{(LFP-UA-V)} \qquad \frac{f(y) \sqsubseteq y \quad x \not\sqsubseteq y}{x \not\sqsubseteq \mu f}\ \text{(LFP-UA-R)}$$

3. *(For GFP-OA)*

$$\frac{f^{\alpha}(\top) \sqsubseteq y \quad y \sqsubseteq x}{\nu f \sqsubseteq x}\ \text{(GFP-OA-V)} \qquad \frac{y \sqsubseteq f(y) \quad y \not\sqsubseteq x}{\nu f \not\sqsubseteq x}\ \text{(GFP-OA-R)}$$

4. *(For GFP-UA)*

$$\frac{y \sqsubseteq f(y) \quad x \sqsubseteq y}{x \sqsubseteq \nu f}\ \text{(GFP-UA-V)} \qquad \frac{f^{\alpha}(\top) \sqsubseteq y \quad x \not\sqsubseteq y}{x \not\sqsubseteq \nu f}\ \text{(GFP-UA-R)}$$

*Proof.* These are immediate consequences of Corollary 2.2.7. For example, in Item 1b, assuming $\mu f \sqsubseteq x$ implies

$$y \;\sqsubseteq\; f^{\alpha}(\bot) \;\sqsubseteq\; \mu f \;\sqsubseteq\; x,$$

where the second inequality is from Corollary 2.2.7. This contradicts with the assumption $y \not\sqsubseteq x$. □

One can see in the last corollary that the Knaster–Tarski and Cousot–Cousot reasoning principles are much like two sides of the same coin: when an approximation problem is verified by one, then it is refuted by the other.

We will exhibit some examples of Knaster–Tarski and Cousot–Cousot witnesses later in Section 3.1 and Chapter 4. The general tendency—when it comes to verification—is that Knaster–Tarski witnesses are more useful than Cousot–Cousot ones. That $y$ is a Knaster–Tarski witness can be checked by only one application of $f$; we just have to compare $y$ and $f(y)$. This is unlike Cousot–Cousot witnesses: computing $f^{\alpha}(\bot)$ or $f^{\alpha}(\top)$ for a smaller $\alpha$ usually does not bring much information; one usually has to go for rather big $\alpha$'s, applying $f$ to $\bot$ or $\top$ many times. This is costly.

For a Knaster–Tarski witness $y$, note that finding a good candidate of $y$ is a different problem, and how to do so has not been discussed in the theory so far. However, one can say that this is how the notion of Knaster–Tarski witness accommodates various *search heuristics*. There are various setting-specific heuristics for finding a candidate $y$, and users are free to choose any of them; once a choice of heuristics finds $y$, whether $y$ is indeed a witness can be easily checked by applying $f$ once.

### 2.2.4 The Kleene Theorem

In the Cousot–Cousot theorem, in case $f$ is not only monotone but also $\omega$-(co)continuous (it preserves the supremum/infimum of an $\omega$-chain $x_0 \sqsubseteq x_1 \sqsubseteq \cdots$ or $x_0 \sqsupseteq x_1 \sqsupseteq \cdots$), then the Cousot–Cousot sequence converges at $\omega$. This is the Kleene theorem.

## Exercises

**Exercise 2.1.** In Definition 2.1.3, show that a supremum of $S$ is necessarily unique if it exists. Show that it may not be unique if $(X, \sqsubseteq)$ is a preorder instead of a poset.

**Exercise 2.2.** Prove Proposition 2.1.4.

**Exercise 2.3.** Let $(L, \sqsubseteq_L)$ and $(M, \sqsubseteq_M)$ be complete lattices, and $f\colon L \to M$ be a monotone function. Show that the following inequalities hold, for each $S \subseteq L$.

$$f(\textstyle\bigsqcap S) \;\sqsubseteq\; \textstyle\bigsqcap f(S) \qquad f(\textstyle\bigsqcup S) \;\sqsupseteq\; \textstyle\bigsqcup f(S)$$

**Exercise 2.4.** Let $(X, \sqsubseteq_X)$ and $(Y, \sqsubseteq_Y)$ be posets, and $f\colon X \to Y$ be a function. Show that, if $f$ is $\bigsqcap$-preserving (or $\bigsqcup$-preserving), then it is monotone.

**Exercise 2.5.** Prove Proposition 2.1.9.

**Exercise 2.6.** In Example 2.1.12, Item 3, find a family $(S_i)_{i \in I}$ of open sets for which the infimum $\bigsqcap_{i \in I} S_i$ in $\mathcal{O}$ does not coincide with the set-theoretic intersection $\bigcap_{i \in I} S_i$. (Hint: such a family is not rare—it can be found for example in the set $\mathbb{R}$ of real numbers with the usual Euclidean topology.)

**Exercise 2.7.** In Definition 2.1.17, given $g \colon Y \to X$, show that its left adjoint is necessarily unique.

**Exercise 2.8.** Let $(L, \sqsubseteq)$ be a complete lattice, and $L' \subseteq L$ be a subset. Consider the poset $(L', \sqsubseteq)$ whose order is inherited from $L$; we are interested in whether $L'$ is a complete lattice or not.

1. Prove the following: for each $S \subseteq L'$, if its infimum $\bigsqcap_L S$ in $L$ happens to be in $L'$, then it is the infimum $\bigsqcap_{L'} S$ in $L'$.
2. Assume that both $(L, \sqsubseteq)$ and $(L', \sqsubseteq)$ are both complete lattices. Show that, for each $S \subseteq L'$, we necessarily have $\bigsqcap_{L'} S \sqsubseteq \bigsqcap_L S$.

**Exercise 2.9.** Let $f \colon X \to Y$ be a function. We define three functions

$$\begin{tikzcd} 2^X \arrow[r, bend left=40, "\exists_f"] \arrow[r, bend right=40, "\forall_f"'] \arrow[r, phantom, shift left=2.5ex, "\perp"] \arrow[r, phantom, shift right=2.5ex, "\perp"] & 2^Y \arrow[l, "f^{-1}"'] \end{tikzcd}$$

between the complete lattices $2^X$ and $2^Y$, by the following.

$$\begin{aligned} \exists_f(P) &\stackrel{\text{def}}{=} \{f(x) \mid x \in P\} \\ f^{-1}(Q) &\stackrel{\text{def}}{=} \{x \mid f(x) \in Q\} \\ \forall_f(P) &\stackrel{\text{def}}{=} \{y \in Y \mid \forall x \in X. \, (f(x) = y \text{ implies } x \in P)\,\} \end{aligned}$$

Show that, indeed, we have two adjuctions $\exists_f \dashv f^{-1}$ and $f^{-1} \dashv \forall_f$. It follows (from Proposition 2.1.18) that $\exists_f$ is $\bigsqcup$-preserving, $\forall_f$ is $\bigsqcap$-preserving, and $f^{-1}$ is both $\bigsqcup$- and $\bigsqcap$-preserving.

(Extensions of the above observations are made in Section 3.5.2 for a “non-deterministic function,” i.e. a binary relation $R \subseteq X \times Y$, instead of a (deterministic) function $f \colon X \to Y$.)

**Exercise 2.10.** Prove **??**.

**Exercise 2.11.** Prove **??**

**Exercise 2.12.** In **??**, we can use Theorem 2.1.16 for $\bigsqcap$-preserving $\Box$. Describe the left adjoint of $\Box$.

Dually, describe the right adjoint of ($\bigsqcup$-preserving) $\Diamond$.

(The former adjunction is studied in [9] in a general categorical setting. See also [10, Chapter 6].)

**Exercise 2.13.** Let $(L, \sqsubseteq)$ be a complete lattice. Let $S_i \subseteq L$ be a subset, for each index $i \in I$. Show that

$$\textstyle\bigsqcap_{i \in I} \left(\bigsqcap S_i\right) \;=\; \bigsqcap \left(\bigcup_{i \in I} S_i\right)$$

holds, where $\bigcup_{i \in I} S_i$ denotes the set-theoretic union. (Hint: use Proposition 2.1.4 and reduce the claim to a property of the meta-level universal quantification.)

Show, in particular, that

$$x \sqcap (\bigsqcap S) \;=\; \bigsqcap\{x \sqcap s \mid s \in S\}$$

holds for each $x \in L$ and $S \subseteq L$.

Show that dual holds, too. That is,

$$\bigsqcup_{i \in I} (\bigsqcup S_i) \;=\; \bigsqcup \left(\bigcup_{i \in I} S_i\right), \qquad x \sqcup (\bigsqcup S) \;=\; \bigsqcup\{x \sqcup s \mid s \in S\}.$$

# Chapter 3

# Safety and Reachability

Some leader

## 3.1 Safety and Reachability in Terms of Fixed Points

We organize basic safety and reachability problems as follows. Here, the intuition is that $i$ is an "initiality property" and $p$ is a "desired property." We discuss their concrete examples in Sections 3.2 and 3.3.

