### Contents

1. **Introduction** 4  
   1.1 Notational Preliminaries 4

2. **Transition Systems** 5

3. **Linear-Time Properties** 6  
   3.1 Linear-Time Behaviour of Transition Systems 7  
   3.2 Safety Properties and Invariants 9  
   3.2.1 Invariants 9  
   3.2.2 Safety Properties 9  
   3.2.3 Regular Safety Properties 11  
   3.2.4 Safety Properties and Trace Equivalence 12  
   3.2.5 König’s Lemma 15  
   3.2.6 Proof of Proposition 3.27 17  
   3.3 Liveness Properties 18  
   3.4 Safety vs Liveness 19

4. **Topological Approach** 19  
   4.1 Generalities 19  
   4.1.1 Adherence 21  
   4.1.2 The Topological Decomposition Theorem 21  
   4.1.3 Bases and Subbases 22  
   4.2 Spaces of $\omega$-Words 23  
   4.2.1 Topological Safety and Liveness 24

5. **Partial Orders and Complete Lattices** 24  
   5.1 Partial Orders 25  
   5.2 Complete Lattices 26  
   5.3 Closure Operators 28  
   5.4 Galois Connections 28  
   5.5 Prefix and Closure 29  
   5.5.1 Alternative Proof of the Decomposition Theorem 3.42 31  
   5.6 Further Properties of Closure Operators and Galois Connections 31  
   5.6.1 On the Kleisli Construction 33

6. **Observable Properties** 33  
   6.1 Observable Properties as Clopen Sets 33  
   6.2 Compactness 35  
   6.2.1 The Finite Intersection Property 37  
   6.3 Compact Hausdorff Spaces 37
Contents

7 Linear Temporal Logic
  7.1 The Logic LML of Observable Properties .................................. 38
    7.1.1 Syntax and Semantics of LML ........................................ 38
    7.1.2 Logical Equivalence .................................................. 39
    7.1.3 Observable Properties .............................................. 41
  7.2 Extending LML with Fixpoints ............................................. 41
    7.2.1 The “Eventually” and “Always” Modalities ......................... 41
    7.2.2 Positive and Negative Variables in a Formula ..................... 43
    7.2.3 The Knaster-Tarski Fixpoint Theorem ............................. 44
  7.3 The Logic LTL ............................................................... 44
    7.3.1 Syntax and Semantics of LTL ....................................... 45
    7.3.2 Fixpoints and Defined Modalities .................................. 45
    7.3.3 Logical Equivalence ................................................ 46
    7.3.4 Positive Normal Forms ............................................. 47
    7.3.5 Satisfaction of LTL-Formulae by Transition Systems ............... 48

8 Toward Stone Duality ....................................................... 48

9 Bisimulation ................................................................. 50
  9.1 Bisimulation (with Actions) ............................................... 50
  9.2 Bisimilarity and Trace Equivalence .................................... 52
  9.3 The Bisimulation Quotient ................................................ 52

10 On Modal Logics of Transition Systems .................................. 52
  10.1 Syntax and Semantics of HML ............................................ 53
  10.2 Logical Equivalence ...................................................... 55
    10.2.1 Logical Equivalence on Formulae ................................ 55
    10.2.2 Logical Equivalence on States and Bisimilarity ................. 55
  10.3 The Hennessy-Milner Property .......................................... 56
  10.4 Modal Saturation .......................................................... 57
  10.5 Boolean Algebras with Operators ..................................... 58
  10.6 Ultrafilter Extensions of Kripke Models ............................... 60
1 Introduction

While the course is mostly based on the book [BK08], we depart from it in several occasions. These notes mainly cover material which is either not presented in [BK08], or on which we substantially differ from [BK08].

In particular, we refer to [BK08, Chap. 2] for a general introduction to verification and model-checking.¹

1.1 Notational Preliminaries

Notation 1.1 (Unions and Intersections). Let \( X \) be a set.

(1) Given a collection \( C \subseteq \mathcal{P}(X) \), we let \( \bigcup C \) be the unique subset of \( X \) such that

\[
(\forall x \in X) \left( x \in \bigcup C \iff (\exists A \in C)(x \in A) \right)
\]

In particular,

\[
\bigcup \emptyset = \emptyset
\]

Moreover, given a family \( (A_i)_{i \in I} \) of subsets of \( X \), we let

\[
\bigcup_{i \in I} A_i := \bigcup \{A_i \mid i \in I\}
\]

(2) Given a collection \( C \subseteq \mathcal{P}(X) \), we let \( \bigcap C \) be the unique subset of \( X \) such that

\[
(\forall x \in X) \left( x \in \bigcap C \iff (\forall A \in C)(x \in A) \right)
\]

In particular,

\[
\bigcap \emptyset = X
\]

Moreover, given a family \( (A_i)_{i \in I} \) of subsets of \( X \), we let

\[
\bigcap_{i \in I} A_i := \bigcap \{A_i \mid i \in I\}
\]

Notation 1.2 (Finite and Infinite Words). Let \( \Sigma \) be an alphabet (i.e. a set).

(1) We write \( \Sigma^\omega \) for the set of infinite words (actually \( \omega \)-words) or streams over \( \Sigma \), i.e. the set of all \( \sigma : \mathbb{N} \to \Sigma \).

(2) We let \( \Sigma^\infty = \Sigma^* \cup \Sigma^\omega \) be the set of finite or infinite words over \( \Sigma \). The empty word is denoted \( \varepsilon \).

(3) Given \( \sigma \in \Sigma^\infty \) and \( \hat{\sigma} \in \Sigma^* \), we write \( \hat{\sigma} \subseteq \sigma \) to mean that \( \hat{\sigma} \) is a (finite) prefix of \( \sigma \), i.e. that

\[
\forall i < \text{length}(\hat{\sigma}), \; \hat{\sigma}(i) = \sigma(i)
\]

¹Until 2008. For a recent project in the subject, see e.g. https://www.aere.iastate.edu/modelchecker/
Given $\sigma \in \Sigma^\infty$ and $E \subseteq \Sigma^\infty$, we let

$$\text{Pref}(\sigma) := \{ \hat{\sigma} \in \Sigma^* | \hat{\sigma} \subseteq \sigma \}$$

$$\text{Pref}(E) := \bigcup_{\sigma \in E} \text{Pref}(\sigma) \quad (= \bigcup \{ \text{Pref}(\sigma) | \sigma \in E \})$$

Further, we often write $\sigma$ for an $\omega$-word in $\Sigma^\omega$ and $\hat{\sigma}$ for a finite word $\Sigma^*$. 

2 Transition Systems

Fix a set $\text{AP}$ of atomic propositions. Recall from [BK08, Def. 2.1] that a transition system over $\text{AP}$ is a tuple

$$TS = (S, \text{Act}, \rightarrow, I, \text{AP}, L)$$

where

- $S$ is the set of states,
- $I \subseteq S$ is the set of initial states,
- $\text{Act}$ is the set of actions,
- $\rightarrow \subseteq S \times \text{Act} \times S$ is the transition relation,
- $L : S \rightarrow \mathcal{P}(\text{AP})$ is the labelling function.

Example 2.1 (The Beverage Vending Machine of [BK08, Ex. 2.2]). We consider the beverage vending machine (BVM) depicted in Fig. 1. Formally, this transition system has:

state set: $S = \{ \text{pay, soda, beer} \}$, with pay initial;
3 Linear-Time Properties

**action set:** Act = \{ic, gs, gb, \tau\}.

The intention is that in state pay the machine is waiting for the customer to pay. Payment is modeled by the action ic (short for “insert coin”). Upon payment, the machine goes in state select, from which the beverage to be delivered is chosen. This choice is not up to the customer: the two transitions out of the state select are labeled with the same action \tau. From state soda the action gs (short for “get soda”) expresses that the customer will get a soda (and similarly from state beer).

We let the set AP of atomic propositions be \{paid, drink\}. The labelling function L (not drawn in Fig. 1) is given by:

\[
\begin{align*}
L(\text{pay}) &= \emptyset \\
L(\text{soda}) &= \{\text{drink}\} \\
L(\text{select}) &= \{\text{paid}\} \\
L(\text{beer}) &= \{\text{drink}\}
\end{align*}
\]

**Remark 2.2 (The Action \tau).** It is a quite general convention to use the distinguished name \tau as in Ex. 2.1 to denote some (possibly non-deterministic) action internal to the system under consideration, where “internal” means that the outside has no information on what is actually done by the system.

We refer to [BK08, Chap. 2] for further examples.

3 Linear-Time Properties

We follow the approach of [BK08, Chap. 3] with a few differences in terminology and notation. Recall Notation 1.2 from §1.1.

**Definition 3.1.** A linear-time (LT) property over a set AP of atomic propositions is a set \(P \subseteq (2^{AP})^\omega\) of \(\omega\)-words \(\sigma \in (2^{AP})^\omega\).

The idea is that an \(\omega\)-word \(\sigma \in (2^{AP})^\omega\) is a function

\[
\sigma : \mathbb{N} \rightarrow 2^{AP} \\
n \mapsto \sigma(n)
\]

where \(\sigma(n) \subseteq AP\) specifies the set of atomic propositions which \(\sigma\) assumes to hold at time \(n \in \mathbb{N}\).

**Example 3.2.** Recall the BVM of Ex. 2.1 (§2), with set of atomic propositions AP = \{paid, drink\}. The following are linear-time properties on this transition system.

1. \(\sigma \in P\) iff in \(\sigma\), each drink occurs after a paid. Formally:

\[
P = \{\sigma \in (2^{AP})^\omega \mid (\forall n \in \mathbb{N}) (\text{drink} \in \sigma(n) \implies \exists k < n. \text{paid} \in \sigma(k))\}
\]

2. \(\sigma \in P\) iff at every moment, there has been at least as many paid’s as drink’s. Formally:

\[
P = \{\sigma \in (2^{AP})^\omega \mid \forall \hat{\sigma} \subseteq \sigma, \text{Card}\{n \mid \text{drink} \in \hat{\sigma}(n)\} \leq \text{Card}\{n \mid \text{paid} \in \hat{\sigma}(n)\}\}
\]
(3) $\sigma \in P$ iff in $\sigma$, there are infinitely many paid's whenever there are infinitely many drink's. Formally:

$$P = \{ \sigma \in (2^{AP})^\omega \mid (\exists^\infty t)(\text{drink} \in \sigma(t)) \implies (\exists^\infty t)(\text{paid} \in \sigma(t)) \}$$

(4) $\sigma \in P$ iff in $\sigma$, there are at most finitely many drink's whenever there are at most finitely many paid's. Formally:

$$P = \{ \sigma \in (2^{AP})^\omega \mid (\forall^\infty t)(\text{paid} \notin \sigma(t)) \implies (\forall^\infty t)(\text{drink} \notin \sigma(t)) \}$$

**Notation 3.3** (The Quantifiers $\exists^\infty$ and $\forall^\infty$). In Ex. 3.2 we used the quantifiers $\exists^\infty$ and $\forall^\infty$. These customary notations for linear-time properties stand for the following.

The "ininitely many" quantifier $(\exists^\infty t)(\cdots t \cdots)$ unfolds to $(\forall N \in \mathbb{N})(\exists t \geq N)(\cdots t \cdots)$. This precisely means that there are infinitely many $t \in \mathbb{N}$ such that $(\cdots t \cdots)$. For instance, $(\exists^\infty t)(\text{paid} \in \sigma(t))$ means that there are infinitely many $t \in \mathbb{N}$ such that paid $\in \sigma(t)$.

The "ultimately all" quantifier $(\forall^\infty t)(\cdots t \cdots)$ unfolds to $(\exists N \in \mathbb{N})(\forall t \geq N)(\cdots t \cdots)$. This means that there are at most finitely many $t \in \mathbb{N}$ such that $(\cdots t \cdots)$ fails, or equivalently that $(\cdots t \cdots)$ holds for ultimately all $t \in \mathbb{N}$. For instance, $(\forall^\infty t)(\text{paid} \notin \sigma(t))$ means that there are at most finitely many $t \in \mathbb{N}$ such that paid $\in \sigma(t)$, equivalently that paid $\notin \sigma(t)$ for ultimately all $t \in \mathbb{N}$.

We refer to [BK08, Chap. 3] for further examples.

### 3.1 Linear-Time Behaviour of Transition Systems

Fix a transition system $TS = (S, \text{Act}, \rightarrow, I, AP, L)$ over $AP$.

**Definition 3.4** (Path). A (finite or infinite) path in $TS$ is a finite or infinite sequence of states $\pi = (s_i)_{i < n}$ with $n \leq \omega$, which respects the transitions of $TS$ in the sense that for all $i$ such that $i + 1 < n$, we have $s_i \rightarrow a s_{i+1}$ for some $a \in \text{Act}$.

A path $\pi = (s_i)_{i \leq n}$ is **initial** if $s_0$ is initial (i.e. if $s_0 \in I$).

**Definition 3.5** (Trace).

(1) Let $\pi = (s_i)_{i < n}$ be finite or infinite path. The **trace** of $\pi$ is the finite or infinite word

$$L(\pi) := (L(s_i))_{i < n}$$

(2) The set of traces of $TS$ is

$$\text{Tr}(TS) := \{ L(\pi) \mid \pi \text{ finite or infinite initial path of } TS \}$$

We shall write $\text{Tr}^\omega(TS)$ (resp. $\text{Tr}_\text{fin}(TS)$) for the set of infinite (resp. finite) traces of $TS$. 

7
Example 3.6. Recall the BVM of Ex. 2.1 (§2). Its unique infinite trace is $((\emptyset \cdot \{\text{paid}\} \cdot \{\text{drink}\})^\omega$, while its set of finite traces is $\text{Pref} ((\emptyset \cdot \{\text{paid}\} \cdot \{\text{drink}\})^*)$.

Remark 3.7 (Differences with [BK08]). Beware that $\text{Tr}(\text{TS})$ in Def. 3.5 does not coincide with $\text{Traces}(\text{TS})$ as defined in [BK08, §3.2.2]. However, $\text{Tr}_{\text{fin}}(\text{TS})$ does coincide with $\text{Traces}_{\text{fin}}(\text{TS})$ ([BK08, p. 98 & 96]), and $\text{Tr}^\omega(\text{TS})$ is the set of infinite traces in $\text{Traces}(\text{TS})$.

Definition 3.8 (Satisfaction of Linear-Time Properties). We say that $\text{TS}$ satisfies a LT property $P \subseteq (2^{\text{AP}})^\omega$, notation $\text{TS} \models P$, if $\text{Tr}^\omega(\text{TS}) \subseteq P$.

Example 3.9. The BVM of Ex. 2.1 (§2) satisfies all the LT properties of Ex. 3.2.

Remark 3.10. Linear time properties do not take into account the branching structure of transition systems.

Remark 3.11 (Differences with [BK08]). Definition 3.8 coincides with [BK08, Def. 3.11, §3.2.3]. But note our special symbol $\models$ for the satisfaction of LT properties in transition systems, which differs from the notation of [BK08, Def. 3.11]. The reason is that LT properties are properties on the infinite traces of TS’s rather than properties on the TS’s themselves (see Rem. 3.10), while there are well-known modal logics for describing the latter (see §10).

Two transition systems have the same infinite traces if and only if they satisfy the same LT properties.

Proposition 3.12. Given two transition systems $\text{TS}$ and $\text{TS}'$, both over $\text{AP}$, we have $\text{Tr}^\omega(\text{TS}) \subseteq \text{Tr}^\omega(\text{TS}')$ if and only if $\forall P \subseteq (2^{\text{AP}})^\omega, \text{TS}' \models P \implies \text{TS} \models P$

Proof. Exercise! □

Proposition 3.12 is an easy first step in a theme on which we shall come back in §3.2.4 below, namely the comparison of TS’s w.r.t. the LT properties they satisfy.
### 3.2 Safety Properties and Invariants

We now embark in a basic classification of LT properties, which shall be reformulated in §4 with topological notions, and which will be sharpened in §5 using order and lattice theoretic tools. This simple classification considers two families of LT properties:

**Safety Properties**, discussed in this §3.2;

**Liveness Properties**, to be discussed in §3.3.

Safety and liveness properties are related by the following important facts (§3.4):

**The Decomposition Theorem 3.42**: for every LT property \( P \) over \( AP \), there is a safety property \( P_{\text{safe}} \) and a liveness property \( P_{\text{liveness}} \) (both over \( AP \)) such that

\[
P = P_{\text{safe}} \cap P_{\text{liveness}}
\]

**Proposition 3.41**: the only LT property (over \( AP \)) which is both a safety and a liveness property is the “true” property \( (2^{AP})^\omega \).

This classification of LT properties is due to [AS85]. See [BK08, §3.7] for further references.

We fix a set \( AP \) of atomic propositions.

#### 3.2.1 Invariants

An **invariant** is an LT property \( P \subseteq (2^{AP})^\omega \) such that for some propositional formula \( \varphi \) over \( AP \), we have

\[
P = \{ \sigma \in (2^{AP})^\omega \mid \forall i \in \mathbb{N}, \sigma(i) \models \varphi \}
\]

#### 3.2.2 Safety Properties

The idea of **safety properties** is to specify “bad behaviours” that should not occur. Otherwise said, a safety property expresses that “something bad does not occur”. This is formalized as follows for LT properties.
### Definition 3.14 (Safety Property)
We say that \( P \subseteq (2^{\text{AP}})^\omega \) is a **safety property** if there is a (possibly infinite) set of finite words \( P_{\text{bad}} \subseteq (2^{\text{AP}})^* \) such that \( P \) is the set of \( \omega \)-words which avoid \( P_{\text{bad}} \), in the sense that
\[
P = \{ \sigma \in (2^{\text{AP}})^\omega \mid \forall \hat{\sigma} \subseteq \sigma, \ \hat{\sigma} \notin P_{\text{bad}} \}
\]
In this case we say that \( P \) is induced by \( P_{\text{bad}} \).

#### Example 3.15
The LT properties of Ex. 3.2.(1) and (2) are safety properties. The properties of Ex. 3.2.(3) and (4) are not.

#### Example 3.16 (A Traffic Light ([BK08, Ex. 3.23]))
Consider the following transition system
\[
\begin{array}{ccc}
G & \xrightarrow{a} & Y \\
& \xrightarrow{b} & R
\end{array}
\]
with exactly one action, with set of atomic propositions \( \text{AP} = \{G, Y, R\} \), and whose labelling function is given by
\[
\begin{array}{ccc}
G & \mapsto & \{G\} \\
Y & \mapsto & \{Y\} \\
R & \mapsto & \{R\}
\end{array}
\]
A typical safety property on this transition system is “every \( R \) is the immediate successor of a \( Y \)”, which can be formalized as
\[
\{ \sigma \in (2^{\text{AP}})^\omega \mid (\forall n \in \mathbb{N}) \left( R \in \sigma(n) \implies [n > 0 \text{ and } Y \in \sigma(n-1)] \right) \}
\]
Safety properties have a partly finitary nature, since they are generated from sets of finite words \( P_{\text{bad}} \). This suggests to check the satisfaction of safety properties via some inspection of the finite traces of a TS. This is possible under a mild assumption, which is given by the following definition.

#### Definition 3.17
A state \( s \in S \) of a transition system TS is called **terminal** if there are no state \( s' \in S \) and no action \( a \in \text{Act} \) such that \( s \xrightarrow{a} s' \).