**Definition 3.1.1** ((co-)safety, (co-)reachability)**.** The terms *safety*, *reachability*, *co-safety*, and *co-reachability* properties refer to the (negations of) inequalities in the formats shown in the table below. Here $g\colon L \to L$ is a monotone map on a complete lattice $L$, and $i, p \in L$

Table 3.1: (Co-)safety, (co-)reachability properties

| **safety** | **co-safety** |
|---|---|
| $i \sqsubseteq \nu(p \sqcap g(\_))$ | $i \not\sqsubseteq \nu(p \sqcap g(\_))$ |
| **reachability** | **co-reachability** |
| $i \sqsubseteq \mu(p \sqcup g(\_))$ | $i \not\sqsubseteq \mu(p \sqcup g(\_))$ |

It is often stated that safety and reachability are the negations of each other. We do not take this view. Instead, we take a refined view and distinguish the negation of safety (co-safety) from reachability.

**Remark 3.1.2.** An informal remark that may be useful here is that, when we think of the "opposite" or "negation" of $x \sqsubseteq y$, we might be thinking of either $x \sqsupseteq y$ or $x \not\sqsubseteq y$. These two are different, unless $\sqsubseteq$ is the total order.

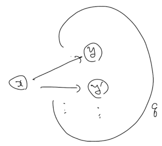
## 3.2 Transition Systems Examples

Using transition systems (Section 1.3), we present two families of examples of the four properties in Definition 3.1.1. The two families correspond to different choices of *(backward) predicate transformers*, namely $\mathsf{Bwd}^{\square}$ and $\mathsf{Bwd}^{\Diamond}$.

### 3.2.1 From the Backward $\Box$-Predicate Transformer $\mathsf{Bwd}^{\Box}$

**Definition 3.2.1** ($\mathsf{Bwd}^{\Box}$)**.** Let $\mathcal{S} = (X, R)$ be a transition system. We define the function $\mathsf{Bwd}^{\Box} \colon 2^X \to 2^X$—we call it the *backward* $\Box$*-predicate transformer* induced by $\mathcal{S}$—by the following. (Here we use the notational convention in Example 2.1.12.2, identifying predicates $X \to 2$ with subsets of $X$.)

$$\mathsf{Bwd}^{\Box} \;:\; 2^X \longrightarrow 2^X, \quad q \longmapsto \{\, x \in X \;\big|\; \forall y \in X.\, (x \to y \text{ implies } y \in q) \,\}.$$



The function $\mathsf{Bwd}^{\Box} \colon 2^X \to 2^X$ is easily shown to be monotone.

Let $g = \mathsf{Bwd}^{\Box}$, and let $i$ and $p$ denote the sets of *initial states* and *desired states*, respectively. Then the four properties in Definition 3.1.1 are interpreted as follows.

**Safety** "From each initial state $x \in i$, every (finite or infinite) path from $x$ stays within $p$ all the time."

Indeed, it is easy to see by induction on $n$ that

$$\begin{aligned} &\big(p \sqcap \mathsf{Bwd}^{\Box}(\_)\big)^n(\top) \\ &\quad = \{\text{states } x \text{ from which every path of length} \leq n \text{ stays in } p\}. \end{aligned}$$

refine the pointer

Now it is not hard to show that $p \sqcap \mathsf{Bwd}^{\Box}(\_) \colon L \to L$ is monotone and preserves the infimum of the Kleene sequence (Section 2.2.4). Therefore we have

$$\begin{aligned} \nu(p \sqcap \mathsf{Bwd}^{\Box}(\_)) &= \bigsqcap_{n<\omega} \big(p \sqcap \mathsf{Bwd}^{\Box}(\_)\big)^n(\top) \qquad \text{by the Kleene theorem} \\ &= \{\text{states } x \text{ from which every path stays in } p\}. \end{aligned}$$

**Co-Safety** "There exists an initial state $x \in i$ from which there is a path that reaches $X \setminus p$."

This is the negation of the safety property above. This might sound like a "reachability property" as commonly understood. But we insist that this co-safety property should be distinguished from a (proper) reachability property, presented in Section 3.2.2 later, where *every* $x \in i$ is required to reach $p$.

**Reachability** "For each initial state $x \in i$, every infinite path from $x$ reaches $p$ eventually."

This sounds different from a common "reachability property" for transition systems (the latter will appear later in Section 3.2.2). The current reachability property—it can be understood as a termination property under demonic nondeterminism, cf. Remark 3.2.3—requires that every path from $x$, if it has not yet visited $p$, will eventually do so or come to a state with no successors.

To prove that the above interpretation indeed coincides with the general fixed-point formulation $i \sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\Box}(\_))$ in Definition 3.1.1, it helps to employ some abstract machinery we introduce later. It is therefore deferred to Example 3.4.11.

**Co-Reachability** "There exists an initial state $x \in I$ that has an infinite path that never reaches $p$."

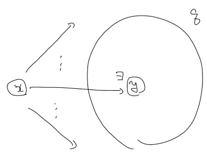
This is the negation of the above reachability property.

### 3.2.2 From the Backward $\Diamond$-Predicate Transformer $\mathsf{Bwd}^{\Diamond}$

For the other class of examples from transition systems, we use the following predicate transformer.

**Definition 3.2.2** ($\mathsf{Bwd}^{\Diamond}$)**.** Assume the setting of Definition 3.2.1. The *backward $\Diamond$-predicate transformer* induced by $\mathcal{S}$ is defined by

$$\mathsf{Bwd}^{\Diamond} \;:\; 2^X \longrightarrow 2^X, \quad q \longmapsto \bigl\{ x \in X \bigm| \exists y \in X.\, (y \in q \text{ and } x \to y) \bigr\}.$$



It is again easy to see that $\mathsf{Bwd}^{\Diamond}$ is monotone.

Let $g = \mathsf{Bwd}^{\Diamond}$, and let $i$ and $p$ be the same as in Section 3.2.1. The four problems in Definition 3.1.1 are interpreted as follows.

**Safety** "From each initial state $x \in i$, there is an infinite path that stays in $p$ all the time."

This is similar to the co-reachability property in Section 3.2.1 but the difference is that here we require "safety" for all initial states.

The proof that the above interpretation is correct is much like for the reachability property in Section 3.2.1. It is therefore deferred to Example 3.4.12.

**Co-Safety** "There is an initial state $x \in i$ from which every infinite path reaches $X \setminus p$ eventually."

**Reachability** "From each initial state $x \in i$, there is a path that reaches $p$ eventually."

This is a common reachability specification that ensures that every $x \in I$ is good in the sense of reaching $p$. To show that the above interpretation coincides with the fixed-point formulation $i \sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\Diamond}(\_))$, we can show by induction on $n$ that

$$\begin{aligned}&\bigl(p \sqcup \mathsf{Bwd}^{\Diamond}(\_)\bigr)^n(\bot)\\ &= \{x \mid x \text{ has a path of length} \le n \text{ that reaches } p\}.\end{aligned}$$

It is not hard to see that $p \sqcup \mathsf{Bwd}^{\Diamond}(\_)$ preserves the supermum of the Kleene sequence from $\bot$ (Section 2.2.4). Therefore we have

refine the pointer

$$\begin{aligned}\mu(p \sqcup \mathsf{Bwd}^{\Diamond}(\_)) &= \bigsqcup_{n<\omega} \bigl(p \sqcup \mathsf{Bwd}^{\Diamond}(\_)\bigr)^n(\bot) && \text{by the Kleene theorem}\\ &= \{\text{states } x \text{ with a path that reaches } p\}.\end{aligned}$$

**Co-Reachability** "There is an initial state $x \in i$ that has no path that reaches $p$ eventually."

**Remark 3.2.3.** The contrast between the two predicate transformers $\mathsf{Bwd}^{\Box}$ and $\mathsf{Bwd}^{\Diamond}$ in Sections 3.2.1 and 3.2.2 is observed in many contexts.

One is in *normal modal logic* (as suggested by the notations): $\mathsf{Bwd}^{\Box}$ uses the *box modality* $\Box\varphi$ in normal modal logic (all successors must satisfy $\varphi$), while $\mathsf{Bwd}^{\Diamond}$ uses the *diamond modality* $\Diamond\varphi$ (there must be a successor that satisfies $\varphi$). See e.g. [3]. Indeed, later in Chapter 9, we formalize a general notion of modality and derive predicate transformers systematically from modalities.

Another is the contrast between the *demonic* and *angelic* notions of nondeterminism. It appears in process theory, control theory, planning theory, etc.

- In demonic nondeterminism, the choice of a successor by the adversarial environment, and we want to ensure safety or reachability *no matter* what the environment does. This is modeled by $\mathsf{Bwd}^{\Box}$.
- In contrast, in angelic nondeterminism, the choice of successor by us on the system side, and we can steer the system so that it exhibits a desired safety or reachability property. This is modeled by $\mathsf{Bwd}^{\Diamond}$. In this setting, checking safety or reachability should desirably yield a *scheduler* as well—it tells which successor should be chosen.

We note that there is a well-known duality between $\Box$ and $\Diamond$, and thus between $\mathsf{Bwd}^{\Box}$ and $\mathsf{Bwd}^{\Diamond}$. It will be formalized in Proposition 3.4.10.

## 3.3 Markov Chains Examples

We present some probabilistic examples here. They are defined for Markov chains (Definition 1.3.2).

### 3.3.1 The Backward Average Predicate Transformer $\mathsf{Bwd}^{\mathsf{Av}}$

The following construct transforms, in a backward manner, a fuzzy predicate (Definition 1.3.4) to another along a transition kernel $\delta$. It has been studied in the context of semantics and verification of probabilistic programs [**?**, **?**, **?**] and in the context of dynamic programming [**?**]. In the latter, the construct is often called the *Bellman operator*.

**Definition 3.3.1** ($\mathsf{Bwd}^{\mathsf{Av}}$)**.** Let $\mathcal{S} = (X, \delta)$ be a Markov chain. We define the function $\mathsf{Bwd}^{\mathsf{Av}} \colon [0,1]^X \to [0,1]^X$—we call it the *backward average predicate transformer* induced by $\mathcal{S}$—by the following.

$$\begin{aligned} \mathsf{Bwd}^{\mathsf{Av}} \;&:\; [0,1]^X \longrightarrow [0,1]^X, \quad q \longmapsto \mathsf{Bwd}^{\mathsf{Av}}(q), \\ \mathsf{Bwd}^{\mathsf{Av}}(q)(x) \;&\overset{\text{def}}{=}\; \sum_{x' \in X} \delta(x)(x') \cdot q(x'). \end{aligned}$$

The fuzzy predicate $\mathsf{Bwd}^{\mathsf{Av}}(q)$ assigns, to each state $x \in X$, the average value of $q$ in its successors $x'$. The average is weighted by the transition probabilities $\delta(x)(x')$.

Question: any predicate tranformer for almost-sure reachability? I first thought that $\mathsf{Bwd}^{\Box}$ might work but it does not since it fails for the "stay here with 1/2, terminate with 1/2" example. The technique is probably more related to the fairness theorem $\to$ find a good lattice-theoretic abstraction of fairness?

### 3.3.2 Safety and Reachability

We shall look at the instance of Definition 3.1.1, where $g = \mathsf{Bwd}^{\mathsf{Av}} \colon [0,1]^X \to [0,1]^X$ is from Definition 3.3.1 and $i, p \in [0,1]^X$ are fuzzy predicates. Furthermore, in order to avoid measure-theoretic complexities, we restrict $p$ to be *sharp* in the following sense.

**Definition 3.3.2** (sharp predicate, $\chi_S$)**.** A fuzzy predicate $p \colon X \to [0,1]$ is *sharp* if it takes 0 or 1 as its values, that is, there is a predicate $\widetilde{p} \colon X \to 2$ such that

$$p \;=\; \bigl(X \xrightarrow{\widetilde{p}} 2 \;=\; \{0,1\} \longrightarrow [0,1]\bigr).$$

A sharp predicate is therefore idenfied with a subset $S \subseteq X$ of $X$. We write $\chi_S \colon X \to [0,1]$ for this predicate. Concretely, $\chi_S$ is the *characteristic function* of $S$: $\chi_S(x)$ is 1 if $x \in S$, and is 0 otherwise.

In what follows, we assume that $p = \chi_P$, i.e. that it is a sharp predicate with a *safe/target* set $P \subseteq X$.

1) Write the measure-theoretic general version, and 2) put a pointer to it

**Safety** (Assuming the sharpness of $p = \chi_P$) "for each state $x \in X$, the probability that a path from $x$ stays in $P$ all the time is at least $i(x) \in [0,1]$."

The coincidence of this interpretation and the general formulation $i \sqsubseteq \nu(p \sqcap g(\_))$ is established in Example 3.4.14, after we develop some abstract machinery.

**Co-Safety** (Assuming the sharpness of $p = \chi_P$) "there is a state $x \in X$ from which the probability of staying in $P$ all the time is strictly smaller than $i(x)$."

This is clearly equivalent to the following: "there is a state $x \in X$ from which the probability of reaching $P$ some time is strictly larger than $1 - i(x)$."

**Reachability** (Assuming the sharpness of $p = \chi_P$) "for each state $x \in X$, the probability that a path from $x$ reaches $P$ is at least $i(x) \in [0,1]$."

To see the correctness of this interpretation, we observe that

$$\begin{aligned}&\bigl(\chi_P \sqcup \mathsf{Bwd}^{\mathsf{Av}}(\_)\bigr)^n(\bot)\\ =\;& \mathsf{Pr}(P \text{ is reached from } x \text{ with a path of length} \le n).\end{aligned}$$

This is easily proved by induction on $n$.

It is not hard to see that $\chi_P \sqcup \mathsf{Bwd}^{\mathsf{Av}}(\_)$ preserves the supermum of the Kleene sequence from $\bot$ (Section 2.2.4). Therefore we have refine the pointer

$$\mu\bigl(\chi_P \sqcup \mathsf{Bwd}^{\mathsf{Av}}(\_)\bigr) \;=\; \bigsqcup_{n<\omega} \bigl(\chi_P \sqcup \mathsf{Bwd}^{\mathsf{Av}}(\_)\bigr)^n(\bot) \qquad \text{by the Kleene theorem}$$

$$= \mathsf{Pr}(P \text{ is reached from } x \text{ with some path}).$$

We used an obvious fact that, for a path to reach $P$, it has to do so within $n$ steps for some finite $n$.

**Co-Reachability** (Assuming the sharpness of $p = \chi_P$) "there is a state $x \in X$ from which the probability of reaching $P$ is strictly smaller than $i(x)$."

This is clearly equivalent to the following: "there is a state $x \in X$ such that, the probability of staying outside $P$ all the time is strictly larger than $1 - i(x)$."

## 3.4 Involutions in Safety and Reachability

In Sections 3.4 and 3.5, we exhibit two abstract techniques for transforming safety and reachability properties into equivalent properties formulated in terms of fixed points.

The first one, presented here, is intuitively about "negation." Such intuitions have already appeard in the examples in Sections 3.2 and 3.3.

**Remark 3.4.1.** Another advanced transformation, from reachability to a (non-equivalent) fixed-point property, is by *ranking*. It is presented later in Chapter 10.

### 3.4.1 Involution

**Definition 3.4.2** (the opposite poset $L^{\mathrm{op}}$)**.** Let $L = (L, \sqsubseteq_L)$ be a poset. The *opposite* poset $L^{\mathrm{op}}$ of $L$ is defined by $L^{\mathrm{op}} \overset{\mathrm{def}}{=} (L, \sqsupseteq_L)$. That is,

$$\frac{x \sqsubseteq y \quad \text{in } L^{\mathrm{op}}}{\overline{\;y \sqsubseteq x \quad \text{in } L\;}}\ .$$

It is clear that $(L^{\mathrm{op}})^{\mathrm{op}} = L$.

The notion of complete lattice comes with a symmetry in the following sense.

**Lemma 3.4.3.** *Let $L$ be a complete lattice. Then so is $L^{\mathrm{op}}$, where the infimums and supremums are given by the supremums and infimums in $L$, respectively.* □

**Definition 3.4.4** (involution)**.** Let $L$ be a poset. An *involution* is a monotone function $\neg\colon L^{\mathrm{op}} \to L$ such that $\neg \circ \neg = \mathrm{id}_L$, as in

$$\begin{tikzcd} L \arrow[r, "\neg"] \arrow[dr, equal] & L^{\mathrm{op}} \arrow[d, "\neg"] \\ & L. \end{tikzcd}$$

Some remarks are in order. Firstly, a monotone function of the type $f\colon L_1^{\mathrm{op}} \to L^2$ "reverses" the order, in the sense that

$$x \sqsubseteq y \text{ in } L_1 \quad \text{implies} \quad f(y) \sqsubseteq f(x) \text{ in } L_2.$$

Such a function is also called an *antitone function* from $L_1$ to $L_2$.

Secondly, in Definition 3.4.4, note that the two functions $\neg\colon L \to L^{\mathrm{op}}$ and $\neg\colon L^{\mathrm{op}} \to L$ refer to the same set-theoretic function (they carry each $x$ to the same $\neg(x)$). Endowing it with two types ($L \to L^{\mathrm{op}}$ and $L^{\mathrm{op}} \to L$) is justified by the following observation.

**Lemma 3.4.5.** *Let $f\colon L_1 \to L_2$ be a monotone function. Then it is also a monotone function $f\colon L_1^{\mathrm{op}} \to L_2^{\mathrm{op}}$ with the reversed orders.* □

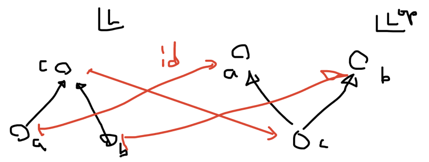
The latter $f$ is often denoted by $f^{\mathrm{op}}$ for distinction.

Thirdly, since $\neg$ is converse to itself as a (set-theoretic) function, it is bijective. Therefore finding an involution on $L$ is to find that $L$ is the same shape upside down.