#### Proposition 3.18 (Satisfaction of Safety Properties)
Let \( P \subseteq (2^{\text{AP}})^\omega \) be a safety property induced by \( P_{\text{bad}} \). Given a transition system TS over \( \text{AP} \) and without terminal states, we have
\[
\text{TS} \models P \iff \text{Tr}_{\text{fin}}(\text{TS}) \cap P_{\text{bad}} = \emptyset
\]

**Proof.** Exercise!

The assumption that TS has no terminal state is unavoidable in Prop. 3.18.

#### Example 3.19
Consider the following transition system TS (with exactly one action):
\[
\begin{array}{ccc}
\{a\} & \{b\} \\
\xrightarrow{\bullet} & \xrightarrow{\bullet} & \xrightarrow{\top}
\end{array}
\]
Let \( P \) be the safety property induced by \( P_{\text{bad}} = \{a\}^*\{b\} \). Then TS satisfies \( P \) since the only infinite trace of TS is \( \{a\}^\omega \). But \( \{a\}\{b\} \) is a finite trace in TS which belongs to \( P_{\text{bad}} \).
3 Linear-Time Properties

3.2.3 Regular Safety Properties

We essentially follow here [BK08, §4.2], hence momentarily jumping to Chapter 4 (Regular Properties) of the latter.

Definition 3.20 (Regular Safety Property). A safety property \( P \subseteq (2^{AP})^\omega \) is regular if it is induced by a regular set \( P_{bad} \subseteq (2^{AP})^* \).

Let \( P \subseteq (2^{AP})^\omega \) be the regular safety property induced by the regular set \( P_{bad} \subseteq (2^{AP})^* \). Fix an NFA

\[
(A : 2^{AP}) = (Q, \Delta, Q_0, F)
\]

which recognizes \( P_{bad} \). Note that we can assume \( P_{bad} \) to be suffix-closed, and that for \( q \in F \) and \( A \in 2^{AP} \) we have \((q, A, q') \in \Delta\) iff \( q' = q \).

Consider now a transition system \( TS \) over \( AP \):

\[
TS = (S, Act, \rightarrow, I, AP, L)
\]

We define the product transition system

\[
TS \otimes A := (S_\otimes, Act, \rightarrow_\otimes, I_\otimes, AP_\otimes, L_\otimes)
\]
as follows:

- The set of states is \( S_\otimes := S \times Q \).
- The transition relation \( \rightarrow_\otimes \) is defined by the rule

\[
\frac{s \xrightarrow{a} s'}{(q, L(s'), q') \in \Delta} \quad \frac{(s, q) \xrightarrow{a} (s', q')}{(s, q) \xrightarrow{a} (s', q')}
\]

Note that it is the label of the target state \( s' \) of \( s \xrightarrow{a} s' \) which is used as input letter of \( A \).

- The set of initial states \( I_\otimes \) is the set of all pairs \((s_0, q)\) such that \( s_0 \) is initial in \( TS \) and such that we have \((q_0, L(s_0), q)\) for some initial \( q_0 \in Q_0 \).
- \( A_\otimes := Q \).
- \( L_\otimes(s, q) := \{q\} \).

Since the accepting states \( F \) of \( A \) are assumed to be sink states, we can reduce checking \( TS \models P \) to checking that \( TS \otimes A \) satisfies the invariant property induced by

\[
\varphi_A := \bigwedge_{q \in F} \neg q
\]

Note that if \( TS \) has no terminal states, then it follows from Prop. 3.18 that we have

\[
TS \models P \quad \text{iff} \quad \text{Tr}_{\text{fin}}(TS) \cap L(A) = \emptyset
\]
Proposition 3.21. Assume that $TS$ has no terminal states. Then $TS \models P$ iff the transition system $TS \otimes A$ satisfies the invariant induced by $\varphi_A$.

**Proof.** Exercise!

**Remark 3.22.** An immediate consequence of Prop. 3.21 is that it is decidable whether a given finite $TS$ satisfies a given regular safety property. This actually extends (in a non-trivial way) to (sufficiently regularly generated) infinite $TS$'s. We refer to [Wal16] for an overview (outside the scope of this course).

### 3.2.4 Safety Properties and Trace Equivalence

We now continue the task began in Prop. 3.12 (§3.1), and compare transition systems w.r.t. the safety properties they satisfy.

We begin with the following direct consequence of Prop. 3.18 (§3.2.2) for finite traces.

**Lemma 3.23.** Consider $TS$ and $TS'$, both over $AP$ and both without terminal states. We have

$$\text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS') \iff \forall P \subseteq (2^{AP})^\omega \text{ safety, } \hspace{0.5cm} TS' \models P \implies TS \models P$$

**Proof.** Assume first that $\text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$. Let $P$ be induced by $P_{\text{bad}}$ and such that $TS' \models P$. Then by Prop. 3.18 we have $\text{Tr}_{\text{fin}}(TS') \cap P_{\text{bad}} = \emptyset$, from which we get $\text{Tr}_{\text{fin}}(TS) \cap P_{\text{bad}} = \emptyset$, and the result follows, again by Prop. 3.18.

For the converse, let $P$ be the safety property induced by

$$P_{\text{bad}} := (2^{AP})^\ast \setminus \text{Tr}_{\text{fin}}(TS')$$

Then $TS' \models P$ by definition. But then by Prop. 3.18 we have $\text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$ whenever $TS \models P$.

It is fairly easy to see that $TS$ must have no terminal state in Lem. 3.23.

**Example 3.24.** Consider the following two transition systems:

$$\begin{array}{ccc}
\{a\} & \{b\} & \{a\} \\
\rightarrow & \bullet & \rightarrow \\
\cup & \bullet & \cup
\end{array}$$

These $TS$'s satisfy the same LT properties (and in particular the same safety properties) since they have the same infinite traces (Prop. 3.12, §3.1). But the $TS$ on the left-hand side has finite traces that the other one does not have.

The following shows that the assumption on $TS'$ in Lem. 3.23 cannot be omitted either.

**Example 3.25.** Let $TS$ be the following transition system:
Let $TS'$ be the transition system depicted in Fig. 3 (with infinitely many initial states). Then both $TS$ and $TS'$ have set of finite traces $\{a\}^* \cup \{a\}^+ \{b\}^*$. Note that $TS$ has the infinite trace $\{a\}\{b\}^\omega$, while the only infinite trace of $TS'$ is $\{a\}^\omega$. In particular, $TS'$ satisfies the safety property induced by $P_{bad} = \{a\}^+ \{b\}$, but this property is not satisfied by $TS$.

We now look for an analogue of Prop. 3.12 (§3.1) for safety properties. To this end, we shall forbid transition systems which (as the $TS'$ of Ex. 3.25) have infinitely many initial states. This actually lead to the following assumption.

**Definition 3.26** (Finitely Branching TS). A transition system $TS = (S, Act, \rightarrow, I, AP, L)$ is **finitely branching** when the two following conditions are satisfied:

(i) $I$ is finite, and
(ii) for every $s \in S$, there are at most finitely many $s' \in S$ such that $s \rightarrow^a s'$ for some $a \in \text{Act}$.

**Proposition 3.27.** Consider $TS$ and $TS'$, both over $AP$. Assume that $TS$ has no terminal state and that $TS'$ is finitely branching. Then

$$\text{Tr}_\omega(TS) \subseteq \text{Tr}_\omega(TS') \iff \text{Tr}_\text{fin}(TS) \subseteq \text{Tr}_\text{fin}(TS')$$

**Corollary 3.28.** Consider finitely branching $TS$ and $TS'$, both over $AP$ and both without terminal states. Then we have

$$\text{Tr}_\omega(TS) = \text{Tr}_\omega(TS') \iff \forall P \subseteq (2^{AP})^\omega \text{ safety, } \neg TS' \models P \iff \neg TS \models P$$

Example 3.24 shows that the assumption on $TS$ cannot be omitted in Prop. 3.27. As for the assumption on $TS'$, one can consider the following mild modification of Ex. 3.25.

**Example 3.29.** Let $TS$ be the following transition system:

```
{a} → {b}
```

Figure 4: A transition system with infinitely many initial states for Ex. 3.29.
3 Linear-Time Properties

Let $TS'$ be the transition system depicted in Fig. 4 (with infinitely many initial states). Then both $TS$ and $TS'$ have set of finite traces $\{a\}^* \cup \{a\}^+ \{b\}^*$. But $TS$ has the infinite trace $\{a\}^\omega$, while all infinite traces of $TS'$ have the form $\{a\}^+ \{b\}^\omega$.

**Remark 3.30.** For Ex. 3.25 and Ex. 3.29, the transition systems of Fig. 3 and Fig. 4 could have been replaced by (non finitely branching) transition systems with exactly one initial state.

3.2.5 König’s Lemma

Proposition 3.27 relies on a principle of infinite combinatorics known as König’s Lemma. It basically says that if an infinite tree is finitely branching, then it has an infinite path.

We first define the required notions.

**Definition 3.31.**

1. A **tree** over a set $A$ is a set $T \subseteq A^*$ which is closed under prefix: if $u \in T$ and $v \subseteq u$ then $v \in T$.

2. A tree $T$ over $A$ is **finitely branching** if for each $u \in T$, there are at most finitely many $a \in A$ such that $u.a \in T$.

3. An **infinite path** in a tree $T$ over $A$ is an $\omega$-word $\pi \in A^\omega$ whose finite prefixes belong all to $T$:

   $$\forall n \in \mathbb{N}, \quad \pi(0) \cdots \pi(n) \in T$$

Note that a tree $T$ over $A$ is automatically finitely branching if $A$ is finite.

**Lemma 3.32** (König’s Lemma). If $T$ is an infinite tree which is finitely-branching, then $T$ has an infinite path.

**Proof.** Given a tree $T \subseteq A^*$ and $u \in A^*$, we write $T|u$ for the **subtree** of $T$ at $u$: $T|u := \{ v \in T \mid u \subseteq v \text{ or } v \subseteq u \}$

Fix a tree $T \subseteq A^*$, and assume that $T$ is infinite and finitely branching. We build an infinite path $\pi = (a_n)_{n \in \mathbb{N}}$ by induction on $n \in \mathbb{N}$ as follows. First, note that $T$ is the union of the $T|a$ for $a \in A$. Since $T$ is infinite and finitely branching, by the infinite pigeonhole principle there is some $a \in A$ such that $T|a$ is infinite. We let $a_0 := a$.

Iterating this process, we obtain a sequence $(a_n)_{n \in \mathbb{N}}$ such that

- $a_0 \cdots a_n \in T$ for all $n \in \mathbb{N}$,
- $T|(a_0 \cdots a_n)$ is infinite for all $n \in \mathbb{N}$.

Assuming $a_0, \ldots, a_n$ defined, since

$$T|(a_0 \cdots a_n) = \bigcup_{a \in A} T|(a_0 \cdots a_n a)$$

is infinite and finitely branching, by the infinite pigeonhole principle there is some $a \in A$ such that $a_0 \cdots a_n a \in T$ and $T|(a_0 \cdots a_n a)$ is infinite. We let $a_{n+1} := a$. \qed
No assumption of König’s Lemma 3.32 can be omitted. First, $T$ trivially needs be infinite to have an infinite path. Finite branching is also easy to observe.

**Example 3.33.** The following tree over $\mathbb{N}$ is infinite but has no infinite path:

$$
\begin{array}{c}
0 \\
00 \\
000 \\
0001 \\
\ldots \\
1 \\
01 \\
001 \\
\ldots \\
\varepsilon \\
\ldots
\end{array}
$$

Why $T$ is required to be a tree is not too difficult to see neither, but perhaps more subtle. We come back on this in Ex. 4.18 (§4.2).

**Example 3.34.** The set $T_0 := 0^*1 \subseteq \{0, 1\}^*$ is infinite, finitely branching but is not a tree: $T_0$ is not closed under prefix since it is prefix-free (if $u \in T_0$ then no proper prefix of $u$ belongs to $T_0$). In particular, it is clear that $T_0$ has no infinite path.

On the other hand, the tree $T := \text{Pref}(T_0)$ has a unique infinite path, namely $0^\omega$. But note that no prefix of $0^\omega$ belongs to $T_0$! See Fig. 5 (in which nodes are the $v \in T$).

**Remark 3.35** (On Definition 3.31). The notion of tree in Def. 3.31 formally differs from the graph-theoretic one. See e.g. [Kec95, 4.13 (§4B)] for a comparison.

**Remark 3.36** (References). A striking aspect of König’s Lemma 3.32 is that there are recursive infinite trees $T \subseteq \{0, 1\}^*$ with no recursive infinite path (see e.g. [TvD88, Chap. 4, §7.6] or [Sim10, Lem. VIII.2.15]). We refer to [Sim10, I.8.8 and §III.7] for the axiomatic strength of König’s Lemma (outside the scope of this course).

König’s Lemma 3.32 is an important tool for various topics related to this course. First, see [BBJ07, §26.2] for an approach based on logic and an application to graphs (namely Ramsey’s Theorem). Moreover, König’s Lemma 3.32 has important applications in the theory of automata on $\omega$-words, see e.g. [GTW02, §3] (also [PP04, §1.9]) or [VW08, §2.2.1] (outside the scope of this course). Last but not least, König’s Lemma 3.32 is strongly related to topological compactness. We come back on this in §6.2 (Prop. 6.12, Rem. 6.13 and Rem. 6.14).
3 Linear-Time Properties

3.2.6 Proof of Proposition 3.27

We can now prove Prop. 3.27.

**Proof.** Assume first that $\text{Tr}^\omega(TS) \subseteq \text{Tr}^\omega(TS')$. Then given $\hat{\sigma} \in \text{Tr}_{\text{fin}}(TS)$, since $TS$ has no terminal states we have $\hat{\sigma} \subseteq \sigma$ for some $\sigma \in \text{Tr}^\omega(TS) \subseteq \text{Tr}^\omega(TS')$, and it follows that $\hat{\sigma} \in \text{Tr}_{\text{fin}}(TS')$.

For the converse, assume $\text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$ and let $\sigma \in \text{Tr}^\omega(TS)$. Then for all $\hat{\sigma} \subseteq \sigma$ we have $\hat{\sigma} \in \text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$. As a consequence, for all $n \in \mathbb{N}$ there is in $TS'$ a finite initial path

$$\pi_n = s_0^n \cdots s_n^n$$

such that

$$L'(\pi_n) \subseteq \sigma$$

But note that we may not have $\pi_n \subseteq \pi_{n+1}$. We therefore apply König’s Lemma 3.32 to build a suitable infinite path in $TS'$. Consider the tree $T \subseteq (S')^*$ defined as

$$T := \{ u \in (S')^* \mid u \text{ is a finite initial path in } TS' \text{ and } L'(u) \subseteq \sigma \}$$

Then $T$ is evidently a tree. It is finitely branching since $TS'$ is finitely branching. Moreover $T$ is infinite: for all $\hat{\sigma} \subseteq \sigma$ we have $\hat{\sigma} \in \text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$, so there is a finite initial path $u$ in $TS'$ such that $L'(u) = \hat{\sigma}$. By König’s Lemma 3.32, $T$ has an infinite path $\pi$. We have $L'(\pi) = \sigma$ since $L'(\pi(0) \cdots \pi(n)) \subseteq \sigma$ for all $n \in \mathbb{N}$. Moreover, $\pi$ is an initial path in $TS'$ by construction of $T$. \hfill \square

**Direct Proof of Proposition 3.27.** We can nevertheless give a direct proof of Prop. 3.27.

**Proof.** Assume first that $\text{Tr}^\omega(TS) \subseteq \text{Tr}^\omega(TS')$. Then given $\hat{\sigma} \in \text{Tr}_{\text{fin}}(TS)$, since $TS$ has no terminal states we have $\hat{\sigma} \subseteq \sigma$ for some $\sigma \in \text{Tr}^\omega(TS) \subseteq \text{Tr}^\omega(TS')$, and it follows that $\hat{\sigma} \in \text{Tr}_{\text{fin}}(TS')$.

For the converse, assume $\text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$ and let $\sigma \in \text{Tr}^\omega(TS)$. Then for all $\hat{\sigma} \subseteq \sigma$ we have $\hat{\sigma} \in \text{Tr}_{\text{fin}}(TS) \subseteq \text{Tr}_{\text{fin}}(TS')$. As a consequence, for all $n \in \mathbb{N}$ there is in $TS'$ a finite initial path

$$\pi_n = s_0^n \cdots s_n^n$$

such that

$$L'(\pi_n) \subseteq \sigma$$

But note that we may not have $\pi_n \subseteq \pi_{n+1}$. However, since $TS'$ is finitely branching, by the infinite pigeonhole principle there is some state $s \in S'$ such that $s_0^n = s$ for infinitely many $n \in \mathbb{N}$. This induces an infinite subsequence $(\pi_n^0)_{n \in \mathbb{N}}$ of $(\pi_n)_{n \in \mathbb{N}}$ with $\pi_n^0(0) = s$ for all $n \in \mathbb{N}$. Iterating this process, we get for each $k \in \mathbb{N}$ an infinite sequence $(\pi_n^k)_{n \in \mathbb{N}}$ such that

- each $\pi_n^k$ has length at least $n$,
- $(\pi_{n+1}^{k+1})_{n \in \mathbb{N}}$ is an infinite subsequence of $(\pi_n^k)_{n \in \mathbb{N}}$,
• and for each \( k \in \mathbb{N} \), all the
\[
\tilde{\pi}_n^k := \pi_n^k(0) \cdots \pi_n^k(k)
\]
agree for \( n \geq k \).

It follows that \((\tilde{\pi}_n^k)_{k \in \mathbb{N}}\) forms an infinite strictly increasing sequence in \((S')^*\). Then its limit \( \pi \) in \((S')^\omega\) is an infinite initial path in \(TS'\) such that \( L'(\pi) = \sigma \). Hence \( \sigma \in \text{Tr}^\omega(TS') \) and we are done.

### 3.3 Liveness Properties

While safety properties specify that “nothing wrong can happen”, a given safety property may vacuously hold in a “sufficiently inactive” system.

**Example 3.37.** Recall Ex. 3.16 (§3.2.2) and consider the following “traffic light”:

\[ \xymatrix{ \} \quad \ar[r] & G \ar[r] & } \]

with \( \text{AP} = \{G, Y, R\} \) and state labelling \( G \mapsto \{G\} \). This system trivially satisfies the safety property of Ex. 3.16 (“every \( R \) is the immediate successor of a \( Y \)”).

Liveness properties, which form the second part of the classification mentioned in §3.2, specify that “something good will happen”.

**Definition 3.38 (Liveness Property).** We say that \( P \subseteq (2^\text{AP})^\omega \) is a liveness property if for every \( \hat{\sigma} \in (2^\text{AP})^* \) there is some \( \sigma \in P \) such that \( \hat{\sigma} \subseteq \sigma \).

Liveness properties are typically conditions on infinite behaviours.

**Example 3.39.** A typical liveness property which can be used to ensure that the safety property of Ex. 3.37 holds in a non-trivial way is: “\( R \) occurs infinitely often”. Formally:

\[ \{\sigma \in (2^\text{AP})^\omega \mid (\exists \infty t)(R \in \sigma(t))\} \]

This property is satisfied by the transition system of Ex. 3.16 but not by the one of Ex. 3.37.

Example 3.39 suggests to consider the conjunction of a safety property with a liveness property. This is no special case: the Decomposition Theorem 3.42 (in §3.4 below) states that every LT property is the conjunction of a safety property with a liveness property.

**Example 3.40.** Recall the BVM of Ex. 2.1 (§2). The following two properties from Ex. 3.2 are liveness properties:

- \( \sigma \in P \) iff in \( \sigma \), there are infinitely many paid’s whenever there are infinitely many drink’s:
  \[ P = \{\sigma \in (2^\text{AP})^\omega \mid (\exists \infty t)(\text{drink} \in \sigma(t)) \implies (\exists \infty t)(\text{paid} \in \sigma(t))\} \]

- \( \sigma \in P \) iff in \( \sigma \), there are at most finitely many drink’s whenever there are at most finitely many paid’s:
  \[ P = \{\sigma \in (2^\text{AP})^\omega \mid (\forall \infty t)(\text{paid} \notin \sigma(t)) \implies (\forall \infty t)(\text{drink} \notin \sigma(t))\} \]

See [BK08, §3.5] for more.
3.4 Safety vs Liveness

We now turn to the two results relating safety and liveness which were mentioned in §3.2.

**Proposition 3.41** ([BK08, Lem. 3.35]). *The only LT property which is both a safety and a liveness property is the “true” property \((2^{AP})^\omega\).*

**Proof.** First, note that \((2^{AP})^\omega\) is evidently a liveness property. It is also the safety property induced by \(P_{\text{bad}} := \emptyset\).

Conversely, let \(P \subseteq (2^{AP})^\omega\) be both a liveness property and a safety property, say induced by \(P_{\text{bad}}\). Then for every \(\hat{\sigma} \in P_{\text{bad}}\) there must be some \(\sigma \in P\) such that \(\hat{\sigma} \subseteq \sigma\). But this implies \(P_{\text{bad}} = \emptyset\). \(\square\)

**Theorem 3.42** (Decomposition ([BK08, Thm. 3.37])). *For every LT property \(P \subseteq (2^{AP})^\omega\), there is a safety property \(P_{\text{safe}}\) and a liveness property \(P_{\text{liveness}}\) such that*

\[
P = P_{\text{safe}} \cap P_{\text{liveness}}
\]

The Decomposition Theorem 3.42 will be proved in §4.2 as a corollary of a topological decomposition theorem. An alternative proof, based on closure operators and Galois connections and actually following [BK08, Thm. 3.37], is presented in §5.5.1.

4 Topological Approach

Topology has different purposes in this course.

First, we shall see that \(\Sigma^\omega\) (the set of \(\omega\)-words over \(\Sigma\)) can be equipped with a nice notion of topology. In the case of LT properties over \(AP\), the topology on \((2^{AP})^\omega\) provides clean characterizations of safety and liveness, and exhibits the Decomposition Theorem 3.42 as a basic topological fact.

Second, as we shall see in §6 and §7.1, the topology on \((2^{AP})^\omega\) will give us a strong ground on how to build logics (i.e. syntaxes) to describe LT properties. This will be sharpened in §8, with Stone’s Representation Theorem which establishes a deep connection between Boolean algebras on the one hand, and the so called Stone’s spaces on the hand, of which \(\Sigma^\omega\) (for \(\Sigma\) finite) is an important example.

Further, in §10 we shall consider (Hennessy-Milner) modal logic, which allows us to reason on the branching structure of transitions systems, and whose model theory rests on Stone’s Representation Theorem.

To keep things simple, the exposition of this Section is oriented toward presenting (and proving) the Decomposition Theorem 3.42 in its natural topological context.

4.1 Generalities

We expose some basic fundamental concepts and facts on general (or set-theoretic) topological spaces. We refer to [Wil70, Chap. 2] and [Run05, Chap. 3] for most of the material.

**Definition 4.1.** A topological space is a pair \((X, \Omega X)\) where \(X\) is a set and \(\Omega X \subseteq \mathcal{P}(X)\) is a family of subsets of \(X\), called the open subsets of \(X\), and such that
• \(\Omega X\) is closed under unions: given a family \((U_i)_{i \in I}\) of open sets, the set \(\bigcup_{i \in I} U_i\) is open as well, and

• \(\Omega X\) is closed under finite intersections: given a finite family \((U_i)_{i \in I}\) of open sets, the set \(\bigcap_{i \in I} U_i\) is open.

The complements of open sets, i.e. the sets of the form \(X \setminus U\) for \(U\) open, are called closed.

Note that \(\emptyset\) and \(X\) (as resp. the empty union and the empty intersection) are always open (and thus closed) in \((X, \Omega X)\). Moreover, closed sets are closed under arbitrary intersections and finite unions.

**Lemma 4.2.** Let \((X, \Omega X)\) be a topological space.

• Given a family \((C_i)_{i \in I}\) of closed sets, the set \(\bigcap_{i \in I} C_i\) is closed as well.

• Given a finite family \((C_i)_{i \in I}\) of closed sets, the set \(\bigcup_{i \in I} C_i\) is closed.

In order to show that a particular subset of a topological space is open (resp. closed), one usually proceeds by the following basic fact.

**Lemma 4.3.** Let \((X, \Omega X)\) be a topological space.

1. A set \(A \subseteq X\) is open iff for every \(x \in A\) there is an open set \(U \in \Omega X\) such that \(x \in U\) and \(U \subseteq A\).

2. A set \(A \subseteq X\) is closed iff for every \(x \notin A\) there is an open set \(U \in \Omega X\) such that \(x \in U\) and \(U \cap A = \emptyset\).

**Proof.** Exercise! \(\square\)

Every subset \(A\) of topological space \((X, \Omega X)\) is contained in a least closed set \(\overline{A}\).

**Definition 4.4** (Closure of a set). Given a topological space \((X, \Omega X)\) and a set \(A \subseteq X\), the closure \(\overline{A}\) of \(A\) is defined as

\[
\overline{A} := \bigcap \{C \subseteq X \mid A \subseteq C \text{ and } C \text{ is closed}\}
\]

Note that \(\overline{A}\) is closed as an intersection of closed sets. Moreover, \(\overline{A}\) is the least closed set containing \(A\):

• if \(A \subseteq C\) with \(C\) closed, then \(\overline{A} \subseteq C\).

In particular, a set \(A \subseteq X\) is closed iff \(\overline{A} = A\). The following is [Wil70, Thm. 3.7]. See also [Run05, Def. 3.1.19 & Thm. 3.1.20].

**Lemma 4.5.** Given subsets \(A, B \subseteq X\) of a topological space \((X, \Omega X)\), we have

1. \(A \subseteq B\) implies \(\overline{A} \subseteq \overline{B}\),
4 Topological Approach

(2) \( A \subseteq \mathfrak{A}, \)

(3) \( \overline{A} = A, \)

(4) \( \emptyset = \emptyset, \)

(5) \( \overline{A \cup B} = \overline{A} \cup \overline{B}. \)

Proof. Exercise!

Remark 4.6. Given a set \( A, \) an operator \((\overline{\cdot}): \mathcal{P}(A) \to \mathcal{P}(A)\) satisfying all the conditions of Lem. 4.5 is called a Kuratowski closure operator. Closure operators, which are only required to satisfy the first three conditions of Lem. 4.5 are further discussed in the context of partial orders in §5.3 (Ex. 5.19). It in particular follows from Lem. 5.20 that a Kuratowski closure operator \((\overline{\cdot}) : \mathcal{P}(X) \to \mathcal{P}(X)\) induces a topology on \( X, \) with \( C \subseteq X \) closed iff \( \overline{C} = C. \)

4.1.1 Adherence

The following notion is useful to reason on the closure of a set. We refer to [Bou07, Chap. 1] for developments.

Definition 4.7 (Adherent Point). Consider a topological space \((X, \Omega_X)\) and some \( A \subseteq X. \) We say that \( x \in X \) is adherent to \( A \) (or that \( x \) is an adherent point of \( A \)) if \( A \) intersects any open set which contains \( x: \)

\[ \forall U \in \Omega_X, \ x \in U \implies A \cap U \neq \emptyset \]

Remark 4.8 (Terminology). In the English terminology, adherent points are also called points of closure.

Adherent points provide a handy characterization of the closure of a set.

Lemma 4.9. Consider a topological space \((X, \Omega_X)\) and some \( A \subseteq X. \) Then \( x \) is adherent to \( A \) if and only if \( x \in \overline{A}. \)

Proof. Exercise!

4.1.2 The Topological Decomposition Theorem

Definition 4.10 (Dense Set). Let \((X, \Omega_X)\) be a topological space. A set \( D \subseteq X \) is dense if \( D \cap U \neq \emptyset \) for all non-empty open \( U. \)

Theorem 4.11 (Topological Decomposition Theorem). Let \((X, \Omega_X)\) be a topological space. Then for any \( A \subseteq X, \) there is some closed set \( C \) and some dense set \( D \) such that \( A = C \cap D. \)
4 Topological Approach

Proof. Let \( C := A \) and \( D := A \cup (X \setminus \overline{A}) \). The set \( C \) is trivially closed. Moreover,
\[
C \cap D = (\overline{A} \cap A) \cup (\overline{A} \cap (X \setminus \overline{A})) = A
\]
It thus remains to show that \( D \) is dense. So let \( U \) be a non-empty open set. If \( U \cap A = \emptyset \), then \( A \) is included in the closed set \( X \setminus U \). But this implies \( \overline{A} \subseteq X \setminus U \), so that \( U \subseteq X \setminus \overline{A} \).

Note that the density of \( D = A \cup (X \setminus \overline{A}) \) may be easier to see via the notion of adherence (§4.1.1). Indeed consider a non-empty open \( U \) such that \( U \cap (X \setminus \overline{A}) \) is empty. This means that \( U \) is included in \( \overline{A} \) and since \( U \) is non-empty, there is some \( x \in \overline{A} \) such that \( x \in U \). Now, by Lem. 4.9, \( x \) is adherent to \( A \) since \( x \in \overline{A} \), which implies \( U \cap A \neq \emptyset \) since \( x \in U \) with \( U \) open.

Let us finally mention a useful property on dense sets.

Lemma 4.12. Let \((X, \Omega_X)\) be a topological space. A set \( D \subseteq X \) is dense if and only if \( \overline{D} = X \).

Proof. Exercise!

Remark 4.13 (Alternative Proof of Thm. 4.11). Lemma 4.12, together with the fact that \((-\)\) is a Kuratowski closure operator (see Lem. 4.5 and Rem. 4.6) gives a more direct proof of the Topological Decomposition Theorem 4.11, similar in spirit to the proof of [BK08, Thm. 3.37] (see §5.5.1). The argument goes as follows. Taking \( D := A \cup (X \setminus \overline{A}) \) as in the proof of Thm. 4.11, by Lem. 4.5 we have
\[
\overline{D} = \overline{A \cup (X \setminus \overline{A})}
\]
It follows that \( \overline{D} = X \), and thus that \( D \) is dense by Lem. 4.12.

4.1.3 Bases and Subbases

It is often convenient to define a topology from more atomic data than the direct description of open sets.

Lemma 4.14 (Base). Consider a set \( X \) together with a family of sets \( B \subseteq \mathcal{P}(X) \) which is closed under finite intersections. Let \( \Omega_X \) consist of all the \( \bigcup_{i \in I} U_i \) for \((U_i)_{i \in I}\) a family of elements of \( B \). Then \((X, \Omega_X)\) is a topological space.

Proof. Exercise!

A family \( B \) as in Lem. 4.14 is a base of the topology \( \Omega_X \). In practice, it is often more convenient to generate a base as the closure under finite intersections of an arbitrary family \( B_0 \) subsets of \( X \). Such \( B_0 \) are the called subbases of \( \Omega_X \).
4 Topological Approach

4.2 Spaces of $\omega$-Words

Definition 4.15 (The Topology on Streams). Given a non-empty set $\Sigma$, we equip $\Sigma^{\omega}$ with the topology induced by the subbase $\{\text{ext}(u)\}_{u \in \Sigma^*}$, where

$$\text{ext}(u) := \{ \sigma \in \Sigma^{\omega} \mid u \subseteq \sigma \}$$

Note that $\Sigma^{\omega} = \text{ext}(\varepsilon)$. Also, if $u, v \in \Sigma^*$ are incomparable w.r.t. the prefix order then $\text{ext}(u) \cap \text{ext}(v) = \emptyset$. Moreover, $v \subseteq u$ obviously implies $\text{ext}(u) \subseteq \text{ext}(v)$. We actually have the following.

Lemma 4.16. Assume that $\Sigma$ has at least two elements. Given $u, v \in \Sigma^*$ we have

$$\text{ext}(u) \subseteq \text{ext}(v) \quad \text{iff} \quad v \subseteq u$$

Proof. Exercise!

As a consequence, every open of $\Sigma^{\omega}$ is a union of sets of the form $\text{ext}(u)$ for $u \in \Sigma^*$. Lemma 4.3 gives a quite useful characterization of the open (resp. closed) subsets of $\Sigma^{\omega}$.

Lemma 4.17. Let $\Sigma$ be a non-empty set.

1. A set $P \subseteq \Sigma^{\omega}$ is open iff for every $\sigma \in P$ there is a finite word $\hat{\sigma} \in \Sigma^*$ such that $\hat{\sigma} \subseteq \sigma$ and $\beta \in P$ for all $\beta \in \Sigma^{\omega}$ such that $\hat{\sigma} \subseteq \beta$.

2. A set $P \subseteq \Sigma^{\omega}$ is closed iff for every $\sigma \not\in P$ there is a finite word $\hat{\sigma} \in \Sigma^*$ such that $\hat{\sigma} \subseteq \sigma$ and $\beta \not\in P$ for all $\beta \in \Sigma^{\omega}$ such that $\hat{\sigma} \subseteq \beta$.

In particular, if $C \subseteq \Sigma^{\omega}$ is closed, then given $\sigma \in \Sigma^{\omega}$, we have $\sigma \in C$ whenever for all $\hat{\sigma} \subseteq \sigma$ there is some $\beta \in C$ with $\hat{\sigma} \subseteq \beta$.

Example 4.18 (Closed Sets from Trees (§3.2.5)). Recall Def. 3.31. Given a tree $T \subseteq \Sigma^*$, the set $\text{cl}(T)$ of infinite paths of $T$ is a closed subset of $\Sigma^{\omega}$. For instance, the closed set $\text{cl}($Pref($0^*1$)) $\subseteq \{0,1\}^{\omega}$ is the singleton $\{0^{\omega}\}$ (Ex. 3.34). Actually, we shall see in §5.5 (Cor. 5.28) that the closed subsets of $\Sigma^{\omega}$ are exactly the cl($T$) for trees $T \subseteq \Sigma^*$. See e.g. [Kec95, §2B] for more.

Notation 4.19. Given $U \subseteq \Sigma^*$ we let

$$\text{ext}(u) := \{ \sigma \in \Sigma^{\omega} \mid u \subseteq \sigma \}$$

$$\text{ext}(U) := \bigcup_{u \in U} \text{ext}(u)$$

Remark 4.20 (A Base on Streams). The set $\mathcal{B}_\Sigma \subseteq \mathcal{P}(\Sigma^{\omega})$ consisting of all sets of the form $\text{ext}(U)$ for $U \subseteq \Sigma^*$ finite can be used as a base for a topology on $\Sigma^{\omega}$. It is easy to see that this topology coincides with that of Def. 4.15.

Proof. Exercise!
5 Partial Orders and Complete Lattices

Remark 4.21 (On Finite or Infinite Words). While we focus on infinite words $\sigma \in \Sigma^\omega$, it is sometimes useful to topologize the set $\Sigma^\infty$ of finite or infinite words (see Notation 1.2, §1.1). A good (advanced) example in the context of this course is [VVK05]. We refer to [PP04, §III.4] for a detailed account of $\Sigma^\infty$ as a topological space.

Remark 4.22 (An Informal Analogy with Recursively Enumerable Sets). Given a finite word $u \in \Sigma^*$ and an $\omega$-word $\sigma \in \Sigma^\omega$, we can check whether $\sigma \in \text{ext}(u)$ by only inspecting a finite prefix of $\sigma$. Consider now an open set $\text{ext}(W)$ with $W \subseteq \Sigma^*$, and assume that we want to check whether $\sigma \in \text{ext}(W)$. If it happens that $\sigma \in \text{ext}(W)$, then we can know this after checking whether $\sigma \in \text{ext}(w)$ for only finitely many $w \in W$. But if $\sigma \notin \text{ext}(W)$, then we might have to check whether $\sigma \in \text{ext}(w)$ for infinitely many $w \in W$.

This suggests an analogy between membership of an $\omega$-word to a given open subset of $\Sigma^\omega$ on the one hand, and membership of a natural number to a given recursively enumerable set on the other hand. This mere analogy can actually be made formal, as detailed in [Mos09, Chap. 3] (outside the scope of this course).

4.2.1 Topological Safety and Liveness

Lemma 4.23. An LT property is closed if and only if it is a safety property.

Proof. Exercise! \hfill \square

Lemma 4.24. An LT property is dense if and only if it is a liveness property.

Proof. Exercise! \hfill \square

The Decomposition Theorem 3.42 is thus a direct consequence of the Topological Decomposition Theorem 4.11.

Corollary 4.25 (Decomposition (Thm. 3.42)). For every LT property $P \subseteq (2^\text{AP})^\omega$, there is a safety property $P_{\text{safe}}$ and a liveness property $P_{\text{liveness}}$ such that

$$P = P_{\text{safe}} \cap P_{\text{liveness}}$$

Let us finally mention the following alternative characterization of liveness properties.

Corollary 4.26. An LT property $P \subseteq (2^\text{AP})^\omega$ is a liveness property if and only if $P = (2^\text{AP})^\omega$.

Corollary 4.26 is a direct consequence of Lem. 4.24 and Lem. 4.12.

5 Partial Orders and Complete Lattices

In this Section we introduce some basic concepts and facts on partial orders and complete lattices which will be used for different purposes in this course.

First, these tools provide a purely order-theoretic proof of the Decomposition Theorem 3.42 (essentially as in [BK08, Thm. 3.37]). Moreover, some order-theoretic notions
5 Partial Orders and Complete Lattices

which are good generalizations of topological ones can serve as useful abstractions for the latter. Further, some basic order-theoretic notions presented here lay the ground to central lattice-theoretic concepts for Stone’s Representation Theorem (§8).

Second, complete lattices have a nicely behaved notion of fixpoint on which we rely to define (and reason on) the logic LTL in §7.

We mainly refer to [DP02], and we indicate differences in notation and terminology whenever possible.

5.1 Partial Orders

Definition 5.1 ([DP02, Def. 1.2]). A partial order is a pair \((A, \leq)\) where \(A\) is a set and \(\leq\) is a binary relation on \(A\) which is

reflexive: \(a \leq a\) for all \(a \in L\),

transitive: \(a \leq c\) whenever \(a \leq b\) and \(b \leq c\),

antisymmetric: \(a = b\) whenever \(a \leq b\) and \(b \leq a\).

Example 5.2. The following are simple but important examples of partial orders which are not linear (i.e., in which \(a \not\leq b\) may not imply \(b \leq a\)):

1. \((\Sigma^*, \subseteq)\) where \(\Sigma\) has at least two elements.
2. \((\mathcal{P}(X), \subseteq)\) for a set \(X\).
3. \((\Omega X, \subseteq)\) for a topological space \((X, \Omega X)\).

Definition 5.3. The opposite of a partial order \((A, \leq)\) is the partial order \((A, \leq)^{\text{op}} := (A, \geq)\) where \(a \geq b\) iff \(b \leq a\).