**Example 3.4.6.**

1. Recall that 2 designates the two-element complete lattice $0 \sqsubseteq 1$. The Boolean negation function $\neg\colon 2^{\mathrm{op}} \to 2$, given by $\neg(x) \overset{\mathrm{def}}{=} 1 - x$, is an involution.
2. Over the unit interval lattice $[0,1]$ (with the usual order between real numbers), the function $\neg(x) \overset{\mathrm{def}}{=} 1 - x$ is an involution.
3. These two involutions can be lifted to (fuzzy) predicates: we obtain two involutions $\neg\colon (2^X)^{\mathrm{op}} \to 2^X$ and $\neg\colon ([0,1]^X)^{\mathrm{op}} \to [0,1]^X$, defined by $(\neg p)(x) \overset{\mathrm{def}}{=} 1 - p(x)$.

   In the case of $\neg\colon (2^X)^{\mathrm{op}} \to 2^X$, through the identification of predicates with subsets, $\neg$ is nothing but the *complementation* $S \mapsto X \setminus S$.
4. One might wonder why the identity function shown below is not an involution. It is not, since it is an *antitone* (instead of monotone) function from $L^{\mathrm{op}}$ to $L$.




The following rule-based presentation for involutions is useful. Recall that double lines mean implications both ways (if and only if).

**Lemma 3.4.7.** *Let $L$ be a poset and $L\colon L^{\mathrm{op}} \to L$ be an involution. Then we have the following rules valid.*

$$\frac{x \sqsubseteq y}{\overline{\neg y \sqsubseteq \neg x}} \quad \text{and thus, by } \neg \circ \neg = \mathrm{id}, \quad \frac{x \sqsubseteq \neg y}{\overline{y \sqsubseteq \neg x}}, \quad \frac{\neg x \sqsubseteq y}{\overline{\neg y \sqsubseteq x}}, \quad \text{etc.}$$

*Here $x, y \in L$, and the order $\sqsubseteq$ is in $L$ (not in $L^{\mathrm{op}}$).* □

### 3.4.2 Involutions and Fixed Points

Involutions turn many constructs upside down, including fixed points.

**Lemma 3.4.8.** *Let $L$ be a complete lattice, and $\neg\colon L^{\mathrm{op}} \to L$ be an involution.*

1. *For $S \subseteq L$, we have*

$$\neg(\textstyle\bigsqcup S) = \textstyle\bigsqcap\{\neg x \mid x \in S\} \quad \text{and} \quad \neg(\textstyle\bigsqcap S) = \textstyle\bigsqcup\{\neg x \mid x \in S\}.$$

2. *Let $f\colon L \to L$ be a monotone function. We define the* dual *of $f$ to be the function*

$$f^{\neg\neg} \stackrel{\text{def}}{=} \neg \circ f \circ \neg = \left( L^{\mathrm{op}} \xrightarrow{\neg} L \xrightarrow{f} L \xrightarrow{\neg} L^{\mathrm{op}} \right).$$

*Then we have*

$$\neg(\mu f) = \nu(f^{\neg\neg}) \quad \text{and} \quad \neg(\nu f) = \mu(f^{\neg\neg}).$$

*Proof.* For Item 1, we exploit the characterization of $\bigsqcup, \bigsqcap$ in Proposition 2.1.4. For example, the following proves the equality $\neg(\bigsqcup S) = \bigsqcap\{\neg x \mid x \in S\}$.

$$\cfrac{\cfrac{z \sqsubseteq \neg y \quad \text{for each } y \in S}{\overline{y \sqsubseteq \neg z \quad \text{for each } y \in S}}\ \text{Lemma 3.4.7}}{\cfrac{\bigsqcup S \sqsubseteq \neg z}{\overline{z \sqsubseteq \neg(\bigsqcup S)}}\ \text{Lemma 3.4.7}}$$

For Item 2, we prove that first equality; the second is similary. Let us first show that $\neg(\mu f)$ is a fixed point of $f^{\neg\neg}$.

$$f^{\neg\neg}(\neg(\mu f)) = \neg(f(\neg\neg(\mu f))) = \neg(f(\mu f)) = \neg(\mu f) \qquad \text{since } f(\mu f) = \mu f.$$

Now we prove that it is the greatest. Assume $x = f^{\neg\neg}(x)$ (thus in particular $x \sqsubseteq f^{\neg\neg}(x)$); we aim to show that $x \sqsubseteq \neg(\mu f)$.

$$\begin{array}{lll} & x \sqsubseteq f^{\neg\neg}(x) = \neg f(\neg x) & \\ \Longleftrightarrow & f(\neg x) \sqsubseteq \neg x & \text{by Lemma 3.4.7} \\ \Longrightarrow & \mu f \sqsubseteq \neg x & \text{by Theorem 2.2.1} \\ \Longrightarrow & x \sqsubseteq \neg(\mu f) & \text{by Lemma 3.4.7.} \end{array}$$

Therefore $\neg(\mu f)$ coincides with the greatest fixed point $\nu(f^{\neg\neg})$. □

Using the above translations, we obtain the following dual presentation of Table 3.1.

**Proposition 3.4.9** (safety and reachability via involutions)**.** *In the setting of Definition 3.1.1, assume further that $\neg\colon L^{\mathrm{op}} \to L$ is an involution. The following properties are equivalent to those in Table 3.1.*

Table 3.2: (Co-)safety, (co-)reachability properties, via involution

| **safety** | **co-safety** |
|---|---|
| $\mu\big(\neg p \sqcup g^{\neg\neg}(\_)\big) \sqsubseteq \neg i$ | $\mu\big(\neg p \sqcup g^{\neg\neg}(\_)\big) \not\sqsubseteq \neg i$ |
| **reachability** | **co-reachability** |
| $\nu\big(\neg p \sqcap g^{\neg\neg}(\_)\big) \sqsubseteq \neg i$ | $\nu\big(\neg p \sqcap g^{\neg\neg}(\_)\big) \not\sqsubseteq \neg i$ |

### 3.4.3 Transition System Examples

We start with the following basic observation. This "de Morgan-style" duality between □ and ◇ is well-known.

**Proposition 3.4.10.** *For the predicate transformers* $\mathsf{Bwd}^{\Box}$ *and* $\mathsf{Bwd}^{\Diamond}$ *from Definition 3.2.1 and 3.2.2, we have*

$$(\mathsf{Bwd}^{\Box})^{\neg\neg} \;=\; \mathsf{Bwd}^{\Diamond} \quad \textit{and} \quad (\mathsf{Bwd}^{\Diamond})^{\neg\neg} \;=\; \mathsf{Bwd}^{\Box}. \qquad \Box$$

The combination of Propositions 3.4.9 and 3.4.10 eases our analysis of some safety/reachability instances from Sections 3.2.1 and 3.2.2. We present some proofs that we left out there.

**Example 3.4.11** (reachability for $\mathsf{Bwd}^{\Box}$)**.** This is an example from Section 3.2.1. We shall show that the property $i \sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\Box}(\_))$ indeed means "for each initial state $x \in i$, every infinite path from $x$ reaches $p$ eventually."

We let

$$p_\infty \;\stackrel{\text{def}}{=}\; \{x \mid \text{every infinite path from } x \text{ visits } p\} \tag{3.1}$$

and prove

$$p_\infty \;=\; \mu(p \sqcup \mathsf{Bwd}^{\Box}(\_)).$$

The direction $\sqsupseteq$ is easy: $p_\infty$ is clearly a prefixed point of $p \sqcup \mathsf{Bwd}^{\Box}(\_)$ and we can use the Knaster–Tarski theorem (Corollary 2.2.7).

To show the direction $\sqsubseteq$, we use the involution $\neg$ on $2^X$ from Example 3.4.6. By Lemma 3.4.7, it suffices to show

$$\neg\big(\,\mu(p \sqcup \mathsf{Bwd}^{\Box}(\_))\,\big) \;\sqsubseteq\; \neg p_\infty.$$

By Lemma 3.4.8 and Proposition 3.4.10, it suffices to show

$$\nu\big(\neg p \sqcap \mathsf{Bwd}^{\Diamond}(\_)\big) \;\sqsubseteq\; \neg p_\infty.$$

We shall write $p_\nu \stackrel{\text{def}}{=} \nu\big(\neg p \sqcap \mathsf{Bwd}^{\Diamond}(\_)\big)$ for the left-hand side. By (3.1), we have

$$\neg p_\infty \;=\; \{x \mid x \text{ has an infinite path from it that never visits } p\}.$$

Now let $x \in \nu\big(\neg p \sqcap \mathsf{Bwd}^{\Diamond}(\_)\big)$. By expanding the fixed point once, we have $x \in \neg p \sqcap \mathsf{Bwd}^{\Diamond}(p_\nu)$, and thus 1) $x \notin p$ and 2) there is a successor $x_1$ of $x$ that belongs to $p_\nu$. We can continue this operation to find an infinite path $x = x_0 \to x_1 \to \cdots$ that stays outside $p$. This establishes $x \in \neg p_\infty$ and concludes the proof.