We often just write \(A^{\text{op}}\) for the opposite of \((A, \leq)\). Opposites are called duals in [DP02] and are denoted \((A, \leq)^{\partial}\). We refer to [DP02, §1.19 & §1.20] for further comments on opposites.

Definition 5.4 (Monotone Function). Consider partial orders \((A, \leq_A)\) and \((B, \leq_B)\) and a function \(f : A \to B\).

(a) We say that \(f\) is monotone if \(f(a) \leq_B f(a')\) whenever \(a \leq_A a'\).

(b) We say that \(f\) is antimonotone if \(f(a') \leq_B f(a)\) whenever \(a \leq_A a'\).

In other words, a function \(A \to B\) is antimonotone iff it is monotone as a function \(A^{\text{op}} \to B\).
5.2 Complete Lattices

Definition 5.5. Let \((A, \leq)\) be a partial order and consider some set \(S \subseteq A\).

1. An upper bound of \(S\) is some \(b \in A\) such that \(s \leq b\) for all \(s \in S\).

2. A least upper bound (or join) of \(S\) is an upper bound \(\bigvee S\) such that \(\bigvee S \leq b\) for every upper bound \(b\) of \(S\).

A lower bound of \(S\) is an upper bound of \(S\) in \((A, \leq)\) \(\text{op}\). A greatest lower bound (or meet) \(\bigwedge S\) of \(S\) is a least upper bound of \(S\) in \((A, \leq)\) \(\text{op}\).

In words, \(b \in A\) is a lower bound of \(S\) iff \(b \leq s\) for all \(s \in S\), and \(\bigwedge S\) is a lower bound of \(S\) such that \(b \leq \bigwedge S\) for all lower bound \(b\) of \(S\). We refer to [DP02, Def. 2.1] for a slightly more elaborated definition of (least) upper and (greatest) lower bounds.

Remark 5.6. By antisymmetry, joins and meets are unique whenever they exist.

Remark 5.7 (On \(\mathcal{P}(X)\) and \(\Omega X\)). It is easy to see that \((\mathcal{P}(X), \subseteq)\) has all meets and joins, given respectively by intersections and unions.

For \((X, \Omega X)\) a topological space, it follows from the definition that \((\Omega X, \subseteq)\) has all joins. But does it have all meets? This question may be seen as a motivation for the following definition.

Definition 5.8. A complete lattice is a partial order \((L, \leq)\) such that every subset \(S \subseteq L\) has both a join (i.e. least upper bound) \(\bigvee S \in L\) and a meet (i.e. greatest lower bound) \(\bigwedge S \in L\).

Note that a complete lattice \((L, \leq)\) has in particular a least element \(\bot = \bigvee \emptyset \in L\) and a greatest element \(\top = \bigwedge \emptyset \in L\). We repeat that by antisymmetry, joins and meets are unique.

Example 5.9. Given a set \(A\), the set \((\mathcal{P}(A), \subseteq)\) is a complete lattice.

The notion of complete lattice of [DP02, Def. 2.4] relies on the following, which can be rephrased as a consequence of [DP02, Thm. 2.31].

Lemma 5.10. The following are equivalent for a partial order \((L, \leq)\):

(i) \((L, \leq)\) is a complete lattice,

(ii) every subset \(S \subseteq L\) has a join \(\bigvee S \in L\),

(iii) every subset \(S \subseteq L\) has a meet \(\bigwedge S \in L\).

Proof. Exercise! □

A particular case of complete lattices are the frames. They simply abstract the lattice structure of open sets. This apparently candid notion is the basis of considerable developments, see e.g. [Joh82].
Definition 5.11. A frame is a partial order \((L, \leq)\) which has finite meets and all joins, and which satisfies the following infinite distributive law, where \(S\) is an arbitrary subset of \(L\):

\[ a \land \bigvee S = \bigvee \{a \land s \mid s \in S\} \]

Corollary 5.12. Every frame \((L, \leq)\) is a complete lattice.

Example 5.13. For a topological space \((X, \Omega_X)\), the partial order \((\Omega_X, \subseteq)\) is a frame where finite meets are given by finite intersections and joins are given by unions.

Recall that by antisymmetry, meets (and joins) in a partial order are unique whenever they exist. In particular, for a frame \((L, \leq, \land, \lor)\), we have

\[ a \land b = \bigvee \{c \in L \mid c \leq a \text{ and } c \leq b\} \]

Corollary 5.14. For a topological space \((X, \Omega_X)\), the partial order \((\Omega_X, \subseteq)\) is a complete lattice.

Beware that meets of open sets are in general not given by intersections!

Example 5.15. Consider the space \(A^\omega\) for \(A = \{a, b\}\). The set \(S = \bigcap_{n \in \mathbb{N}} \text{ext}(a^n)\) is not open.

Proof. Exercise!

Given a topological space \((X, \Omega_X)\), following the proof of Lem. 5.10, the meet in \(\Omega_X\) of a family of open sets \(S \subseteq \Omega_X\) is given by

\[ \bigwedge S := \bigcup \{U \in \Omega_X \mid \forall V \in S, U \subseteq V\} \]

In other words, \(\bigwedge S\) is the largest open set contained in \(\bigcap S\). This generalizes to the following usual notion.

Definition 5.16 (Interior (see e.g. [Wil70, Def. 3.9] or [Run05, Def. 2.2.22])). Given a topological space \((X, \Omega_X)\), the interior of a set \(A \subseteq X\) is

\[ \hat{A} := \bigcup \{U \in \Omega_X \mid U \subseteq A\} \]

We state the following obvious fact, and refer to [Wil70, §3.9–12] for further material.

Lemma 5.17. Given a topological space \((X, \Omega_X)\), the interior \(\hat{A}\) of \(A \subseteq X\) is the largest open set contained in \(A\).
5.3 Closure Operators

Definition 5.18 ([DP02, Def. 7.1]). A closure operator on a partial order \((L, \leq)\) is a function \(c : L \to L\) which is

- **monotone**: \(a \leq b\) implies \(c(a) \leq c(b)\),
- **expansive**: \(a \leq c(a)\),
- **idempotent**: \(c(c(a)) = c(a)\).

We say that an element \(a \in L\) is **closed** when \(c(a) = a\). We write \(L^c\) for the set of closed elements of \(L\).

Closure operators are in particular an abstraction of the closure operation on subsets of a topological space.

Example 5.19. Given a topological space \((X, \Omega X)\), the operation \(\overline{-}\) is a closure operator on \(\mathcal{P}(X)\) (see Rem. 4.6, §4.1).

Lemma 5.20 ([DP02, Prop. 7.2]). Consider a closure operator \(c\) on a complete lattice \((L, \leq)\). Then \(L^c\) is a complete lattice with meets \(\bigsqcap\) and joins \(\bigsqcup\) given resp. by

\[
\text{\bigsqcap} S = \bigwedge S \quad \text{and} \quad \text{\bigsqcup} S = c(\bigvee S)
\]

**Proof.** Exercise! \(\square\)

We note the following, for the sake of sharpening our intuitions.

Lemma 5.21. Consider a closure operator \(c\) on a complete lattice \((L, \leq)\). Then for all \(a \in L\) we have

\[
c(a) = \bigwedge \{c(b) \mid a \leq c(b)\}
\]

**Proof.** Exercise! \(\square\)

5.4 Galois Connections

Galois connections are the subject of [DP02, §7.23–35]. We differ on notation.

Definition 5.22. Given partial orders \((A, \leq_A)\) and \((B, \leq_B)\), a **Galois connection** 
\(g \dashv f : A \to B\) is given by a pair of functions

\[
g : A \to B \\
f : B \to A
\]

such that for all \(a \in A\) and all \(b \in B\) we have

\[
g(a) \leq_B b \quad \text{iff} \quad a \leq_A f(b)
\]

In a Galois connection \(g \dashv f\), \(g\) (resp. \(f\)) is called the **lower adjoint** (resp. **upper adjoint**).
Remark 5.23. It immediately follows from Def. 5.22 that in a Galois connection \( g \dashv f \), \( f \) is uniquely determined by \( g \) and \( g \) is uniquely determined by \( f \).

**Proof.** Exercise! \( \Box \)

Lemma 5.24 ([DP02, Lem. 7.26]). If \( g \dashv f : A \to B \) form a Galois connection, then both \( f \) and \( g \) are monotone.

**Proof.** First note that for all \( a \in A \) and all \( b \in B \), since \( g(a) \leq_B g(a) \) and \( f(b) \leq_A f(b) \), we have

\[
a \leq_A (f \circ g)(a) \quad \text{and} \quad (g \circ f)(b) \leq_B b
\]

Then for \( a \leq_A a' \) and \( b \leq_B b' \) we have

\[
a \leq_A a' \leq_A (f \circ g)(a') \quad \text{and} \quad (g \circ f)(b) \leq_B b \leq_B b'
\]

and thus

\[
g(a) \leq_B g(a') \quad \text{and} \quad f(b) \leq_A f(b')
\]

\( \Box \)

Lemma 5.25 ([DP02, Prop. 7.27]). If \( g \dashv f : A \to B \) is a Galois connection, then \( f \circ g : A \to A \) is a closure operator.

**Proof.** Exercise! \( \Box \)

We refer to §5.6 for further general properties of Galois connections and closure operators.

5.5 Prefix and Closure

We now describe the closed subsets of \( \Sigma^\omega \) by a closure operator induced by a Galois connection. This in particular gives another proof of the Decomposition Theorem 3.42 (see §5.5.1). We loosely follow the approach [BK08, Chap. 3]. Recall the definition of \( \text{Pref}(\sigma) \) from Notation 1.2 (§1.1).

Given a non-empty set \( \Sigma \), define

\[
\begin{align*}
\text{Pref} : \mathcal{P}(\Sigma^\omega) & \longrightarrow \mathcal{P}(\Sigma^*) \\
P & \mapsto \bigcup \{ \text{Pref}(\sigma) \mid \sigma \in P \}
\end{align*}
\]

\[
\begin{align*}
\text{cl} : \mathcal{P}(\Sigma^*) & \longrightarrow \mathcal{P}(\Sigma^\omega) \\
W & \mapsto \{ \sigma \in \Sigma^\omega \mid \text{Pref}(\sigma) \subseteq W \}
\end{align*}
\]

It is easy to see that these maps form a Galois connection \( \text{Pref} \dashv \text{cl} : \mathcal{P}(\Sigma^\omega) \to \mathcal{P}(\Sigma^*) \).

**Lemma 5.26.** For all \( P \subseteq \Sigma^\omega \) and all \( W \subseteq \Sigma^* \) we have

\[
\text{Pref}(P) \subseteq W \quad \text{iff} \quad P \subseteq \text{cl}(W)
\]
Proof. Exercise!

It thus follows from Lem. 5.25 that
\[ \text{cl} \circ \text{Pref} : \mathcal{P}(\Sigma^\omega) \rightarrow \mathcal{P}(\Sigma^\omega) \]
is a closure operator. Note that
\[
\text{cl}(\text{Pref}(P)) = \{ \sigma \in \Sigma^\omega \mid \text{Pref}(\sigma) \subseteq \text{Pref}(P) \}
= \{ \sigma \in \Sigma^\omega \mid \forall \hat{\sigma} \subseteq \sigma, \exists \beta \in P, \hat{\sigma} \subseteq \beta \}
\]

Proposition 5.27. Given \( P \subseteq \Sigma^\omega \), we have
\[
\overline{P} = \text{cl}(\text{Pref}(P))
\]

Proof. Exercise!

Recall Def. 3.31 (§3.2.5). Note that for a tree \( T \subseteq \Sigma^* \), the set \( \text{cl}(T) \) defined above is exactly the set of infinite paths of \( T \) (see Ex. 4.18, §4.2). We thus obtain the following.

Corollary 5.28. A subset \( C \) of \( \Sigma^\omega \) is closed if and only if \( C \) is the set of infinite paths of some tree \( T \subseteq \Sigma^* \).

Notation 5.29. Given \( P \subseteq \Sigma^\omega \) we often write \( \text{cl}(P) \) for \( \text{cl}(\text{Pref}(P)) \). With this notation, \( \text{cl}(P) \) is closure(\( P \)) in the sense of [BK08, Def. 3.26].

Proposition 5.27, together with the fact that safety properties \( P \subseteq (2^{\text{AP}})^\omega \) are the topologically closed subsets of \( (2^{\text{AP}})^\omega \) (Lem. 4.23), gives the following. A direct proof is nevertheless possible.

Corollary 5.30. An LT property \( P \subseteq (2^{\text{AP}})^\omega \) is a safety property if and only if \( P = \text{cl}(P) \).

Together with Corollary 4.26, Prop. 5.27 gives the following.

Corollary 5.31. An LT property \( P \subseteq (2^{\text{AP}})^\omega \) is a liveness property iff \( \text{cl}(P) = (2^{\text{AP}})^\omega \).

We moreover have the following.

Proposition 5.32. An LT property \( P \subseteq (2^{\text{AP}})^\omega \) is a liveness property iff \( \text{Pref}(P) = (2^{\text{AP}})^* \).

Proof. Exercise!
5.5.1 Alternative Proof of the Decomposition Theorem 3.42

The Galois connection $\text{Pref} \dashv \text{cl} : \mathcal{P}(\Sigma^\omega) \to \mathcal{P}(\Sigma^*)$ gives an alternative, more combinatorial proof of Thm. 3.42, following the lines of [BK08, Thm. 3.37]. The combinatorial content of the argument is contained in the following.

**Lemma 5.33 ([BK08, Lem. 3.36]).** Given $P, Q \subseteq \Sigma^\omega$, we have

$$\text{cl}(P \cup Q) = \text{cl}(P) \cup \text{cl}(Q)$$

Lemma 5.33 directly follows from Lem. 4.5 and Prop. 5.27. A direct (more combinatorial) proof is nevertheless possible.

**Proof.** Exercise!

**Remark 5.34.** As a consequence, $\text{cl} : \mathcal{P}(\Sigma^\omega) \to \mathcal{P}(\Sigma^\omega)$ is a Kuratowski closure operator (see Rem. 4.6).

The proof of Thm. 3.42 given in [BK08] then proceeds by the following. The decomposition has the same shape as in the general topological Thm. 4.11.

**Corollary 5.35 ([BK08, Thm. 3.37]).** For every LT property $P \subseteq (2^{\text{AP}})^\omega$, we have

$$P = \text{cl}(P) \cap (P \cup (2^{\text{AP}})^\omega \setminus \text{cl}(P))$$

where $\text{cl}(P)$ is a safety property and $P \cup (2^{\text{AP}})^\omega \setminus \text{cl}(P)$ is a liveness property.

**Proof.** Exercise!

5.6 Further Properties of Closure Operators and Galois Connections

We gather here some further properties of Galois connections and closure operators. These properties come from the fact that Galois connections and closure operators are particular cases of general notions in category theory, the notions resp. of **adjunction** and **monad**. We generally refer to [ML98] for categorical material, and give references to the corresponding statements.

We begin with the usual join and (resp. meet) preservation of lower (resp. upper adjoints).

**Lemma 5.36 ([DP02, Prop. 7.31]).** If $g \dashv f : A \to B$ is a Galois connection, then $g$ preserves any join which exists in $A$ and $f$ preserves any meet which exists in $B$.

**Proof.** Exercise!

Lemma 5.36 thus implies that $g$ (resp. $f$) preserves joins (resp. meets) whenever it has a lower adjoint (resp. an upper adjoint). This is actually a particular case of a general property of adjoints in category theory (see e.g. [Awo10, §9.5] or [ML98, §V.5]), where we speak of **left** and **right** adjoints for the generalized form of resp. lower and upper adjoints. In various occasions, one is more interested in knowing the **existence** of an adjoint, so as to deduce preservation properties, rather than in the adjoint in itself.

Interestingly, Lem. 5.36 has a converse.
Lemma 5.37 ([DP02, Prop. 7.34]). Assume that \((A, \leq_A)\) and \((B, \leq_B)\) are complete lattices.

(1) If \(f : B \to A\) preserves meets (and is thus monotone), then \(f\) has a lower adjoint \(g : A \to B\).

(2) If \(g : A \to B\) preserves joins (and is thus monotone), then \(g\) has an upper adjoint \(f : B \to A\).

Proof. Exercise!

The categorical generalization of Lem. 5.37 actually involves more complex conditions. See e.g. [ML98, §V.6] and [Awo10, §9.8].

Example 5.38 ((Complete) Heyting Algebras). Let \((A, \leq)\) be a complete lattice. Given \(a \in A\), consider the map
\[
(-) \land a : A \to A, \quad b \mapsto b \land a
\]

It follows from Lem. 5.36 and Lem. 5.37 that \((-) \land a\) has an upper adjoint if and only if \((-) \land a\) preserves all joins. Note that the latter exactly means that for all \(S \subseteq A\), we have
\[
(\bigvee S) \land a = \bigvee \{s \land a \mid s \in S\}
\]

Hence, \(A\) is a frame (Def. 5.11, §5.2) if and only if each map \((-) \land a\) (for \(a \in A\)) has an upper adjoint.

Upper adjoints to \((-) \land a\) are often denoted \(a \Rightarrow (-)\), since \((-) \land a \lhd a \Rightarrow (-)\) means
\[
(b \land a) \leq c \iff b \leq (a \Rightarrow c)
\]

so that \(a \Rightarrow c\) is reminiscent from a logical implication.

Frames are also called complete Heyting algebras. Anticipating on the terminology of §??, a Heyting algebra is a lattice \(A\) (i.e. a partial order in which has all finite joins and all finite meets) such that each map \((-) \land a\) (for \(a \in A\)) has an upper adjoint \(a \Rightarrow (-)\). Note that this implies (by Lem. 5.36) that a Heyting algebra is automatically distributive, in the sense that for all \(a, b, c \in A\) we have
\[
(c \lor b) \land a = (c \land a) \lor (b \land a)
\]

Note also that what we called a “complete Heyting algebra” is nothing else but a Heyting algebra which happens to be complete as a lattice.

Heyting algebras are the appropriate notion of truth values for intuitionistic propositional logic (see e.g. [SU06, §2.4] or [Awo10, §6.3], outside the scope of this course).

We have seen in Lem. 5.25 that Galois connections induce closure operators. The converse, namely that every Galois connections arises from a closure operator is also true. The categorical generalization of closure operators are monads. See e.g. [ML98, Chap VI].

Given a closure operator \(c : A \to A\), we already have looked at the set \(A^c\) of closed elements in §5.3.


**Lemma 5.39** ([DP02, §7.28]). Let \( c : A \to A \) be a closure operator. Then \( c : A \to A^c \) is part of a Galois connection \( c \dashv \iota : A \to A^c \), where \( \iota(a) := a \).

**Proof.** Exercise!

We of course have \( c = \iota \circ c \). The Galois connection of Lem. 5.39 generalizes to the well-known adjunction between a category \( C \) and the Eilenberg-Moore category \( C^T \) of a monad \( T \) on \( C \), see e.g. [ML98, §VI.2].

### 5.6.1 On the Kleisli Construction

The notion of closure operator on a partial order of Def. 5.18 can be generalized to **preorders**. A **preorder** on a set \( A \) is a binary relation which is reflexive and transitive. So the difference with a partial order is that antisymmetry is not required, i.e. we can have \( a \leq b \) and \( b \leq a \) with \( a \neq b \). If \( (A, \leq) \) is a preorder, we say that \( c : A \to A \) is a **closure operator** if \( c \) is monotone, expansive and such that \( c(c(a)) \leq c(a) \) for all \( a \in A \).

The definition of Galois connections between preorders is the same as for partial orders (Def. 5.22), and all properties seen in §5.4 and the present §5.6 generalize to preorders.

In this setting, for a closure operator \( c : A \to A \), let \( \leq_c \subseteq A \times A \) be such that \( a \leq_c a' \) iff \( a \leq c(a') \). The following is a particular case of a second way to generate an adjunction from a monad \( T \) on a category \( C \), namely the adjunction between \( C \) and its Kleisli category \( C_T \). We refer to e.g. [ML98, §VI.5] for details.

**Lemma 5.40.** Let \( c : A \to A \) be a closure operator on a preorder. Then \( c : A \to A \) is part of a Galois connection \( \iota \dashv c : (A, \leq) \to (A, \leq_c) \), where \( \iota(a) := a \).

**Proof.** Exercise!

### 6 Observable Properties

This Section refines the topological approach of §4, with the aim of isolating a natural notion of “observable” linear-time property. This lays the ground to logics for linear-time properties. An important point is that, when AP is finite, LT properties of the form \( \text{ext}(V) \), for a finite \( V \subseteq (2^{2^\text{AP}})^* \), form a Boolean algebra.

#### 6.1 Observable Properties as Clopen Sets

Given sets \( X \) and \( Y \), a function \( f : X \to Y \) induces a function

\[
\begin{align*}
f^{-1} : \mathcal{P}(Y) & \to \mathcal{P}(X) \\
B & \mapsto \{x \mid f(x) \in B\}
\end{align*}
\]

**Lemma 6.1.** Given a function \( f : X \to Y \), the function \( f^{-1} : \mathcal{P}(Y) \to \mathcal{P}(X) \) is a map of complete Boolean algebras from \( (\mathcal{P}(Y), \bigcap, \bigcup, Y \setminus (-), Y, \emptyset) \) to \( (\mathcal{P}(X), \bigcap, \bigcup, X \setminus (-), X, \emptyset) \).
Note that if \( f^{-1} \) preserves unions and intersections, then it also preserves complements, as the complement of \( A \in \mathcal{P}(X) \) is the unique \( B \in \mathcal{P}(X) \) such that \( A \cup B = X \) and \( A \cap B = \emptyset \).

**Proof.** Exercise! \( \square \)

**Definition 6.2** (Continuous Function). Consider topological spaces \((X, \Omega_X)\) and \((Y, \Omega_Y)\).

1. A function \( f : X \to Y \) is **continuous** if \( f^{-1} : \mathcal{P}(Y) \to \mathcal{P}(X) \) restricts to a function \( \Omega_Y \to \Omega_X \), i.e. if \( f^{-1}(V) \) is open in \( X \) whenever \( V \) is open in \( Y \).

2. We say that \( f : X \to Y \) is an **homeomorphism** if \( f \) is a continuous bijection with continuous inverse \( Y \to X \).

**Lemma 6.3.** A function \( f : \Sigma^\omega \to \Gamma^\omega \) is continuous iff

\[
\forall n \in \mathbb{N}, \forall \alpha \in \Sigma^\omega, \exists k \in \mathbb{N}, \forall \beta \in \Sigma^\omega \left( \beta(0) \cdots \beta(k) = \alpha(0) \cdots \alpha(k) \implies f(\beta)(0) \cdots f(\beta)(n) = f(\alpha)(0) \cdots f(\alpha)(n) \right)
\]

**Proof.** Exercise! \( \square \)

In words, a continuous stream function must be able to produce a finite part of its output from a finite part of its input. It is generally admitted that a computable function on streams must be continuous. In particular, a necessary condition for an LT property \( P \subseteq (2^{AP})^\omega \) to be decidable is to have a continuous characteristic function

\[
\chi_P : (2^{AP})^\omega \to 2 \\
\alpha \mapsto \begin{cases} 
1 & \text{if } \alpha \in P \\
0 & \text{otherwise}
\end{cases}
\]

where 2 is endowed with the **discrete** topology, with which every subset is open. This amounts to ask that both \( P \) and \((2^{AP})^\omega \setminus P\) are open, or equivalently that \( P \) is both open and closed.

**Definition 6.4** (Clopen Set). A subset of a topological space is **clopen** if it is both open and closed.

**Lemma 6.5** (The Boolean Algebra of Clopens). Let \((X, \Omega_X)\) be a topological space. The clopens of \( X \) form a sub-Boolean algebra of \((\mathcal{P}(X), (\cdot) \cap (\cdot), (\cdot) \cup (\cdot), X \setminus (\cdot), X, \emptyset)\).

**Proof.** First, clopens are evidently closed under complements. Furthermore, both \( \emptyset \) and \( X \) are clopens. Finally, the open subsets and the closed subsets of \( X \) are closed under binary intersections and binary unions. \( \square \)

**Definition 6.6** (Observable Property). An LT property \( P \subseteq (2^{AP})^\omega \) is **observable** if \( P \) is a clopen subset of \((2^{AP})^\omega\).

Let us look more precisely at the observable properties.
6 Observable Properties

Lemma 6.7. In $\Sigma^\omega$, each set of the form $\text{ext}(u)$ for $u \in \Sigma^*$ is clopen.

Proof. Exercise! $\square$

As a consequence, each finite subset $U \subseteq \Sigma^*$ induces a clopen set $\text{ext}(U) = \bigcup_{u \in U} \text{ext}(u)$. However, the converse is not true in general.

Example 6.8. Consider the Baire space $\mathcal{N} := \mathbb{N}^\omega$. The subset $P \subseteq \mathbb{N}^\omega$ given by

$$P := \bigcup_{n > 0} \text{ext}(n)$$

is obviously open. It is also closed as being the complement of $\text{ext}(0)$. But $P$ cannot be presented as the extension of a finite set $U \subseteq \mathbb{N}^*$.

We shall see in Prop. 6.15 that when $\text{AP}$ is finite, the observable $P \subseteq (2^{\text{AP}})^\omega$ are exactly the sets of the form $\bigcup_{u \in U} \text{ext}(u)$ for a finite $U \subseteq (2^{\text{AP}})^\omega$. This relies on a strong topological properties of $(2^{\text{AP}})^\omega$ for finite $\text{AP}$, known as compactness, and whose most basic aspects are presented in §6.2 and §6.3.

6.2 Compactness

We follow here parts of the presentation of [Run05].

Definition 6.9. Let $(X, \Omega_X)$ be a topological space.

- An open cover of a set $A \subseteq X$ is a family of open sets $(U_i)_{i \in I}$ such that $A \subseteq \bigcup_{i \in I} U_i$.
- A set $A \subseteq X$ is compact in $X$ if every open cover $(U_i)_{i \in I}$ of $A$, contains a finite cover of $A$, i.e. there is a finite set $J \subseteq I$ such that $A \subseteq \bigcup_{j \in J} U_j$.
- The space $(X, \Omega_X)$ is compact if $X$ is itself a compact subset of $X$.

The following is a simple consequence of the definitions. In the case of compact Hausdorff spaces (§6.3) it becomes part of a powerful characterization of the compact sets (see Prop. 6.20).

Lemma 6.10. A closed subset of a compact space is compact.

Proof. Let $(X, \Omega_X)$ be a compact space and let $C \subseteq X$ be closed. Given an open covering $U = (U_i)_{i \in I}$ of $C$, we obtain with $U \cup \{X \setminus C\}$ an open covering of $X$. Since $X$ is compact, it has a finite subcover $V \cup \{X \setminus C\}$ where $V = (U_j)_{j \in J}$ for some finite $J \subseteq I$. But then $V$ covers $C$. $\square$

In the case of $\omega$-words, the space $\Sigma^\omega$ is compact if and only if $\Sigma$ is finite. First, it is easy to see that $\Sigma^\omega$ is not be compact when $\Sigma$ is infinite.

Lemma 6.11. Consider the space of $\omega$-words $\Sigma^\omega$ for some non-empty set $\Sigma$. If $\Sigma$ is infinite, then $\Sigma^\omega$ is not compact.
6 Observable Properties

Proof. Indeed, we have

\[ \Sigma^\omega = \bigcup_{a \in \Sigma} \text{ext}(a) \]

But if \( \Sigma \) is infinite, one cannot extract a finite subcover of \( \Sigma^\omega \) from \( \text{(ext}(a))_{a \in \Sigma} \). \( \square \)

We now show that \( \Sigma^\omega \) is compact when \( \Sigma \) is finite. We rely on König’s Lemma 3.32 (§3.2.5).

**Proposition 6.12.** Let \( \Sigma \) be a finite non-empty set. Then \( \Sigma^\omega \) is compact.

**Proof.** Consider an open covering \( (U_i)_{i \in I} \) of \( \Sigma^\omega \). Note that each \( U_i \) is of the form \( \bigcup_{v \in V_i} \text{ext}(v) \) for some \( V_i \subseteq \Sigma^* \). Let \( V := \bigcup_{i \in I} V_i \). We build a prefix-free \( W \subseteq V \) as

\[ W = \bigcup_{n \in \mathbb{N}} W_n, \]

where

1. \( \varepsilon \in W_0 \) iff \( \varepsilon \in V \).
2. Given \( u \in \Sigma^* \) of length \( n + 1 \), we let \( u \in W_{n+1} \) if \( u \in V \) and \( u \) has no prefix in \( \bigcup_{k \leq n} W_k \).

It is clear that \( W \) is prefix-free, in the sense that if \( u \in W \) then \( u \) has no strict prefix in \( W \). Moreover, each \( v \in V \) has a prefix in \( W \). Hence, recalling that \( w \subseteq v \) implies \( \text{ext}(v) \subseteq \text{ext}(w) \) (Lem. 4.16), \( W \) induces a cover of \( \Sigma^\omega \) as

\[ \Sigma^\omega = \bigcup_{v \in V} \text{ext}(v) = \bigcup_{w \in W} \text{ext}(w) \]

Hence we are done if \( W \) is finite. Assume toward a contradiction that \( W \) is infinite. Let \( T \subseteq \Sigma^* \) be the prefix-closure of \( W \) (i.e. \( u \in T \) iff \( u \subseteq w \) for some \( w \in W \)). Then \( T \) is finitely branching as \( \Sigma \) is finite, and \( T \) is infinite as \( W \) is infinite. Hence, by König’s Lemma 3.32, \( T \) has a path \( \pi \). Since \( W \) induces a cover of \( \Sigma^\omega \), we have \( w \subseteq \pi \) for some \( w \in W \). Since \( \pi \) is a path in \( T \), we have \( w.a \subseteq \pi \) for some \( w.a \in T \) with \( a \in \Sigma \). By definition of \( T \), we must have \( w.a \subseteq v \) for some \( v \in W \), but this is impossible since \( v \in W \) would then have a strict prefix \( w \in W \). \( \square \)

**Remark 6.13** (Tychonoff Theorem – Compactness of Product Spaces). The conjunction of Lem. 6.11 with Prop. 6.12 is an instance of Tychonoff Theorem. We refer to [Wil70, Thm. 17.8] and to [Run05, Thm. 3.3.21]. It is an easy exercise to show that “our” topology on \( \Sigma^\omega \) is the product topology in the usual sense (taking \( \Sigma \) discrete), see e.g. [Wil70, Chap. 3, §8] or [Run05, Def. 3.3.19]. Tychonoff Theorem is known to be equivalent to the Axiom of Choice. In the simple case of \( \Sigma^\omega \), we used König’s Lemma 3.32, a much weaker principle of infinite combinatorics (see e.g. [Sim10, Ex. I.8.8]).

**Remark 6.14.** It is possible to prove Prop. 6.12 without explicitly relying on König’s Lemma 3.32 (see e.g. [PP04, §III.3.5]). Actually, one can obtain König’s Lemma 3.32 from a strengthening of Prop. 6.12 stating the relative compactness of subspaces of \( \Sigma^\omega \) (with \( \Sigma \) possibly infinite). See e.g. [PP04, Ex. III.8.6] (outside the scope of this course).

**Proposition 6.15** (Observable Property – The Compact Case). If \( \text{AP} \) is finite, then \( P \subseteq (2^{\text{AP}})^\omega \) is observable iff \( P = \bigcup_{u \in U} \text{ext}(u) \) for some finite \( U \subseteq (2^{\text{AP}})^* \).
Proof. We already know from Lem. 6.7 and Lem. 6.5 that the condition is sufficient. Let \( P \subseteq (2^{AP})^\omega \) be clopen, hence compact open. Then \( P = \bigcup_{u \in U} \text{ext}(u) \) for some \( U \subseteq (2^{AP})^\ast \). Since \( P \) is compact, there is a finite subset \( V \subseteq U \) such that \( P \subseteq \bigcup_{u \in V} \text{ext}(u) \). But this implies \( P = \bigcup_{u \in V} \text{ext}(u) \) as \( V \subseteq U \).

### 6.2.1 The Finite Intersection Property

The following characterization of compact spaces is useful in practice. It directly follows from the definitions.

**Definition 6.16** (Finite Intersection Property). Given a set \( A \), a family of sets \( F \subseteq \mathcal{P}(A) \) has the **finite intersection property** for every finite \( F \subseteq F \), we have \( \bigcap F \neq \emptyset \).

**Lemma 6.17.** A space \( (X, \Omega_X) \) is compact iff for every family of closed sets \( F \) with the finite intersection property, we have \( \bigcap F \neq \emptyset \).

**Proof.** Exercise!

### 6.3 Compact Hausdorff Spaces

Compact spaces with the following separation property enjoy a particularly simple characterization of their compact subsets. See e.g. [Wil70, Chap. 5, §13] or [Run05, Def. 3.13].

**Definition 6.18** (Hausdorff Space). A topological space \( (X, \Omega_X) \) is **Hausdorff** (or \( T_2 \)) if for any distinct points \( x, y \in X \), there are disjoint opens \( U, V \) such that \( x \in U \) and \( y \in V \).

**Example 6.19.** Spaces of \( \omega \)-words \( \Sigma^\omega \) are Hausdorff.

Here comes the announced characterization of the compacts subsets of a compact Hausdorff space. Recall from Lem. 6.10 that the closed subsets of a compact spaces are always compact.

**Proposition 6.20.** In an Hausdorff space, each compact set is closed.

**Proof.** Consider an Hausdorff space \( (X, \Omega_X) \) and fix a compact set \( C \subseteq X \). We show that \( C \) is closed using Lem. 4.3. So let \( x \notin C \). Since \( X \) is Hausdorff, for each \( y \in C \) there are disjoint open sets \( U_y, V_y \) such that \( x \in U_y \) and \( y \in V_y \). Hence \( (V_y)_{y \in C} \) is an open cover of \( C \). Since \( C \) is compact, \( (V_y)_{y \in C} \) has a finite subcover, say \( V_{y_1}, \ldots, V_{y_n} \). But then \( x \) belongs to the open set \( U := U_{y_1} \cap \cdots \cap U_{y_n} \). Moreover, since each \( U_{y_i} \) is disjoint from \( V_{y_i} \), it follows that \( U \) is disjoint from each \( V_{y_i} \) and thus from \( C \subseteq V_{y_1} \cup \cdots \cup V_{y_n} \).

As a consequence, in a compact Hausdorff space, the compact sets are exactly the closed sets, and the clopen sets are exactly the compact open sets.
7 Linear Temporal Logic

Linear Temporal Logic (LTL) is a modal logic to express linear-time properties. In the field of computer science, temporal logics for linear-times properties were introduced by [Pnu77]. We refer to [BdRV02] for a comprehensive introduction to modal logic.

This Section presents LTL in a step-wise manner, starting from the notion of observable property drawn in §6. We mostly build from [BK08, Chap. 5], but differ in various aspects. In particular, we discuss standard material on the computation of fixpoints in complete lattices which goes beyond [BK08] and for which we mainly refer to [DP02].

7.1 The Logic LML of Observable Properties

Fix a set AP of atomic propositions. We are going to define a linear-time modal logic LML such that, when AP is finite, the formulae of LML describe exactly the clopens of \((2^{AP})^\omega\).

7.1.1 Syntax and Semantics of LML

We assume given a countably infinite set \(X = \{X, Y, Z, \ldots\}\) of variables. The formulae of LML are given by the following grammar:

\[
\begin{align*}
\varphi, \psi & ::= \top \mid \bot \mid X \mid a \quad (\text{where } X \in X \text{ and } a \in \text{AP}) \\
& \mid \varphi \land \psi \mid \varphi \lor \psi \mid \neg \varphi \\
& \mid \bigcirc \varphi
\end{align*}
\]

The formulae of LML are to be interpreted as subsets of \((2^{AP})^\omega\). In particular, the interpretation of a formula \(\varphi\) with variables among \(X_1, \ldots, X_n\) depends on a valuation of the \(X_i\)'s as sets \(A_i \subseteq (2^{AP})^\omega\).

**Definition 7.1** (Valuations and Formulae with Parameters).

1. A **valuation** of a set of variables \(V \subseteq X\) is a function \(\rho : V \to \mathcal{P}((2^{AP})^\omega)\).
2. A **formula with parameters** is a pair \((\varphi, \rho)\) of a formula \(\varphi\) and a valuation \(\rho : V \to \mathcal{P}((2^{AP})^\omega)\) where \(V\) contains all the variables of \(\varphi\).

We often speak of a **formula \(\varphi\) with parameters** \(\rho\) for the pair \((\varphi, \rho)\).

Consider a formula \(\varphi\) with parameters \(\rho\). We define the interpretation \([\varphi]_\rho \subseteq (2^{AP})^\omega\) by induction on \(\varphi\) as follows:

\[
\begin{align*}
[X]_\rho & := \rho(X) \\
[a]_\rho & := \{\sigma \in (2^{AP})^\omega \mid a \in \sigma(0)\} \\
[\top]_\rho & := (2^{AP})^\omega \\
[\bot]_\rho & := \emptyset \\
[\varphi \land \psi]_\rho & := [\varphi]_\rho \cap [\psi]_\rho \\
[\varphi \lor \psi]_\rho & := [\varphi]_\rho \cup [\psi]_\rho \\
[\neg \varphi]_\rho & := (2^{AP})^\omega \setminus [\varphi]_\rho \\
[\bigcirc \varphi]_\rho & := \{\sigma \in (2^{AP})^\omega \mid \sigma[1] \in [\varphi]_\rho\}
\end{align*}
\]
where, for \( i \in \mathbb{N} \), \( \sigma|_i \in (2^{AP})^\omega \) is the function which takes \( k \in \mathbb{N} \) to \( \sigma(i + k) \in 2^{AP} \), e.g.

\[
\sigma = \sigma(0) \cdot \sigma(1) \cdot \ldots \cdot \sigma(n) \cdot \ldots
\]

\[
\sigma|_1 = \sigma(1) \cdot \sigma(2) \cdot \ldots \cdot \sigma(n+1) \cdot \ldots
\]

**Notation 7.2.** Other propositional connectives are defined as usual:

\[
\varphi \to \psi := \neg \varphi \lor \psi
\]

\[
\varphi \leftrightarrow \psi := (\varphi \to \psi) \land (\psi \to \varphi)
\]

**Definition 7.3.** We say that \( \sigma \in (2^{AP})^\omega \) **satisfies** a formula \( \varphi \) with parameters \( \rho \) if \( \sigma \in J_{\varphi}^K \rho \).

**Lemma 7.4.** If \( \rho = \rho' \) for all variables which actually occur in \( \varphi \), then \( [\varphi]_\rho = [\varphi]_{\rho'} \).