**Example 3.4.12** (safety for $\mathsf{Bwd}^{\Diamond}$)**.** This is an example from Section 3.2.2. Let us prove that the concrete property

$$p_\infty \;\stackrel{\text{def}}{=}\; \{x \in X \mid x \text{ has an infinite path that stays in } p\}$$

coincides with $\nu(p \sqcap \mathsf{Bwd}^{\Diamond}(\_))$. The proof goes much like in Example 3.4.11. Indeed,

- $p_\infty$ is a postfixed point of $p \sqcap \mathsf{Bwd}^{\Box}(\_)$, which establishes $p_\infty \sqsubseteq \nu(p \sqcap \mathsf{Bwd}^{\Diamond}(\_))$ by the Knaster–Tarski theorem; and
- any element $x \in \nu(p \sqcap \mathsf{Bwd}^{\Diamond}(\_))$ yields an infinite path $x = x_0 \to x_1 \to \cdots$ that stays in $p$, by unfolding the fixed point. This proves that $x \in p_\infty$.

### 3.4.4 Markov Chain Examples

We start with the following observation: $\mathsf{Bwd}^{\mathsf{Av}}$ is dual to itself.

**Proposition 3.4.13.** *For the predicate transformer* $\mathsf{Bwd}^{\mathsf{Av}}$ *from Definition 3.3.1, we have*

$$(\mathsf{Bwd}^{\mathsf{Av}})^{\neg\neg} \;=\; \mathsf{Bwd}^{\mathsf{Av}}. \qquad \square$$

Therefore, to the safety and reachability properties in Table 3.1 with $g = \mathsf{Bwd}^{\mathsf{Av}}$, the dual properties in Table 3.2 are all equivalent, with $g^{\neg\neg} = \mathsf{Bwd}^{\mathsf{Av}}$. We use this to confirm the interpretation of the safety property for $\mathsf{Bwd}^{\mathsf{Av}}$.

**Example 3.4.14** (safety for $\mathsf{Bwd}^{\mathsf{Av}}$)**.** We consider the safety property in Section 3.4.4. Let $p_\infty : X \to [0,1]$ be the fuzzy predicate defined by

$$p_\infty(x) \;\stackrel{\text{def}}{=}\; \mathsf{Pr}\bigl(\text{a path from } x \text{ stays in } P \text{ all the time}\bigr).$$

Our goal is to show

$$p_\infty \;=\; \nu(p \sqcap \mathsf{Bwd}^{\mathsf{Av}}(\_)).$$

Using the involution $\neg\colon [0,1]^X \to [0,1]^X$ in Example 3.4.6 and then using Lemma 3.4.8, this is equivalent to showing

$$\neg p_\infty \;=\; \mu(\neg p \sqcup \mathsf{Bwd}^{\mathsf{Av}}(\_)).$$

We argue as follows.

$$\begin{aligned}
&(\neg p_\infty)(x)\\
&= \mathsf{Pr}(\text{a path from } x \text{ reaches } X \setminus P \text{ eventually})\\
&= \textstyle\bigsqcup_{n<\omega} \mathsf{Pr}(\text{a path from } x \text{ reaches } X \setminus P \text{ within } n \text{ steps})\\
&= \Bigl(\textstyle\bigsqcup_{n<\omega} \bigl(\neg p \sqcup \mathsf{Bwd}^{\mathsf{Av}}(_)\bigr)^n(\bot)\Bigr)(x) \qquad (*)\\
&= \bigl(\mu(\neg p \sqcup \mathsf{Bwd}^{\mathsf{Av}}(_))\bigr)(x).
\end{aligned}$$

For the last equality we used the Kleene theorem (and that $\mathsf{Bwd}^{\mathsf{Av}}$ is $\bigsqcup$-preserving); for (*) we used that

$$\mathsf{Pr}(\text{a path from } x \text{ reaches } X \setminus P \text{ within } n \text{ steps}) \;=\; \Bigl(\bigl(\neg p \sqcup \mathsf{Bwd}^{\mathsf{Av}}(\_)\bigr)^n(\bot)\Bigr)(x) \qquad \text{for each } x \in X,$$

which we can show easily by induction.

## 3.5 Adjoints in Safety

Cite Bart's Galois connection paper

### 3.5.1 General Translation

It is observed in [11] that the safety problem, as formulated as in Definition 3.1.1, allows another characterization when $f$ has a right adjoint (cf. Definition 2.1.17).

**Proposition 3.5.1** (safety and adjoints)**.** *Let $L$ be a complete lattice, $f, g \colon L \to L$ be monotone maps, and $i, p \in L$. If $f \dashv g$ as in*

$$\begin{tikzcd} L \arrow[r, bend left, "f"] \arrow[r, phantom, "\perp"] & L, \arrow[l, bend left, "g"] \end{tikzcd}$$

*then we have*

$$\mu(i \sqcup f(\_)) \;\sqsubseteq\; p \qquad\Longleftrightarrow\qquad i \;\sqsubseteq\; \nu(p \sqcap g(\_)).$$

*Proof.* We reason as follows.

$$\begin{array}{lll}
\multicolumn{3}{l}{\mu(i \sqcup f(_)) \;\sqsubseteq\; p} \\
\Longleftrightarrow & i \sqcup f(y) \sqsubseteq y \text{ and } y \sqsubseteq p \quad \text{for some } y \in L & \text{by Corollary 2.2.8} \\
\overset{(*)}{\Longleftrightarrow} & i \sqsubseteq y,\ f(y) \sqsubseteq y, \text{ and } y \sqsubseteq p \quad \text{for some } y \in L & \\
\Longleftrightarrow & i \sqsubseteq y,\ y \sqsubseteq g(y), \text{ and } y \sqsubseteq p \quad \text{for some } y \in L & \text{by } f \dashv g \\
\overset{(*)}{\Longleftrightarrow} & i \sqsubseteq y \text{ and } y \sqsubseteq p \sqcap g(y) \quad \text{for some } y \in L & \\
\Longleftrightarrow & i \;\sqsubseteq\; \nu(p \sqcap g(_)) & \text{by Corollary 2.2.8.}
\end{array}$$

In $(*)$, we used the universality of $\sqcup, \sqcap$ (Proposition 2.1.4).

**Remark 3.5.2.** The last observation in Proposition 3.5.1 is exploited in [11] for devising an IC3/PDR-type algorithm, in which a positive witness (for verification) and a negative witness (for refutation) are simultaneously searched for. See **??** for more.

?

**Remark 3.5.3.** One might wonder if a characterization much like Proposition 3.5.1 is possible for reachability as well. However, the proof is built on a delicate combination of $\mu$ vs. $\nu$, $\sqcup$ vs. $\sqcap$, and left vs. right adjoints. It does not seem to generalize easily to other combinations.

### 3.5.2 Transition System Examples

For the two predicate transformers $\mathsf{Bwd}^{\Box}, \mathsf{Bwd}^{\Diamond}$ for transition systems (Section 3.2), we have the following adjunctions.

**Proposition 3.5.4.** *Let $\mathcal{S} = (X, R)$ be a transition system, much like in Definition 3.2.1 and 3.2.2. We define the* forward $\Box$- and $\Diamond$-predicate transformers *$\mathsf{Fwd}^{\Box}, \mathsf{Fwd}^{\Diamond} \colon 2^X \to 2^X$ as follows.*

$$\begin{array}{lll}
\mathsf{Fwd}^{\Box} \;:\; 2^X \longrightarrow 2^X, & p \longmapsto \bigl\{\, y \in X \bigm| \exists x \in X.\, (x \to y \textit{ and } x \in p) \,\bigr\}, \\
\mathsf{Fwd}^{\Diamond} \;:\; 2^X \longrightarrow 2^X, & p \longmapsto \bigl\{\, y \in X \bigm| \forall x \in X.\, (x \to y \textit{ implies } x \in p) \,\bigr\}.
\end{array}$$

*Then we obtain the following adjunctions: $\mathsf{Fwd}^{\Box} \dashv \mathsf{Bwd}^{\Box}$ and $\mathsf{Bwd}^{\Diamond} \dashv \mathsf{Fwd}^{\Diamond}$, as in*

$$\begin{tikzcd} 2^X \arrow[r, bend left, "\mathsf{Fwd}^{\Box}"] \arrow[r, phantom, "\perp"] & 2^X, \arrow[l, bend left, "\mathsf{Bwd}^{\Box}"] \end{tikzcd} \qquad \begin{tikzcd} 2^X \arrow[r, bend left, "\mathsf{Fwd}^{\Diamond}"] \arrow[r, phantom, "\top"] & 2^X. \arrow[l, bend left, "\mathsf{Bwd}^{\Diamond}"] \end{tikzcd}$$

Recall that $x \to y$ denotes $(x, y) \in R$ (Definition 1.2.1). Therefore these predicate transformers are indeed working in a forward manner, transforming a predicate $p$ on predecessors $x$ into one on successors $y$. This observation can also be seen as an extension of the adjunctions $\exists_f \dashv f^{-1} \dashv \forall_f$ induced by a (deterministic) function $f$, the adjunctions central in Lawvere's categorical modeling of quantifiers. See Exercise 2.9.

Show one

*Proof.* Easy, left as an exercise.

What about other adjunctions for $\mathsf{Bwd}^{\Box}, \mathsf{Bwd}^{\Diamond}$? We can show that they do not exist. Here we use Proposition 2.1.18.