In particular, if \( \varphi \) is **closed**, i.e. contains no free variable, then \( [\varphi]_\rho \) does not depend on \( \rho \). In this case, we just write \( [\varphi]_\rho \) for \( [\varphi]_\rho \).

**Notation 7.5.** For a closed \( \varphi \), we write \( \sigma \vdash \varphi \) for \( \sigma \in J_{\varphi}^K \rho \).

The relation \( \sigma \vdash \varphi \) (for \( \varphi \) closed) can be given an inductive definition.

**Remark 7.6.** The relation \( \sigma \vdash \varphi \) is the least relation such that

\[
\sigma \vdash a \quad \text{iff} \quad a \in \sigma(0)
\]

\[
\sigma \vdash \top
\]

\[
\sigma \not\vdash \bot
\]

\[
\sigma \vdash \varphi \land \psi \quad \text{iff} \quad \sigma \vdash \varphi \text{ and } \sigma \vdash \psi
\]

\[
\sigma \vdash \varphi \lor \psi \quad \text{iff} \quad \sigma \vdash \varphi \text{ or } \sigma \vdash \psi
\]

\[
\sigma \vdash \neg \varphi \quad \text{iff} \quad \sigma \not\vdash \varphi
\]

\[
\sigma \vdash \Box \varphi \quad \text{iff} \quad \sigma|_1 \vdash \varphi
\]

### 7.1.2 Logical Equivalence

**Definition 7.7** (Logical Equivalence). Given formulae \( \varphi \) and \( \psi \) with free variables in \( V \subseteq X \), we say that \( \varphi \) and \( \psi \) are **logically equivalent** (notation \( \varphi \equiv \psi \)) if for all valuation \( \rho : V \to P((2^{AP})^\omega) \) we have

\[
[\varphi]_\rho = [\psi]_\rho
\]

**Lemma 7.8.** All the equivalences of Fig. 6 hold.

**Remark 7.9.** The notion of logical equivalence \( \equiv \) given in Def. 7.7, for which we follow [BK08, Def. 5.17], is **not** the usual one (see e.g. [BdRV02, Def. 5.29]). In Def. 7.7 as well as in [BK08, Def. 5.17], logical equivalence is defined as a **semantic** equivalence, whereas [BdRV02, Def. 5.29] defines \( \equiv \) as **provable** equivalence in a given axiomatic system.

While the “right” notion of logical equivalence is that of [BdRV02, Def. 5.29] (see also e.g. [DP02, §11.11–16]), we stick to the semantic notion of [BK08, Def. 5.17] since the present notes do not cover axiomatic and deductive approaches to logic.
Semilattices Laws:

\[ \phi \lor \phi \equiv \phi \quad \phi \land \phi \equiv \phi \]
\[ \phi \lor \psi \equiv \psi \lor \phi \quad \phi \land \psi \equiv \psi \land \phi \]
\[ \phi \lor \bot \equiv \phi \quad \phi \land \top \equiv \phi \]
\[ \phi \lor (\psi \land \theta) \equiv (\phi \lor \psi) \land \theta \quad \phi \land (\psi \land \theta) \equiv (\phi \land \psi) \land \theta \]

Absorptive Laws (Lattice Laws):

\[ \phi \lor (\phi \land \psi) \equiv \phi \]
\[ \phi \land (\phi \lor \psi) \equiv \phi \]

Distributive Laws:

\[ \phi \lor (\psi \land \theta) \equiv (\phi \lor \psi) \land (\phi \lor \theta) \]
\[ \phi \land (\psi \lor \theta) \equiv (\phi \land \psi) \lor (\phi \land \theta) \]

Boolean Algebra Laws:

\[ \phi \land \neg \phi \equiv \bot \quad \phi \lor \neg \phi \equiv \top \]

Duality (De Morgan) Laws:

\[ \phi \land \psi \equiv \neg (\neg \phi \lor \neg \psi) \quad \phi \lor \psi \equiv \neg (\neg \phi \land \neg \psi) \quad \phi \equiv \neg \neg \phi \]

Modal Laws:

\[ \Box (\phi \land \psi) \equiv \Box \phi \land \Box \psi \quad \Box \top \equiv \top \]
\[ \Box (\phi \lor \psi) \equiv \Box \phi \lor \Box \psi \quad \Box \bot \equiv \bot \]
\[ \Box (\neg \phi) \equiv \neg \Box \phi \]

Figure 6: Some Usual Laws.
7 Linear Temporal Logic

7.1.3 Observable Properties

We now turn to the promised fact that when AP is finite, the closed formulae of LML exactly correspond to the observable (i.e. clopen) properties on \((2^{\text{AP}})^\omega\). Recall from Prop. 6.15 that when AP is finite, the clopen subsets of \((2^{\text{AP}})^\omega\) are exactly the finite unions of sets of the form \(\text{ext}(\hat{\sigma})\) for \(\hat{\sigma} \in (2^{\text{AP}})^*\). Recall moreover from Lem. 6.5 that clopen sets are closed under complements, finite unions and finite intersections.

**Proposition 7.10.** For each closed LML-formula \(\varphi\), \(\mathcal{[\varphi]}\) is a clopen subset of \((2^{\text{AP}})^\omega\).

**Proof.** Exercise! \hfill \(\square\)

**Proposition 7.11.** Assume that AP is finite. Then for any clopen \(P \subseteq (2^{\text{AP}})^\omega\) there is a closed LML-formula \(\varphi\) such that \(P = \mathcal{[\varphi]}\).

**Proof.** Exercise! \hfill \(\square\)

Proposition 7.11 may not hold when AP is infinite.

**Example 7.12.** Let \(\text{AP} := \mathbb{N}\) and let \(2\mathbb{N} \subseteq \text{AP}\) consist of the even numbers. Note that \(\sigma \in \text{ext}(2\mathbb{N})\) iff \(\sigma(0) = 2\mathbb{N}\) and that \(\text{ext}(2\mathbb{N})\) is clopen by Lem. 6.7. It is easy to see that there is no closed formula \(\varphi\) such that \(\text{ext}(2\mathbb{N}) = \mathcal{[\varphi]}\).

**Proof.** Exercise! \hfill \(\square\)

7.2 Extending LML with Fixpoints

As seen in Prop. 7.10, the logic LML has a very limited expressive power. In particular, it can only express few safety properties, and it follows from Prop. 3.41 that the only expressible liveness property is the “true property” \((2^{\text{AP}})^\omega\). We shall therefore look for extensions of LML.

7.2.1 The “Eventually” and “Always” Modalities

Typical logical constructs one may wish to add to LML are the **Eventually** and **Always** modalities, noted resp. \(\diamond \varphi\) and \(\Box \varphi\), and with

\[
\begin{align*}
\mathcal{[\diamond \varphi]}_\rho & := \{ \sigma \in (2^{\text{AP}})^\omega \mid \exists i \in \mathbb{N}, \sigma|_i \in \mathcal{[\varphi]}_\rho \} \\
\mathcal{[\Box \varphi]}_\rho & := \{ \sigma \in (2^{\text{AP}})^\omega \mid \forall i \in \mathbb{N}, \sigma|_i \in \mathcal{[\varphi]}_\rho \}
\end{align*}
\]

In the spirit of Notation 7.5, for a closed \(\varphi\) we write

\[
\sigma \models \diamond \varphi \iff \exists i \in \mathbb{N}, \sigma|_i \models \varphi
\]

\[
\sigma \models \Box \varphi \iff \forall i \in \mathbb{N}, \sigma|_i \models \varphi
\]

**Example 7.13.** Let \(a \in \text{AP}\).

(1) \(\sigma \models \diamond a\) iff \(a \in \sigma(i)\) for some \(i \in \mathbb{N}\).

The formula \(\diamond a\) defines an open liveness property \(\mathcal{[\diamond a]} \subseteq (2^{\text{AP}})^\omega\).
7 Linear Temporal Logic

(2) \( \sigma \vdash \Box a \iff a \in \sigma(i) \) for all \( i \in \mathbb{N} \).

The formula \( \Box a \) defines a safety property \( \Box a \subseteq (2^{\text{AP}})^\omega \).

(3) \( \sigma \vdash \Diamond a \iff a \notin \sigma(i) \) for infinitely many \( i \in \mathbb{N} \).

Note that \( \Box \Diamond \varphi \rightarrow \Diamond \Box \varphi \) is always true. The formulae \( \Box \Diamond a \) and \( \Diamond \Box a \) define liveness properties \( \Diamond \Box a, \Box \Diamond a \subseteq (2^{\text{AP}})^\omega \) which are not closed nor open.

Let us investigate the semantics of \( \Diamond \varphi \) and \( \Box \varphi \), with the aim of looking for behaviour which could be easily generalized. First note the following basic equivalences, where \( \equiv \) stands for the obvious extension of Def. 7.7.

**Lemma 7.14.** We have

\[
\begin{align*}
\Diamond \varphi & \equiv \neg \Box \neg \varphi \\
\Box \varphi & \equiv \neg \Diamond \neg \varphi \\
\Diamond \Diamond \varphi & \equiv \varphi \lor \Diamond \Diamond \varphi \\
\Box \Diamond \varphi & \equiv \varphi \land \Box \Diamond \varphi
\end{align*}
\]

Intuitively, the first two equivalences of Lem. 7.14 say that \( \Diamond \) and \( \Box \) can be seen as De Morgan duals of each other. The last two could be rephrased as follows.

- \( \Diamond \varphi \) holds at the current time step iff either \( \varphi \) holds at the current time step or \( \Diamond \varphi \) holds at the next time step.

- \( \Box \varphi \) holds at the current time step iff \( \varphi \) holds at the current time step and \( \Box \varphi \) holds at the next time step.

If we allowed for formulae with infinite disjunctions and conjunctions, we could state

\[
\begin{align*}
\Diamond \varphi & \equiv \varphi \lor \Diamond \varphi \lor \Diamond \Diamond \varphi \lor \cdots & \equiv \bigvee_{n \in \mathbb{N}} \Diamond \Diamond \cdots \Diamond \varphi \\
\Box \varphi & \equiv \varphi \land \Box \varphi \land \Box \Diamond \varphi \land \cdots & \equiv \bigwedge_{n \in \mathbb{N}} \Diamond \Diamond \cdots \Diamond \varphi
\end{align*}
\]

We shall rather look for finitary representations of such infinite behaviors, with extensions of LML with fixpoints of functions \( \mathcal{P}((2^{\text{AP}})^\omega) \to \mathcal{P}((2^{\text{AP}})^\omega) \) induced by formulae as follows.

**Notation 7.15.** Given a formula \( \varphi \) with parameters \( \rho \) and a variable \( X \), we write \( \llbracket \varphi \rrbracket_\rho(X) \) for the function

\[
\llbracket \varphi \rrbracket_\rho(X) : \mathcal{P}((2^{\text{AP}})^\omega) \to \mathcal{P}((2^{\text{AP}})^\omega)
\]

**Lemma 7.16.** Let \( \varphi \) be a formula with parameters \( \rho \). Consider the formulae

\[
\varphi_{\Diamond}(X) := \varphi \lor \Diamond X \qquad \varphi_{\Box}(X) := \varphi \land \Box X
\]

Then we have:
### 7.2.2 Positive and Negative Variables in a Formula

We note here the simple fact that if the variable $X$ occurs under an even (resp. odd) number of negations in $\varphi$, then $\llbracket \varphi \rrbracket_\rho(X)$ is a monotone (resp. antimonotone) function of $(\mathcal{P}((2^{AP})^\omega), \subseteq)$.

We use the following inductive notions of positive (resp. negative) variable in a formula $\varphi$ in order to express that a variable occurs under an even (resp. odd) number of negations in $\varphi$.

**Definition 7.17 (Positive Negative Variables).** Given an LML-formula $\varphi$ and a variable $X$, the relations $X \text{ Pos } \varphi$ ($X$ is **positive** in $\varphi$) and $X \text{ Neg } \varphi$ ($X$ is **negative** in $\varphi$) are defined by induction on Fig. 7.

**Lemma 7.18.** Consider a formula $\varphi$ with parameters $\rho$ and a variable $X$.

1. If $X \text{ Pos } \varphi$, then $\llbracket \varphi \rrbracket(X)$ is a monotone function on $(\mathcal{P}((2^{AP})^\omega), \subseteq)$.
2. If $X \text{ Neg } \varphi$, then $\llbracket \varphi \rrbracket(X)$ is an antimonotone function on $(\mathcal{P}((2^{AP})^\omega), \subseteq)$.

---

**Proof.** Exercise! □

---

Figure 7: Positive and Negative Occurrences in a Formula.

<table>
<thead>
<tr>
<th>$X \text{ Pos } \varphi$</th>
<th>$X \text{ Pos } \psi$</th>
<th>$X \text{ Pos } \varphi \land \psi$</th>
<th>$X \text{ Pos } \varphi \lor \psi$</th>
<th>$X \text{ Neg } \varphi$</th>
<th>$X \text{ Neg } \psi$</th>
<th>$X \text{ Neg } \varphi \land \psi$</th>
<th>$X \text{ Neg } \varphi \lor \psi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X \text{ Pos } X$</td>
<td>$X \text{ Pos } Y$</td>
<td>$X \text{ Pos } a$</td>
<td>$X \text{ Pos } \top$</td>
<td>$X \text{ Pos } \bot$</td>
<td>$X \text{ Neg } \top$</td>
<td>$X \text{ Neg } \bot$</td>
<td>$X \text{ Neg } \circ \varphi$</td>
</tr>
</tbody>
</table>

(1) $\llbracket \Diamond \varphi \rrbracket_\rho$ is the least element of $(\mathcal{P}((2^{AP})^\omega), \subseteq)$ such that

\[
\llbracket \Diamond \varphi \rrbracket_\rho = \llbracket \varphi \rrbracket_\rho(\llbracket \Diamond \varphi \rrbracket_\rho)
\]

(2) $\llbracket \Box \varphi \rrbracket_\rho$ is the greatest element of $(\mathcal{P}((2^{AP})^\omega), \subseteq)$ such that

\[
\llbracket \Box \varphi \rrbracket_\rho = \llbracket \varphi \rrbracket_\rho(\llbracket \Box \varphi \rrbracket_\rho)
\]
7.2.3 The Knaster-Tarski Fixpoint Theorem

**Definition 7.19** (Fixpoints ([DP02, Def. 8.14])).

1. A **fixpoint** of a function \( f : X \to X \) is an \( x \in X \) such that \( f(x) = x \).
2. Let \( L \) be a partial order and let \( f : L \to L \) be monotone. We say that \( a \in L \) is a **pre-fixpoint** of \( f \) if \( f(a) \leq a \), and that \( a \in L \) is a **post-fixpoint** of \( f \) if \( a \leq f(a) \).

A monotone function \( f : L \to L \) on a complete lattice has always a least fixpoint \( \mu(f) \in L \) and a greatest fixpoint \( \nu(f) \in L \). Intuitively, the least fixpoint \( \mu(f) \) can always be obtained as the least pre-fixpoint of \( f \). Dually, the greatest fixpoint of \( \nu(f) \) can always be obtained as the greatest post-fixpoint of \( f \).

**Theorem 7.20** (Knaster-Tarski Fixpoint Theorem ([DP02, Thm. 2.35])). Let \( L \) be a complete lattice and let \( f : L \to L \) be a monotone function. Then the **least fixpoint** \( \mu(f) \) and the **greatest fixpoint** \( \nu(f) \) are given resp. by

\[
\begin{align*}
\mu(f) &= \bigwedge \{ a \in L \mid f(a) \leq a \} \\
\nu(f) &= \bigvee \{ a \in L \mid a \leq f(a) \}
\end{align*}
\]

**Proof.** Exercise! □

**Remark 7.21** (On the Modal \( \mu \)-Calculus). The full extension of LML with fixpoints is the (linear-time) modal \( \mu \)-calculus, a powerful logic, due to [Koz83], whose study would lead us too far for this course. We refer to e.g. [VW08, §6] and [GTW02, BW18] and references therein for more material on the modal \( \mu \)-calculus. At the semantic level, we refer to [DP02, §8.27–31] for reasoning principles with least and greatest fixpoints.

We finally note the following duality between least and greatest fixpoints.

**Lemma 7.22.** Let \( \varphi \) be a formula with parameters \( \rho \) and assume that \( X \) Pos \( \varphi \). Then

\[
\nu(\llbracket \varphi \rrbracket_\rho(X)) = (2^{AP})^\omega \setminus \mu(\llbracket \psi \rrbracket_\rho(X)) \quad \text{and} \quad \mu(\llbracket \varphi \rrbracket_\rho(X)) = (2^{AP})^\omega \setminus \nu(\llbracket \psi \rrbracket_\rho(X))
\]

7.3 The Logic LTL

The logic LTL is the extension of LML with a limited form of fixpoints, which can be presented as follows. Consider a formula \( \theta(X) \) with \( X \) Pos \( \theta \). Then using the laws of Fig. 6, we can put \( \theta(X) \) in disjunctive normal form, and obtain

\[
\theta(X) \equiv \psi \lor \bigvee_{i \in I} \left( \varphi_i \land \bigwedge_{j \in J} \Box^{n_{i,j}} X \right)
\]

where \( X \) does not occur in \( \psi \) nor in the \( \varphi_i \)'s. If we further assume that in \( \theta \), \( X \) occurs **under exactly one** \( \Box \), then we have

\[
\theta(X) \equiv \psi \lor \bigvee_{i \in I} (\varphi_i \land \Box X) \\
\equiv \psi \lor (\varphi \land \Box X)
\]
where $X$ does not occur in $\psi$ nor in $\varphi$.

The logic LTL is the extension of LML with least (and greatest by the duality of Lem. 7.22) fixpoints of formulae of the form $\theta(X) = \psi \lor (\varphi \land \Box X)$. Concretely, we extend the formulae of LML with a modality $\varphi \mathrel{U} \psi$ (pronounced $\varphi$ until $\psi$), whose semantics is the least fixpoint of $\theta(X) = \psi \lor (\varphi \land \Box X)$.

Our order of presentation does not follow [BK08].

### 7.3.1 Syntax and Semantics of LTL

The formulae of LTL are given by the following grammar:

$$\varphi, \psi ::= \top | \bot | X | a \quad \text{(where } X \in \mathcal{X} \text{ and } a \in \mathcal{AP})$$

$$| \varphi \land \psi | \varphi \lor \psi | \neg \varphi$$

$$| \Diamond \varphi | \varphi \mathrel{U} \psi$$

The semantics of LTL-formulae extends that of LML with the clause:

$$\llbracket \varphi \mathrel{U} \psi \rrbracket_\rho := \{ \sigma \in (2^{\mathcal{AP}})^\omega \mid \exists i \in \mathbb{N}, \sigma[i] \in \llbracket \psi \rrbracket_\rho \text{ and } \forall j < i, \sigma[j] \in \llbracket \varphi \rrbracket_\rho \}$$

We extend the notation $\sigma \models \varphi$ of Notation 7.5. This gives, for closed $\varphi, \psi$,

$$\sigma \models \varphi \mathrel{U} \psi \iff \exists i \in \mathbb{N}, \sigma[i] \models \psi \text{ and } \forall j < i, \sigma[j] \models \varphi$$

### 7.3.2 Fixpoints and Defined Modalities

It is now time to check that $\varphi \mathrel{U} \psi$ is indeed the least fixpoint of $\theta(X) = \psi \lor (\varphi \land \Box X)$.