**Lemma 3.5.5.** *Assume the setting of Proposition 3.5.4.*

1. $\mathsf{Bwd}^{\Box}$ *has no right adjoint, since it is not* $\bigsqcup$*-preserving. For a concrete example, consider a transition system* $x_0 \to x_1, x_0 \to x_2$*, and predicates* $q_1 = \{x_1\}, q_2 = \{x_2\}$.
2. $\mathsf{Bwd}^{\Diamond}$ *has no left adjoint, since it is not* $\bigsqcap$*-preserving. For a concrete example, we can use the same as above.* □

We also note the following easy observation: a de Morgan-style duality.

**Lemma 3.5.6.** *In the setting of Proposition 3.5.4, we have*

$$(\mathsf{Fwd}^{\Box})^{\neg\neg} = \mathsf{Fwd}^{\Diamond} \quad \textit{and} \quad (\mathsf{Fwd}^{\Diamond})^{\neg\neg} = \mathsf{Fwd}^{\Box}. \qquad \Box$$

Comparing Proposition 3.5.4 with the formulations and characterizations of safety and reachability (Tables 3.1 and 3.2), we find the following instances of the safety translation lemma (Proposition 3.5.1).

**Proposition 3.5.7.** *In the setting of Proposition 3.5.4, we have the following equivalences around the safety property with respect to* $\mathsf{Bwd}^{\Box}$*, see Section 3.2.1.*

$$\begin{tikzcd}
i \sqsubseteq \nu(p \sqcap \mathsf{Bwd}^{\Box}(_)) \arrow[r, Leftrightarrow, "\textit{Proposition 3.5.1}"] \arrow[d, Leftrightarrow, "\textit{Proposition 3.4.9}"'] & \mu(i \sqcup \mathsf{Fwd}^{\Box}(_)) \sqsubseteq p \arrow[d, Leftrightarrow, "\textit{Proposition 3.4.9}"] \\
\mu(\neg p \sqcup \mathsf{Bwd}^{\Diamond}(_)) \sqsubseteq \neg i \arrow[r, Leftrightarrow, "\textit{Proposition 3.5.1}"'] & \neg p \sqsubseteq \nu(\neg i \sqcap \mathsf{Fwd}^{\Diamond}(_))
\end{tikzcd}$$

### 3.5.3 Markov Chain Examples

In the case of Markov chains, it turns out that Proposition 3.5.1 is useless per se. This is because of the following observation.

**Lemma 3.5.8.** *Let* $\mathcal{S} = (X, \delta)$ *be a Markov chain, as in Definition 3.3.1. The predicate transformer* $\mathsf{Bwd}^{\mathsf{Av}}$ *has no left or right adjoint.*

*Proof.* In view of Proposition 2.1.18, it suffices to show that $\mathsf{Bwd}^{\mathsf{Av}}$ does not preserve $\bigsqcup$ or $\bigsqcap$.

To show this, the following minimal example will do. Let $X = \{x_0, x_1, x_2\}$ be the state space with $\delta(x_0)(x_1) = \delta(x_0)(x_2) = 1/2$. Consider the fuzzy predicates $q, q' \in [0, 1]^X$, given by $q(x_1) = 1, q(x_2) = 0$ and $q'(x_1) = 0, q'(x_2) = 1$. Then the supremum $q \sqcup q'$ assigns 1 to both $x_1$ and $x_2$; this yields $\big(\mathsf{Bwd}^{\mathsf{Av}}(q \sqcup q')\big)(x_0) = 1$. However, $\big(\mathsf{Bwd}^{\mathsf{Av}}(q)\big)(x_0) = 1\big(\mathsf{Bwd}^{\mathsf{Av}}(q')\big)(x_0) = 1/2$, showing that $\mathsf{Bwd}^{\mathsf{Av}}$ does not preserve $\bigsqcup$. The same example works for $\bigsqcap$. □

The above failure of $\bigsqcup, \bigsqcap$-preservation can be attributed to the following observation:

$$\max\{a_0, a_1\} + \max\{b_0, b_1\} \;\neq\; \max\{a_0 + b_0, a_1 + b_1\}.$$

Indeed, the left-hand side is in general above the right-hand side, since the latter misses some combinations such as $a_0 + b_1$.

A countermeasure is introduced in [11]:

$$\begin{tikzcd}
L^{\downarrow} \arrow[rr, dashed, "f^{\downarrow}"] \arrow[rr, phantom, "\top"] & & L^{\downarrow} \arrow[ll, bend left, "\exists g"] \\
L \arrow[u, tail, "(_)^{\downarrow}"] \arrow[rr, "f"'] & & L. \arrow[u, tail, "(_)^{\downarrow}"']
\end{tikzcd} \tag{3.2}$$

Here,

– we extend a complete lattice $L$ and embed it in the *complete lattice of lowersets* $L^{\downarrow}$.
– The embedding $L \rightarrowtail L^{\downarrow}$ is so-called the *free* $\bigsqcup$*-completion*: it 1) equips all supremums (although we assumed already that $L$ has them), and 2) lifts any monotone function $h\colon L \to L'$, with the codomain $L'$ equipped with all supremums, to a $\bigsqcup$-preserving one.

$$\begin{tikzcd}
L^{\downarrow} \arrow[dr, dashed, "\widetilde{h}\ (\bigsqcup\text{-pres.})"] & \\
L \arrow[u, tail, "(_)^{\downarrow}"] \arrow[r, "h"'] & L'
\end{tikzcd}$$

– We apply the above universality of $L \rightarrowtail L^{\downarrow}$ to the function $L \xrightarrow{f} L \rightarrowtail L^{\downarrow}$ (bottom-left to bottom-right to top-right in (3.2)), obtaining a $\bigsqcup$-preserving function $f^{\downarrow}\colon L^{\downarrow} \to L^{\downarrow}$.
– Since $f^{\downarrow}$ is $\bigsqcup$-preserving, by Theorem 2.1.16, we obtain the right adjoint $g$ to $f^{\downarrow}$.

In fact, this countermeasure is an instance of a well-known categorical construction, namely the *Yoneda embedding* as a free cocompletion. (Here the Yoneda embedding can take the form $\mathbf{y}\colon L \to 2^{L^{\mathrm{op}}}$, instead of $L \to \mathbf{Set}^{L^{\mathrm{op}}}$, since $L$ is a poset (i.e. 2-enriched). One can easily show that $2^{L^{\mathrm{op}}} \cong L^{\downarrow}$.) We revisit this theory in Chapter 5.

## 3.6 Summary: Problem Formulations and Characterizations

We summarize the problem formulations and their characterizations via various translations. We also present their instances for transition systems and Markov chains.

A summary of the general picture is shown in Table 3.3. Here "if $\neg$" designates the equivalences in the presence of an involution (Table 3.2). Additionally, safety allows an adjoint presentation; this comes from Section 3.5.1.

Table 3.3: General (co-)safety, (co-)reachability properties, summary

| | **safety** | | **co-safety** |
|---|---|---|---|
| | $i \sqsubseteq \nu(p \sqcap g(\_))$ | | $i \not\sqsubseteq \nu(p \sqcap g(\_))$ |
| $\stackrel{\text{if } \neg}{\Longleftrightarrow}$ | $\mu\big(\neg p \sqcup g^{\neg\neg}(\_)\big) \sqsubseteq \neg i$ | $\stackrel{\text{if } \neg}{\Longleftrightarrow}$ | $\mu\big(\neg p \sqcup g^{\neg\neg}(\_)\big) \not\sqsubseteq \neg i$ |
| $\stackrel{\text{if } f \dashv g}{\Longleftrightarrow}$ | $\mu(i \sqcup f(\_)) \sqsubseteq p$ | $\stackrel{\text{if } f \dashv g}{\Longleftrightarrow}$ | $\mu(i \sqcup f(\_)) \not\sqsubseteq p$ |
| | **reachability** | | **co-reachability** |
| | $i \sqsubseteq \mu(p \sqcup g(\_))$ | | $i \not\sqsubseteq \mu(p \sqcup g(\_))$ |
| $\stackrel{\text{if } \neg}{\Longleftrightarrow}$ | $\nu\big(\neg p \sqcap g^{\neg\neg}(\_)\big) \sqsubseteq \neg i$ | $\stackrel{\text{if } \neg}{\Longleftrightarrow}$ | $\nu\big(\neg p \sqcap g^{\neg\neg}(\_)\big) \not\sqsubseteq \neg i$ |

### 3.6.1 For Transition Systems

For transition systems, we have two backward predicate transformers of interest ($\mathsf{Bwd}^{\square}, \mathsf{Bwd}^{\diamond}$), and the overall pictures for them are presented in Tables 3.4 and 3.5. Note that the adjoint translation only works for $\mathsf{Bwd}^{\square}$.

### 3.6.2 For Markov Chains

For Markov chains, the two backward predicate transformer of our interest ($\mathsf{Bwd}^{\mathsf{Av}}$) is self-dual (Proposition 3.4.13) and has no left or right adjoint (Lemma 3.5.8), so the overall picture is simple. Its summary is shown in Table 3.6

## 3.7 Algorithms and Reasoning Principles

We have obtained fixed point characterizations of safety and reachability properties in Tables 3.3 to 3.6. All of them are about under- or over-approximating an lfp or gfp, thus the reasoning principles in Corollary 2.2.8—from the Knaster–Tarski and Cousot–Cousot theorems—readily apply.

### 3.7.1 Exact Algorithms

The algorithms in Section 1.3 compute the (least or greatest) fixed point in question *exactly* by the Kleene iteration. These can be thought of as special cases of the application of the reasoning principles in Corollary 2.2.8.

- In the case of (LFP-OA-V), the exact computation of the fixed point $\mu f$ amounts to finding $y = f^{\alpha_0}(\bot)$ for large enough $\alpha_0$. This particular way of using the reasoning principle is *complete*: any $x$ such that $x \sqsubseteq \mu f$ is below this $y$.

  The case of (GFP-UA-R) is similar.
- Similarly to the above, in the case of (GFP-UA-V), the exact computation of the fixed point $\nu f$ amounts to finding a *complete* $y$, namely the one given by $\nu f$. Indeed, it is a post-fixed point ($\nu f \sqsubseteq f(\nu f)$), and any $x$ such that $x \sqsubseteq \nu f$ is below this $y$.

Table 3.4: (Co-)safety, (co-)reachability properties for transition systems and $\mathsf{Bwd}^{\Box}$, summary

| | safety | | co-safety |
|---|---|---|---|
| | $i \sqsubseteq \nu(p \sqcap \mathsf{Bwd}^{\Box}(\_))$ | | $i \not\sqsubseteq \nu(p \sqcap \mathsf{Bwd}^{\Box}(\_))$ |
| $\iff$ | $\mu\big(\neg p \sqcup \mathsf{Bwd}^{\Diamond}(\_)\big) \sqsubseteq \neg i$ | $\iff$ | $\mu\big(\neg p \sqcup \mathsf{Bwd}^{\Diamond}(\_)\big) \not\sqsubseteq \neg i$ |
| $\iff$ | $\mu(i \sqcup \mathsf{Fwd}^{\Box}(\_)) \sqsubseteq p$ | $\iff$ | $\mu(i \sqcup \mathsf{Fwd}^{\Box}(\_)) \not\sqsubseteq p$ |
| $\iff$ | $\neg p \sqsubseteq \nu(\neg i \sqcap \mathsf{Fwd}^{\Diamond}(\_))$ | $\iff$ | $\neg p \not\sqsubseteq \nu(\neg i \sqcap \mathsf{Fwd}^{\Diamond}(\_))$ |
| | **reachability** | | **co-reachability** |
| | $i \sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\Box}(\_))$ | | $i \not\sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\Box}(\_))$ |
| $\iff$ | $\nu\big(\neg p \sqcap \mathsf{Bwd}^{\Diamond}(\_)\big) \sqsubseteq \neg i$ | $\iff$ | $\nu\big(\neg p \sqcap \mathsf{Bwd}^{\Diamond}(\_)\big) \not\sqsubseteq \neg i$ |

Table 3.5: (Co-)safety, (co-)reachability properties for transition systems and $\mathsf{Bwd}^{\Diamond}$, summary

| | safety | | co-safety |
|---|---|---|---|
| | $i \sqsubseteq \nu(p \sqcap \mathsf{Bwd}^{\Diamond}(\_))$ | | $i \not\sqsubseteq \nu(p \sqcap g(\_))$ |
| $\iff$ | $\mu\big(\neg p \sqcup \mathsf{Bwd}^{\Box}(\_)\big) \sqsubseteq \neg i$ | $\iff$ | $\mu\big(\neg p \sqcup \mathsf{Bwd}^{\Box}(\_)\big) \not\sqsubseteq \neg i$ |
| | **reachability** | | **co-reachability** |
| | $i \sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\Diamond}(\_))$ | | $i \not\sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\Diamond}(\_))$ |
| $\iff$ | $\nu\big(\neg p \sqcap \mathsf{Bwd}^{\Box}(\_)\big) \sqsubseteq \neg i$ | $\iff$ | $\nu\big(\neg p \sqcap \mathsf{Bwd}^{\Box}(\_)\big) \not\sqsubseteq \neg i$ |

Table 3.6: (Co-)safety, (co-)reachability properties for Markov chains and $\mathsf{Bwd}^{\mathsf{Av}}$, summary

| | safety | | co-safety |
|---|---|---|---|
| | $i \sqsubseteq \nu(p \sqcap \mathsf{Bwd}^{\mathsf{Av}}(\_))$ | | $i \not\sqsubseteq \nu(p \sqcap \mathsf{Bwd}^{\mathsf{Av}}(\_))$ |
| $\iff$ | $\mu\big(\neg p \sqcup \mathsf{Bwd}^{\mathsf{Av}}(\_)\big) \sqsubseteq \neg i$ | $\iff$ | $\mu\big(\neg p \sqcup \mathsf{Bwd}^{\mathsf{Av}}(\_)\big) \not\sqsubseteq \neg i$ |
| | **reachability** | | **co-reachability** |
| | $i \sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\mathsf{Av}}(\_))$ | | $i \not\sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\mathsf{Av}}(\_))$ |
| $\iff$ | $\nu\big(\neg p \sqcap \mathsf{Bwd}^{\mathsf{Av}}(\_)\big) \sqsubseteq \neg i$ | $\iff$ | $\nu\big(\neg p \sqcap \mathsf{Bwd}^{\mathsf{Av}}(\_)\big) \not\sqsubseteq \neg i$ |

The case of (LFP-UA-R) is similar.

The exact computation of fixed points is generally feasible for finite-state transition systems, but not for infinite-state transition systems (consider an infinite chain $x_0 \to x_1 \to \cdots$) or for Markov chains (in case $\delta(x_0)(x_0) = \delta(x_0)(x_1) = 1/2$, the reachability probability from $x_0$ to $x_1$ only converges to 1 asymptotically).

More explicit principles, martingales, examples.

### 3.7.2 Concrete Algorithms

We exhibit the relationship with the first examples in Chapter 1.
Firstly, on the demonic safety problem in Section 1.2.2.2:

- it is the safety problem for the predicate transformer $\mathsf{Bwd}^{\square}$.
- Then Algorithm 1 is nothing but the Kleene iteration for the fixed point $\nu(p \sqcap \mathsf{Bwd}^{\square}(\_))$, in the first characterization $i \sqsubseteq \nu(p \sqcap \mathsf{Bwd}^{\square}(\_))$ in Table 3.4.
- Algorithm 2 relies on the second "involution" characterization $\mu\big(\neg p \sqcup \mathsf{Bwd}^{\diamond}(\_)\big) \sqsubseteq \neg i$.
- Algorithm 3 relies on the third "adjoint" characterization $\mu(i \sqcup \mathsf{Fwd}^{\square}(\_)) \sqsubseteq p$.
- All these point to the possibility of yet another algorithm; it should rely on the fourth characterization $\neg p \sqsubseteq \nu(\neg i \sqcap \mathsf{Fwd}^{\diamond}(\_))$.

Secondly, on the angelic reachability problem in Section 1.2.2.3:

- it is the reachability problem for the predicate transformer $\mathsf{Bwd}^{\diamond}$.
- Then Algorithm 4 is nothing but the Kleene iteration for the fixed point $\mu(p \sqcup \mathsf{Bwd}^{\diamond}(\_))$, in the first characterization $i \sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\diamond}(\_))$ in Table 3.5.
- Algorithm 5 relies on the second "involution" characterization $\nu\big(\neg p \sqcap \mathsf{Bwd}^{\square}(\_)\big) \sqsubseteq \neg i$.

The pointwise forward algorithm (Algorithm 6) does not appear evidently in Table 3.5. It can be derived in the following axiomatic way.

$$
\begin{aligned}
& i \sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\diamond}(\_)) \\
\iff & \{x_0\} \sqsubseteq \mu(p \sqcup \mathsf{Bwd}^{\diamond}(\_)) \quad \text{for each } x_0 \in I \\
\overset{(*)}{\iff} & \{x_0\} \sqcap \mu(p \sqcup \mathsf{Bwd}^{\diamond}(\_)) \neq \emptyset \quad \text{for each } x_0 \in I \\
\iff & \mu(p \sqcup \mathsf{Bwd}^{\diamond}(\_)) \not\sqsubseteq \neg\{x_0\} \quad \text{for each } x_0 \in I,
\end{aligned}
$$

and

$$
\begin{aligned}
& \mu(p \sqcup \mathsf{Bwd}^{\diamond}(\_)) \sqsubseteq \neg\{x_0\} \\
\iff & p \sqsubseteq \nu(\neg\{x_0\} \sqcap \mathsf{Fwd}^{\diamond}(\_)) \quad \text{by Proposition 3.5.1 and } \mathsf{Bwd}^{\diamond} \dashv \mathsf{Fwd}^{\diamond} \text{ (Proposition 3.5.4)} \\
\iff & \mu(\{x_0\} \sqcup \mathsf{Fwd}^{\square}(\_)) \sqsubseteq \neg p \quad \text{by Proposition 3.4.9 and } \mathsf{Fwd}^{\square} = (\mathsf{Fwd}^{\diamond})^{\neg\neg}.
\end{aligned}
$$

In the above reasoning, restricting the initial predicate to a singleton allows the reasoning step $(*)$. We are using this general fact:

$$\text{Assume } S \text{ is a singleton. Then } S \subseteq T \text{ if and only if } S \cap T \neq \emptyset. \tag{3.3}$$

After the step $(*)$, we can exploit the adjunction $\mathsf{Bwd}^{\diamond} \dashv \mathsf{Fwd}^{\diamond}$.