**Lemma 7.23** ([BK08, Lem. 5.18]). Given formulae $\varphi, \psi$ with parameters $\rho$, $\llbracket \varphi \mathrel{U} \psi \rrbracket_\rho$ is the least fixpoint of $\llbracket \theta \rrbracket_\rho(X) : \mathcal{P}(2^{\mathcal{AP}})^\omega \to \mathcal{P}(2^{\mathcal{AP}})^\omega$, where

$$\theta(X) := \psi \lor (\varphi \land \Box X)$$

**Proof.** Exercise!

With the notations of §7.2.3, Lem. 7.23 says that for $\varphi, \psi$ with parameters $\rho$ and with $\theta(X)$ as in Lem. 7.23 we have

$$\llbracket \varphi \mathrel{U} \psi \rrbracket_\rho = \mu X. \llbracket \theta \rrbracket_\rho(X)$$

Using the duality of Lem. 7.22, we can obtain a syntactic representation of the greatest fixpoint of $\theta(X)$, known as the weak until modality. It follows from Lem. 7.22 that the greatest fixpoint $\nu X. \llbracket \theta \rrbracket_\rho(X)$ of $\llbracket \theta \rrbracket_\rho(X)$ is given by

$$\nu X. \llbracket \theta \rrbracket_\rho(X) = (2^{\mathcal{AP}})^\omega \setminus \mu X. \llbracket \neg \theta[-X/X] \rrbracket_\rho(X)$$

By using the laws of Fig. 6, we have

$$\neg \theta[-X/X] \equiv \neg (\psi \lor (\varphi \land \Box \neg X))$$

$$\equiv \neg \psi \land \neg (\varphi \land \Box \neg X)$$

$$\equiv \neg \psi \land (\neg \varphi \lor \Box X)$$

$$\equiv (\neg \psi \land \neg \varphi) \lor (\neg \psi \land \Box X)$$

45
It thus follows from Lem. 7.23 that
\[ \mu X. [\neg \theta [\neg X/X]]_\rho (X) = \llbracket \neg \psi \lor (\psi \lor \varphi) \rrbracket_\rho \]
so that
\[ \nu X. [\theta]_\rho (X) = \llbracket \neg (\neg \psi \lor (\psi \lor \varphi)) \rrbracket_\rho \]

**Notation 7.24 (Weak Until).** Given formulae \( \varphi \) and \( \psi \), we let
\[ \varphi W \psi := \neg (\neg \psi \lor (\psi \lor \varphi)) \]

The above discussion leads us to the expected:

**Lemma 7.25.** Given formulae \( \varphi \) and \( \psi \) with parameters \( \rho \), \( \llbracket \varphi W \psi \rrbracket_\rho \) is the greatest fixpoint of \( \llbracket \theta \rrbracket_\rho (X) : P((2^{AP})^\omega) \to P((2^{AP})^\omega) \), where
\[ \theta(X) := \psi \lor (\varphi \land \Box X) \]

It is then direct to define the modalities \( \Diamond \) and \( \square \) discussed in §7.2.1. Recall from Lem. 7.16 that \( \Diamond \varphi \) and \( \square \varphi \) are respectively the least and greatest fixpoints of
\[ \varphi \Diamond (X) := \varphi \lor \Box X \quad \varphi \Box (X) := \varphi \land \Box X \]

**Notation 7.26 (Eventually and Always).** Given a formula \( \varphi \), we let
\[ \Diamond \varphi := \top U \varphi \quad \Box \varphi := \varphi W \bot \]

Finally, note that while we have presented LTL as the extension of LML with least (and greatest) fixpoints of formulae of the form \( \theta(X) = \psi \lor (\varphi \land \Box X) \), there are quite simple (positive and guarded) fixpoints which are not definable in LTL.

**Proposition 7.27.** Let \( a \in AP \). There is no closed LTL-formula \( \varphi \) such that \( \llbracket \varphi \rrbracket \) is the greatest fixpoint of \( \theta(X) := a \land \Box \Box X \).

Proposition 7.27 is part of a non-trivial theory. We refer to e.g. [PP04, Chap. VIII] and references therein for details.

### 7.3.3 Logical Equivalence

The notion of logical equivalence for LTL is exactly that of LML (Def. 7.7) extended to the formulae of LTL. In addition to the rules of Fig. 6, we have the equivalences for LTL-formulae of Fig. 8.

**Lemma 7.28.** All the equivalences of Fig. 6 and Fig. 8 hold.

We refer to [BK08, Fig. 5.7, p. 248] and [BK08, §5.1.5] for further equivalences. We nevertheless note the two following facts. First, there is a simple direct description of \( \varphi W \psi \).
Modal Duality Laws:

\[ \Diamond \varphi \equiv \neg \Box \neg \varphi \quad \Box \varphi \equiv \neg \Diamond \neg \varphi \]

Modal Operators Laws:

\[ \Diamond (\varphi \lor \psi) \equiv \Diamond \varphi \lor \Diamond \psi \quad \Diamond \bot \equiv \bot \]
\[ \Box (\varphi \land \psi) \equiv \Box \varphi \land \Box \psi \quad \Box \top \equiv \top \]

Distributive \( \Box / \lor \) Law:

\[ \Box (\varphi \lor \psi) \equiv \Box \varphi \lor \Box \psi \]

Expansion Laws:

\[ \varphi \lor \psi \equiv \psi \lor (\varphi \land \Box \psi) \]
\[ \Diamond \varphi \equiv \varphi \lor \Box \Box \varphi \]
\[ \Box \varphi \equiv \varphi \land \Box \Box \varphi \]

Figure 8: Some Usual LTL Laws.

Lemma 7.29 ([BK08, Lem. 5.19]). We have

\[ \varphi W \psi \equiv (\varphi U \psi) \lor \Box \varphi \]

Proof. Exercise!

Second, Lem. 7.22 gives the following dualities.

Lemma 7.30. We have

\[ \neg (\varphi W \psi) \equiv \neg \psi U (\neg \varphi \land \neg \psi) \]
\[ \neg (\varphi U \psi) \equiv \neg \psi W (\neg \varphi \land \neg \psi) \]

Proof. Exercise!

### 7.3.4 Positive Normal Forms

By extending the syntax of LTL with the weak until modality \( \varphi W \psi \), the equivalences of §7.3.3 allow us to “reduce” each formula to a formula in positive normal form, i.e. to a formula in which negations are only allowed on atomic formulae \( a \in \text{AP} \) and on variables \( X \in \mathcal{X} \). This, however, comes with an exponential blow-up if one uses the equivalences of Lem. 7.30. A solution for this is, instead of extending the syntax of LTL with \( W \), to extend it with the formal dual \( R \) of \( U \), called release and such that

\[ \varphi R \psi \equiv \neg (\neg \varphi U \neg \psi) \]
\[ \varphi U \psi \equiv \neg (\neg \varphi R \neg \psi) \]

We refer to [BK08, §5.1.5] for details.
7.3.5 Satisfaction of LTL-Formulae by Transition Systems

Consider a transition system $TS = (S, Act, \rightarrow, I, AP, L)$ over AP. A (closed) LTL-formula $\varphi$ defines a linear-time property $[[\varphi]] \subseteq (2^{AP})^\omega$. Hence, we can specialize the notion of satisfaction of LT properties (Def. 3.8) to the following.

**Definition 7.31.** Given $TS$ and $\varphi$ as above, we say that $TS$ satisfies $\varphi$ (notation $TS \models \varphi$) if $TS \models [[\varphi]]$ (i.e. if $Tr_\omega(TS) \subseteq [[\varphi]]$).

Definition 7.31 corresponds to [BK08, Def. 5.7]. We refer to [BK08, §5.1.2 & 5.1.3] for examples.

8 Toward Stone Duality

**Warning (On AP).** In this section we always assume that AP is a finite non-empty set.

In §6, we devised a topological notion of “observable properties”, which consist of the Boolean algebra of clopen sets of a topological space. For spaces $(2^{AP})^\omega$, this amounts to the Boolean algebra of sets of the form $\text{ext}(W)$ for some finite $W \subseteq (2^{AP})^\omega$. In §7, we devised LML, a base modal logic for linear-time properties, whose formulae define exactly the observable properties on $(2^{AP})^\omega$. We noted that LML is very weak w.r.t. the linear-time properties discussed in §3, and considered LTL, and extension of LML with a restricted form of least and greatest fixpoints.

In this Section, we shall discuss further topological properties of spaces $(2^{AP})^\omega$, which allow for recovering the whole set $(2^{AP})^\omega$ from the Boolean algebra of clopen sets of its topology, i.e. from LML.

We use the following notation.

**Notation 8.1.** Given a compact Hausdorff space $(X, \Omega)$, we let $K \Omega$ be the set of compact open subsets of $X$.

Recall from Lem. 6.10 and Prop. 6.20 that for $(X, \Omega)$ compact Hausdorff, $K \Omega$ coincides with the set of clopen subsets of $X$. By Prop. 6.12 and Prop. 6.15, we in particular have

$$K \Omega((2^{AP})^\omega) = \{ \text{ext}(W) \mid W \subseteq (2^{AP})^\omega \text{ is finite} \}$$

Given an $\omega$-word $\sigma \in (2^{AP})^\omega$, let

$$\mathcal{F}_\sigma := \{ \text{ext}(W) \in K \Omega((2^{AP})^\omega) \mid \sigma \in \text{ext}(W) \}$$

The following observations are easy. Recall that $\text{ext}(\varepsilon) = (2^{AP})^\omega$ and that $\text{ext}(\emptyset) = \emptyset$.

1. $(2^{AP})^\omega \in \mathcal{F}_\sigma$ and $\emptyset \notin \mathcal{F}_\sigma$.

2. If $U \in \mathcal{F}_\sigma$ and $U \subseteq V$ with $V \in K \Omega((2^{AP})^\omega)$ then $V \in \mathcal{F}_\sigma$.

3. If $U \in \mathcal{F}_\sigma$ and $V \in \mathcal{F}_\sigma$ then $U \cap V \in \mathcal{F}_\sigma$. 

48
8 Toward Stone Duality

(4) If \( U \cup V \in \mathcal{F}_\sigma \) with \( U, V \in K\Omega((2^{AP})^\omega) \), then either \( U \in \mathcal{F}_\sigma \) or \( V \in \mathcal{F}_\sigma \).

Given a Boolean algebra \( B \), subsets \( \mathcal{F} \subseteq B \) satisfying the above conditions are called prime filters on \( B \).

Note that for \( \sigma, \beta \in (2^{AP})^\omega \), we evidently have \( \mathcal{F}_\sigma \neq \mathcal{F}_\beta \) whenever \( \sigma \neq \beta \). Hence \( (2^{AP})^\omega \) can be embedded into the set of prime filters on \( K\Omega((2^{AP})^\omega) \). Actually, \( (2^{AP})^\omega \) is in bijection with the set of prime filters on \( K\Omega((2^{AP})^\omega) \). This fact, which is part of Stone’s Representation Theorem, holds for any Stone space.

**Definition 8.2 (Stone Space).** A **Stone space** is a topological space \( (X, \Omega) \) which is compact (see Def. 6.9), and satisfies the two following conditions:

\( (X, \Omega) \) is **T_0**: for any distinct points \( x, y \in X \), there is an open containing one and not the other, i.e. there is some \( U \in \Omega \) such that either \( (x \in U \text{ and } y \notin U) \) or \( (x \notin U \text{ and } y \in U) \).

\( (X, \Omega) \) is **zero-dimensional**: the clopen subsets of \( X \) form a base for the topology.

Note that every Stone space \( (X, \Omega) \) is Hausdorff (Def. 6.18).

**Example 8.3.** It follows from Lem. 6.7 that \( A^\omega \) is zero-dimensional, whether or not \( A \) is finite. Hence \( (2^{AP})^\omega \) is a Stone space by Prop. 6.12.

We shall target the two following instances of Stone’s Representation Theorem:

- Every Boolean algebra \( B \) is isomorphic to the Boolean algebra \( K\Omega(\text{Sp}(B)) \) for some Stone space \( (\text{Sp}(B), \Omega(\text{Sp}(B))) \), called the **spectrum** of \( B \).
- Every Stone space \( (X, \Omega) \) is homeomorphic to the spectrum of the Boolean algebra \( K\Omega \).

We refer to [Joh82, Cor. II.4.4] for the full statement of Stone’s Representation Theorem.

Let us finally illustrate the **logical** relevance of these matters in our context.

**Definition 8.4.**

(1) Let \( \mathcal{L}(\text{LML}) \) be the set of closed LML-formulae quotiented by logical equivalence \( \equiv \)

(in the sense of Def. 7.7 and Fig. 6, §7.1.2).

(2) Let \( \mathcal{L}(\text{LTL}) \) be the set of closed LTL-formulae quotiented by logical equivalence \( \equiv \)

(in the sense of §7.3.3).

**Notation 8.5.** In this Section 8, \( L \) stands for either LML or LTL.

We shall always notationally confuse a closed \( L \)-formula \( \varphi \) with its quotient \( [\varphi]_\equiv \in \mathcal{L}(L) \), where, as usual

\[ [\varphi]_\equiv = \{ \psi \mid \psi \text{ is a closed } L \text{-formula such that } \varphi \equiv \psi \} \]
We equip \( \mathcal{L}(L) \) with the partial order

\[
\varphi \leq \psi \coloneqq (\varphi \rightarrow \psi) \equiv \top
\]

Note that

\[
\varphi \leq \psi \quad \text{iff} \quad \varphi \equiv (\varphi \land \psi) \quad \text{iff} \quad (\varphi \lor \psi) \equiv \psi
\]

and that \( \leq \) is indeed a partial order on \( \mathcal{L}(L) \) (i.e. \( \varphi \equiv \psi \) if \( \varphi \leq \psi \) and \( \psi \leq \varphi \)).

It follows from Prop. 7.10 and Prop. 7.11 (§7.1.3) that we can identify \( K\Omega((2^{AP})^\omega) \) with \( \mathcal{L}(LML) \). The properties of prime filters underlined above can then be rephrased as follows, for a set \( \mathcal{F} \subseteq \mathcal{L}(LML) \):

- \( \mathcal{F} \) is non-empty (\( \top \in \mathcal{F} \)) and coherent (\( \bot \notin \mathcal{F} \)).
- \( \mathcal{F} \) is a theory:
  - \( \varphi \in \mathcal{F} \) and \( \varphi \leq \psi \) imply \( \psi \in \mathcal{F} \),
  - \( \varphi, \psi \in \mathcal{F} \) implies \( \varphi \land \psi \in \mathcal{F} \).
- \( \mathcal{F} \) is complete (\( \varphi \lor \psi \in \mathcal{F} \) implies either \( \varphi \in \mathcal{F} \) or \( \psi \in \mathcal{F} \), so that for every \( \varphi \) we have either \( \varphi \in \mathcal{F} \) or \( \neg \varphi \in \mathcal{F} \)).

Then, the existence of a bijection between \( (2^{AP})^\omega \) and the set of prime filters over \( K\Omega((2^{AP})^\omega) \) can be read as a completeness theorem:

- Every complete consistent theory \( \mathcal{F} \subseteq \mathcal{L}(LML) \) has a model, i.e. there is some \( \sigma \in (2^{AP})^\omega \) such that for all \( \varphi \in \mathcal{L}(LML) \), we have \( \sigma \models \varphi \) iff \( \varphi \in \mathcal{F} \).

**Remark 8.6 (On Lindenbaum-Tarski Algebras).** The set \( \mathcal{L}(L) \) defined in Def. 8.4 is reminiscent from **Lindenbaum-Tarski algebras** (see e.g. [BdRV02, Def. 5.31]). However, Lindenbaum-Tarski algebras are usually defined as the quotient of formulae w.r.t. provable logical equivalence (see also Rem. 7.9).

### 9 Bisimulation

We essentially follow here [BK08, §7.1], with a few slight changes in notation.

#### 9.1 Bisimulation (with Actions)

We begin with the usual notion.

**Definition 9.1** (Bisimulation). Consider t.s. \( TS_0 \) and \( TS_1 \) with \( TS_i = (S_i, \text{Act}, \rightarrow_i, I_i, \text{AP}, L_i) \). A **bisimulation** between \( TS_0 \) and \( TS_1 \) is a relation \( \mathcal{R} \subseteq S_0 \times S_1 \) such that for all \( (s_0, s_1) \in \mathcal{R} \) we have

- (i) \( L_0(s_0) = L_1(s_1) \), and
- (ii) for each \( i \in \{0, 1\} \) and each \( \alpha \in \text{Act} \), if \( s_i \xrightarrow{\alpha_i} s'_i \) in \( TS_i \) then there is \( s'_{i-1} \) in \( TS_{1-i} \) such that \( s_{i-1} \xrightarrow{\alpha_{1-i}} s'_{i-1} \) and \( (s_0', s_1') \in \mathcal{R} \).
Note that in Def. 9.1, $TS_0$ and $TS_1$ are required to be over the same sets $Act$ and $AP$ of actions and atomic propositions.

**Definition 9.2.** We write $TS_0 \approx TS_1$ if there is a bisimulation $R$ between $TS_0$ and $TS_1$ such that moreover

- for each $i \in \{0, 1\}$, for all $s_i \in I_i$ there is $s_{1-i} \in I_{1-i}$ such that $(s_0, s_1) \in R$.

Definition 9.2 corresponds to [BK08, Def. 7.1] (but with a slightly different notation).

**Definition 9.3** (Bisimilarity). Consider transition systems $TS_0$ and $TS_1$ as in Def. 9.1. We say that $s_0 \in S_0$ and $s_1 \in S_1$ are bisimilar (notation $s_0 \sim s_1$) if there is a bisimulation $R \subseteq S_0 \times S_1$ such that $(s_0, s_1) \in R$. The relation $\sim$ is called the **bisimilarity relation** over $TS_0$ and $TS_1$.

We now turn to the basic properties of bisimulations.

**Lemma 9.4.**

1. Given a transition system $TS$, we have $s \sim s$ for each state $s$ of $TS$.

2. Given transition systems $TS_0$ and $TS_1$, if $R$ is a bisimulation between $TS_0$ and $TS_1$, then $R^{-1} = \{(s_1, s_0) \mid (s_0, s_1) \in R\}$ is a bisimulation between $TS_1$ and $TS_0$.

3. Given transition systems $TS_0$, $TS_1$ and $TS_2$, if $R$ is a bisimulation between $TS_0$ and $TS_1$, and $T$ is a bisimulation between $TS_1$ and $TS_2$, then $T \circ R$ is a bisimulation between $TS_0$ and $TS_2$, where

$$T \circ R = \{(s_0, s_2) \mid \exists s_1, (s_0, s_1) \in R \text{ and } (s_1, s_2) \in T\}$$

**Proof.** Exercise!

**Lemma 9.5.**

1. Given $TS_0$ and $TS_1$, the bisimilarity relation $\sim$ over $TS_0$ and $TS_1$ is a bisimulation between $TS_0$ and $TS_1$.

2. Given $TS_0$ and $TS_1$, the bisimilarity relation $\sim$ is the coarsest bisimulation between $TS_0$ and $TS_1$ (i.e. given any bisimulation $R$ between $TS_0$ and $TS_1$, we have $R \subseteq \sim$).

3. For every $TS$, the bisimilarity relation $\sim$ over $TS$ and itself is an equivalence relation.

**Proof.** Exercise!
9.2 Bisimilarity and Trace Equivalence

It follows from the definition that we have.