One can say that the above derivation of Algorithm 6 is somewhat awkward: it relies on the fact (3.3) that is very specific to the lattice $2^X$; thus it is unlikely to generalize to other settings such as probabilistic. The derivation via singletons also results in a pointwise algorithm, that seems suboptimal, as we discussed in Section 1.2.2.3.

# Chapter 4

# Fixed Points in Complete Lattices: Examples

## 4.1 Extended Example: Modal $\mu$-Calculus

Use Cousot–Cousot for illustration. Saturation within finite steps.

Use the syntax of POPL'16

Mention GFP-OA

## 4.2 Extended Example: Partical and Total Correctness of Programs with While Loops

(Address Tachio's question at POPL'16)

Start first by small-step

partical correctness vs. total correctness

Two semantics of operational flavor: small-step and big-step

Adjunction between them

Mention GFP-OA

## 4.3 Extended Example: Reachability Analysis by Ranking Functions

(Very different from invariants)

## 4.4 Example: Partition Refinement

Use Cousot–Cousot for illustration. Saturation within finite steps. Minimization of DFA?

Mention GFP-OA

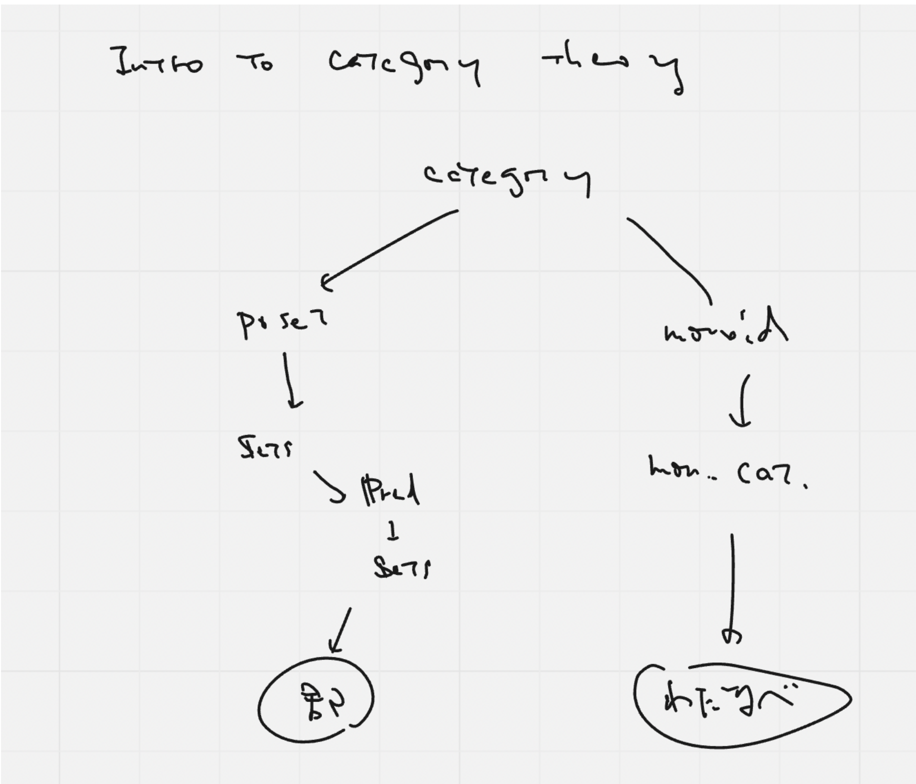
# Chapter 5

# Fixed Points in Categories

As a generalization of Chapter 2, but talking about size. Add pointers from Chapter 2 to this chapter.

- Coinduction
- Some related topics, only very briefly
    - duality-based modal logics

2023/06/15: Revise the story!



# Chapter 6

# Lattice-Theoretic Progress Measures

The theory of [8], detailed

# Chapter 7

# Fixed-Point Reasonig Up-to

Review the works by Pous, and integrate in the theory

# Chapter 8

# Preliminaries II: $\mathbf{CLat}_{\sqcap}$-Fibrations

[Komorida+, NGC Hagiya sp. issue] has got a good introduction

- In this paper, we focus on $\mathbf{CLat}_{\sqcap}$-fibrations
- Introduce first as indexed complete lattices
- The Grothendieck construction $\to$ $\mathbf{CLat}_{\sqcap}$-fibration. Motivate them using examples, such as $\mathbf{Pred} \to \mathbf{Set}$ and $\mathbf{Top} \to \mathbf{Set}$
- Examples
  - Lax slice categories with an ordered object $\Omega$ [**?**] (NB. This may look similar to the codensity situation but is different)

# Chapter 9

# Fibrational Weakest Preconditions as a Foundation

- As a foundation used throughout the paper
- A monadic framework is studied in [**?**], where the theme is the compatibility with the Kleisli composition of computations. In this paper we do not assume such a monad structure
- (Also discuss the strongest post-condition? See [**?**, Section 4.1])

# Chapter 10

# Ranking functions—we verify lfp specifications

- Spell out axiomatics in the setting of Chapter 3. Use well-foundedness. Cite good references, such as Sriram's
- submartingales from fibrations?
- (discuss relationships with Natsuki's LICS'17 paper)

# Chapter 11

# Codensity Lifting

- Important class of “observational” lifting. Generalizing the Kantorovich lifting
- Relationship with $\top\top$-lifting
- Duality with the Wasserstein lifting
- Use in quantiative algebraic reasoning? (Adamek LICS 2022)

# Chapter 12

# Extracting games

- Yuichi's codensity games?
- Quentin's codensity games?
- Delayed/fair simulation?
- Buechi automata, parity automata?

# Chapter 13

# (Co-)induction up-to

- Filippo's results
- Mayuko's IC3?

# Chapter 14

# Predicate abstraction

- Natsuki's timed automata
- Urbat et al., FSCD'21 (GSOS in a fibration)
- CEGAR

# Chapter 15

# IA-FC coincidence

- Mayuko's coincidence? [CONCUR'21]
- Coalgebraic trace semantics?

# Bibliography


[1] S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D.M. Gabbai and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, vol. 3, pp. 1–168. Oxford Univ. Press, 1994. 22

[2] A. Arnold and D. Niwiński. *Rudiments of $\mu$-Calculus*. Studies in Logic and the Foundations of Mathematics. Elsevier, Amsterdam, 2001. 20

[3] P. Blackburn, M. de Rijke and Y. Venema. *Modal Logic*. No. 53 in Tracts in Theor. Comp. Sci. Cambridge Univ. Press, 2001. 32

[4] P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In R.M. Graham, M.A. Harrison and R. Sethi, editors, *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977*, pp. 238–252. ACM, 1977. 16

[5] P. Cousot and R. Cousot. Constructive versions of Tarski's fixed point theorems. *Pacific Journal of Mathematics*, 82(1):43–57, 1979. 21

[6] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Math. Textbooks. Cambridge Univ. Press, 1990. 9

[7] P.R. Halmos. *Algebraic Logic*. American Mathematical Society, 2006. 3

[8] I. Hasuo, S. Shimizu and C. Cîrstea. Lattice-theoretic progress measures and coalgebraic model checking. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*, pp. 718–732. 2016. 51

[9] B. Jacobs. The temporal logic of coalgebras via Galois algebras. *Math. Struct. in Comp. Sci.*, 12:875–903, 2002. 26

[10] B. Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, vol. 59 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2016. 20, 26

[11] M. Kori, F. Ascari, F. Bonchi, R. Bruni, R. Gori and I. Hasuo. Exploiting adjoints in property directed reachability analysis. In C. Enea and A. Lal, editors, *Computer Aided Verification - 35th International Conference, CAV 2023, Paris, France, July 17-22, 2023, Proceedings, Part II*, vol. 13965 of *Lecture Notes in Computer Science*, pp. 41–63. Springer, 2023. 38, 39, 41

[12] D. Kozen. Results on the propositional $\mu$-calculus. *Theor. Comp. Sci.*, 27(3):333–354, 1983. 20

[13] S. Mac Lane. *Categories for the Working Mathematician.* Springer, Berlin, 2nd edn., 1998. 16

[14] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic. A First Introduction to Topos Theory.* Springer, New York, 1992. 13

[15] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comp. Sci.*, 249:3–80, 2000. 20

[16] S. Vickers. *Topology Via Logic.* No. 5 in Tracts in Theor. Comp. Sci. Cambridge Univ. Press, 1989. 13