**Proposition 9.6** ([BK08, Thm. 7.6]). Given $TS_0$ and $TS_1$ over both over $AP$ and $Act$, if $TS_0 \approx TS_1$ then $\text{Tr}^\omega(TS_0) = \text{Tr}^\omega(TS_1)$.

**Corollary 9.7.** Given $TS_0$ and $TS_1$ over both over $AP$ and $Act$, if $TS_0 \approx TS_1$ then for all LT property $P \subseteq (2^{AP})^\omega$, we have

$$TS_0 \models P \text{ if and only if } TS_1 \models P$$

In particular, if $TS_0 \approx TS_1$, then for every LTL-formula $\varphi$ we have

$$TS_0 \models \varphi \text{ if and only if } TS_1 \models \varphi$$

9.3 The Bisimulation Quotient

Given a transition system $TS$, let $TS_\sim$ be the transition system with

- as states the equivalence classes $[s]_\sim$ of $\sim$,
- as initial states the equivalence classes of initial states of $TS$,
- as transitions, we let $[s]_\sim \stackrel{\alpha}{\rightarrow} [s']_\sim$ if $s \stackrel{\alpha}{\rightarrow} s'$,
- as labeling, note that if $s \sim t$ then $L(s) = L(t)$, so that we can put $L([s]_\sim) := L(s)$.

**Lemma 9.8.** $TS \approx TS_\sim$.

**Proof.** Exercise! □

10 On Modal Logics of Transition Systems

In this Section, we study a modal logic on transition systems (in the sense of §2 and [BK08, Def. 2.1]) which properly deals with their transition structure. We consider here the Hennessy-Milner Logic (HML). We loosely follow [Sti11] for the presentation of HML and [BdRV02] for the general theory of modal logic.

Kripke Frames and Models

The tradition of modal logic (in the sense of e.g. [BdRV02, Chap. 1]) leads us to distinguish the following structure in a transition system $TS = (S, Act, \rightarrow, I, AP, L)$:

- a transition structure given by $(S, (\stackrel{\alpha}{\rightarrow})_{\alpha \in Act})$,
- a logical (model) structure given by the state labelling $L : S \rightarrow P(AP)$,
- a “pointed” structure given by the initial states $I \subseteq S$. 

52
We use the following adaptation of the notions of [BdRV02, Chap. 1].

**Definition 10.1 (Kripke Frame and Model).** Fix Act and AP.

- A **Kripke frame** over Act is given by a set of states $S$ together with a relation $\rightarrow \subseteq S \times \text{Act} \times S$.

- A **Kripke model** over Act and AP is given by a Kripke frame $(S, \text{Act}, \rightarrow)$ together with a state labelling $L : S \rightarrow \mathcal{P}(\text{AP})$.

A t.s. is thus a Kripke model $(S, \text{Act}, \rightarrow, \text{AP}, L)$ equipped with a set of initial states $I \subseteq S$.

### 10.1 Syntax and Semantics of HML

Fix a set AP of **atomic propositions** and a set Act of **actions**. The formulae of HML are given by the following grammar:

$$
\varphi, \psi ::= \top \mid \bot \mid a \quad \text{(where} \ a \ \in \ \text{AP)} \\
\mid \varphi \land \psi \mid \varphi \lor \psi \mid \neg \varphi \\
\mid [\alpha]\varphi \mid \langle \alpha \rangle \varphi \quad \text{(where} \ \alpha \ \in \ \text{Act)}
$$

**Notation 10.2.** Other propositional connectives are defined as usual (see also Notation 7.2, §7.1.1):

$$
\varphi \rightarrow \psi := \neg \varphi \lor \psi \\
\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi)
$$

Consider a Kripke model $M = (S, \text{Act}, \rightarrow, \text{AP}, L)$. The interpretation $[\varphi] \in \mathcal{P}(S)$ of an HML-formula $\varphi$ is defined by induction on $\varphi$ as follows:

$$
[[a]] := \{s \in S \mid a \in L(s)\} \\
[[\top]] := S \\
[[\bot]] := \emptyset \\
[[\varphi \land \psi]] := [[\varphi]] \cap [[\psi]] \\
[[\varphi \lor \psi]] := [[\varphi]] \cup [[\psi]] \\
[[\neg \varphi]] := S \setminus [[\varphi]] \\
[[[\alpha]\varphi]] := \{s \in S \mid \forall s' \in S, \text{ if } s \xrightarrow{\alpha} s' \text{ then } s' \in [[\varphi]]\} \\
[[\langle \alpha \rangle \varphi]] := \{s \in S \mid \exists s' \in S, s \xrightarrow{\alpha} s' \text{ and } s' \in [[\varphi]]\}
$$

The following usual notions are presented e.g. in [BdRV02, §1.3] (with slight variations in notation).

**Definition 10.3 (Modal Satisfaction).** Consider a Kripke model $M = (S, \text{Act}, \rightarrow, \text{AP}, L)$ and an HML-formula $\varphi$.

(1) We say that a state $s \in S$ **satisfies** $\varphi$ (notation $s \models \varphi$) if $s \in [[\varphi]]$ ([BdRV02, Def. 1.20]).
(2) We say that \( M \) satisfies \( \varphi \) (notation \( M \models \varphi \)) if \( s \in \llbracket \varphi \rrbracket \) for all \( s \in S \) ([BdRV02, Def. 1.21]).

We say that \( \varphi \) is valid (notation \( \models \varphi \)) is \( M \models \varphi \) for every Kripke model \( M \) (over \( \text{Act} \) and \( \text{AP} \)).

Remark 10.4. We shall be mostly concerned with the local satisfaction relation \( \models \). The notion of satisfaction in a given Kripke model \( M \models \varphi \) only interests us as a means to define logical validity \( \models \varphi \). As a consequence, we shall not bother in seriously considering the possible notion for transition systems

\[ TS \models \top \text{ iff } \forall s \in I, s \models \varphi \]

since it would induce the same notion of logical validity (by changing initial states of t.s.’s).

Remark 10.5. Similarly as with LML (Rem. 7.6, §7.1.1) and LTL (§7.3.1), we can give an inductive characterization of the relation \( s \models \varphi \) (s forces \( \varphi \)):

\[
\begin{align*}
s \models a & \quad \text{iff } a \in L(s) \\
s \models \top & \\

s \not\models \bot & \\

s \models \varphi \land \psi & \quad \text{iff } s \models \varphi \text{ and } s \models \psi \\

s \models \varphi \lor \psi & \quad \text{iff } s \models \varphi \text{ or } s \models \psi \\

s \models \neg \varphi & \quad \text{iff } s \not\models \varphi \\

s \models [\alpha] \varphi & \quad \text{iff } \text{ for all } s' \in S \text{ such that } s \xrightarrow{\alpha} s', \text{ we have } s' \models \varphi \\

s \models (\alpha) \varphi & \quad \text{iff } \text{ there is some } s' \in S \text{ such that } s \xrightarrow{\alpha} s' \text{ and } s' \models \varphi
\end{align*}
\]

Remark 10.6. One gets the usual basic modal logic by taking \( \text{Act} = 1 \) (see e.g. [BdRV02, Def. 1.9]).

Example 10.7 (LML as an instance of HML). Fix \( \text{Act} = \{\bullet\} \). We define the following Kripke model \( M((2^{\text{AP}})^\omega) \) on streams:

\[
M((2^{\text{AP}})^\omega) := ( (2^{\text{AP}})^\omega, \text{Act}, \rightarrow, \text{AP}, L )
\]

where

\[
\begin{align*}
\sigma \xrightarrow{\bullet} \beta & \quad \text{iff } \beta = \sigma|1 \\
\sigma \models a & \quad \text{iff } a \in L(\sigma)
\end{align*}
\]

Then for all HML-formula \( \varphi \) and each \( \sigma \in (2^{\text{AP}})^\omega \), we have

\[
\begin{align*}
\sigma \models (\bullet) \varphi & \quad \text{iff } \sigma|1 \models \varphi \\
\sigma \models [\bullet] \varphi & \quad \text{iff } \sigma \models \varphi
\end{align*}
\]

Hence both modalities \( (\bullet) \) and \( [\bullet] \) collapse to the \( \circ \) modality of LML. It is then easy to see that HML and LML have the same expressive power over \( M((2^{\text{AP}})^\omega) \).

Moreover, two streams \( \sigma, \beta \in (2^{\text{AP}})^\omega \) are bisimilar iff they are equal.

Proof. Exercise!
10.2 Logical Equivalence

We shall consider two notions of logical equivalence with HML, first the logical equivalence of formulae, similar to that of LML and LTL seen in §7, and second the logical equivalence of states of Kripke models.

10.2.1 Logical Equivalence on Formulae

Similarly as with LML and LTL, HML has a notion of logical equivalence on formulae.

**Definition 10.8 (Logical Equivalence on Formulae).** Two HML-formulae $\varphi$ and $\psi$ are **logically equivalent** (notation $\varphi \equiv \psi$), if $M \models \varphi \leftrightarrow \psi$.

Hence $\varphi \equiv \psi$ iff $M \models \varphi \leftrightarrow \psi$ for every Kripke model $M$. But this is equivalent to $[\varphi] = [\psi]$ within every Kripke model $M$.

**Lemma 10.9.** We have $\varphi \equiv \psi$ iff for every Kripke model $M = (S, \text{Act}, \rightarrow, \text{AP}, L)$ and all $s \in S$,

$$s \models \varphi \iff s \models \psi$$

**Proof.** Exercise!

**Lemma 10.10.** We have, for $\alpha \in \text{Act}$,

$$\langle \alpha \rangle \varphi \equiv \neg[\alpha] \neg \varphi$$

$$[\alpha] \varphi \equiv \neg (\alpha) \neg \varphi$$

as well as

$$\langle \alpha \rangle (\varphi \lor \psi) \equiv \langle \alpha \rangle \varphi \lor \langle \alpha \rangle \psi$$

$$\langle \alpha \rangle (\varphi \land \psi) \equiv [\alpha] \varphi \land [\alpha] \psi$$

$$\langle \alpha \rangle \bot \equiv \bot$$

$$[\alpha] \top \equiv \top$$

10.2.2 Logical Equivalence on States and Bisimilarity

In HML, it is pertinent to consider a notion of logical equivalence on states of Kripke models.

**Definition 10.11.** Consider $M_0$ and $M_1$ with $M_i = (S_i, \text{Act}, \rightarrow_i, \text{AP}, L_i)$. We say that $s_0 \in S_0$ and $s_1 \in S_1$ are **logically equivalent** (notation $s_0 \equiv s_1$) if for all HML-formula $\varphi$ we have

$$s_0 \models \varphi \iff s_1 \models \varphi$$

It is expected that for a modal logic, bisimilarity of states implies logical equivalence.

**Theorem 10.12 ([BdRV02, Thm. 2.20 p. 67]).** If $s_0 \sim s_1$, then $s_0 \equiv s_1$.

**Proof.** Exercise!
10.3 The Hennessy-Milner Property

We shall look for (partial) converses to Thm. 10.12, i.e. for sufficient conditions on a class $\mathcal{M}$ of Kripke models (over fixed Act and AP) such that given $M_0, M_1 \in \mathcal{M}$ and $s_0 \in S_0, s_1 \in S_1$, we have

$$s_0 \sim s_1 \text{ iff } s_0 \equiv s_1$$

This property is called the Hennessy-Milner property for $\mathcal{M}$ (see e.g. [BdRV02, Def. 2.52 p. 92] or [Sti11]). As shown by Ex. 10.13 below, the class $\mathcal{R}$ of all Kripke models (over fixed Act and AP) does not have the Hennessy-Milner property.

**Example 10.13 ([BdRV02, Ex. 2.23 & Fig. 2.5]).** Assume Act = 1 and AP = {$a$}.

Consider the Kripke models

![Diagram](image)

(where all states have label $a$). Then we have $s_0 \equiv s_1$ but $s_0 \not\sim s_1$.

**Remark 10.14.** Showing that $s_0 \equiv s_1$ in Ex. 10.13 can be done directly (see [BdRV02, Ex. 2.23 & Fig. 2.5]). But it is convenient for such tasks to use appropriate tools (e.g. [BdRV02, Prop. 2.31 & Lem. 2.33]) providing suitable induction principles on formulae (actually quite similar to those for Prop. 7.27, namely variants of the Ehrenfeucht-Faissé method, see e.g. [PP04, Chap. VIII]).

Note that the Hennessy-Milner property for a class $\mathcal{M}$ of Kripke models is equivalent to the following condition:

- Given $M_0, M_1 \in \mathcal{M}$, the logical equivalence relation on states $\equiv \subseteq S_0 \times S_1$ is a bisimulation.

It is well-known that the class of image finite Kripke models has the Hennessy-Milner property. This is result is known as the Hennessy-Milner Theorem (see e.g. [BdRV02, Thm. 2.24, p. 69] or [Sti11, Thm. 1.2.3 & Thm. 1.2.4]).

**Definition 10.15 (Image Finite T.S.).** We say that $M$ is image finite if for every $s \in S$ and $\alpha \in \text{Act}$, the set

$$\text{Succ}^\alpha(s) := \{s' \in S \mid s \xrightarrow{\alpha} s'\}$$

of $\alpha$-successors of $s$ is finite.
Proposition 10.16 (Hennessy-Milner Theorem). If $M_0$ and $M_1$ (both over $\text{AP}$ and $\text{Act}$) are \textbf{image finite}, then for all $(s_0, s_1) \in S_0 \times S_1$ we have

$$s_0 \sim s_1 \text{ iff } s_0 \equiv s_1$$

We refer to e.g. [Sti11, Thm. 1.2.4] for a direct proof of Prop. 10.16. In §10.4 we prove Prop. 10.16 using the model-theoretic notion of \textbf{modal saturation}. This notion paves the way toward the main construction and result of this §10, namely that for each Kripke model $M$ there is a (modally saturated) Kripke model $\text{iff}(M)$ (called the \textbf{ultrafilter extension} of $M$) and a function $\pi : S_M \to S_{\text{iff}(M)}$ such that

$$s_0 \equiv s_1 \text{ iff } \pi(s_0) \sim \pi(s_1)$$

10.4 Modal Saturation

We follow [BdRV02, §2.5 p. 91].

Definition 10.17. Let $M = (S, \text{Act}, \to, \text{AP}, L)$.

(1) Given a set of states $T \subseteq S$, a set of formulae $\Phi$ is \textbf{satisfiable} in $T$ if there is a state $s \in T$ such that $s \models \varphi$ for all $\varphi \in \Phi$.

(2) Given a set of states $T \subseteq S$, a set of formulae $\Phi$ is \textbf{finitely satisfiable} in $T$ if every finite subset of $\Phi$ is satisfiable in $T$.

(3) $M$ is \textbf{modally saturated} if for every state $s$, every $\alpha \in \text{Act}$, and every set of formulae $\Phi$, if $\Phi$ is finitely satisfiable in the set of $\alpha$-successors of $s$, then $\Phi$ is satisfiable in the set of $\alpha$-successors of $s$.

Proposition 10.18. If $M$ is image finite, then $M$ is modally saturated.

\textbf{Proof.} Exercise! \hfill \Box

The following Prop. 10.19 and Cor. 10.20 are gathered in [BdRV02, Prop. 2.54, p. 93].

Proposition 10.19. If $M_0$ and $M_1$ are modally saturated, then $\equiv$ is a bisimulation between $M_0$ and $M_1$.

\textbf{Proof.} Exercise! \hfill \Box

Corollary 10.20. If $M_0$ and $M_1$ are modally saturated, then for every $(s_0, s_1) \in S_0 \times S_1$, we have

$$s_0 \sim s_1 \text{ iff } s_0 \equiv s_1$$

Proposition 10.16 is a direct consequence of Prop. 10.18 and Cor. 10.20.
10.5 Boolean Algebras with Operators

Similarly as in Def. 8.4 (§8) we write $\mathcal{L}(\text{HML})$ for the set of HML-formulae (over fixed $\text{Act}$ and $\text{AP}$) quotiented by the logical equivalence relation $\equiv$ of Def. 10.8 (§10.1). Then writing $\varphi$ for $[\varphi]_\equiv \in \mathcal{L}(\text{HML})$ (as in Notation 8.5), the relation

$$\varphi \leq \psi := (\varphi \rightarrow \psi) \equiv \top$$

(see §8) is a partial order on $\mathcal{L}(\text{HML})$, and moreover $(\mathcal{L}(\text{HML}), \leq)$ is a Boolean algebra. Similarly as in §8, for a Kripke model $M = (S, \text{Act}, \rightarrow, \text{AP}, L)$ the map

$$\lbrack - \rbrack : \mathcal{L}(\text{HML}) \rightarrow P(S)$$

$$\varphi \mapsto \lbrack \varphi \rbrack$$

is a morphism of Boolean algebras.

We shall now see an algebraic approach to HML via the notion of Boolean Algebra with Operators (BAO). While this fits quite well in the general setting of Stone Duality (see e.g. [BdRV02, Chap. 5]), we follow here a more naive approach.

**Definition 10.21.** Given a Kripke frame $K = (S, \text{Act}, \rightarrow)$ and $\alpha \in \text{Act}$, define

$$\lbrack \langle \alpha \rangle \rbrack : P(S) \rightarrow P(S)$$

$$A \mapsto \{s \in S \mid \exists s' \in \text{Suc}^\alpha(s), \ s' \in A\}$$

$$\lbrack [\alpha] \rbrack : P(S) \rightarrow P(S)$$

$$A \mapsto \{s \in S \mid \forall s' \in \text{Suc}^\alpha(s), \ s' \in A\}$$

In the case of a Kripke model $M$, we of course have

$$\lbrack \langle \alpha \rangle \varphi \rbrack = \lbrack \langle \alpha \rangle \lbrack \varphi \rbrack \rbrack$$

$$\lbrack [\alpha] \varphi \rbrack = \lbrack [\alpha] \lbrack \varphi \rbrack \rbrack$$

Moreover:

**Lemma 10.22.** Consider a Kripke frame $K = (S, \text{Act}, \rightarrow)$ and let $\alpha \in \text{Act}$.

1. The function $\lbrack \langle \alpha \rangle \rbrack : P(S) \rightarrow P(S)$ is a map of join semilattices.
2. The function $\lbrack [\alpha] \rbrack : P(S) \rightarrow P(S)$ is a map of meet semilattices.

**Proof.** Exercise! □

Lemma 10.10 gives a similar situation for $\mathcal{L}(\text{HML})$.

**Lemma 10.23.** Fix some $\alpha \in \text{Act}$.

1. The function

$$\langle \alpha \rangle : \mathcal{L}(\text{HML}) \rightarrow \mathcal{L}(\text{HML})$$

$$\varphi \mapsto \langle \alpha \rangle \varphi$$

is a map of join semilattices.
The function
\[
[\alpha] : \mathcal{L}(\text{HML}) \rightarrow \mathcal{L}(\text{HML})
\]
\[
\varphi \mapsto [\alpha] \varphi
\]
is a map of meet semilattices.

**Proof.** Exercise!

Of course, the maps \([\langle \alpha \rangle]\) and \([[\alpha]]\) (as well as \(\langle \alpha \rangle\) and \([\alpha]\) over \(\mathcal{L}(\text{HML})\)) are interdefinable. Let us elaborate a bit on this.

**Definition 10.24.** Given Boolean algebras \(B, B'\) and a function \(f : B \rightarrow B'\), the **dual** of \(f\) is the function
\[
f^\partial : B \rightarrow B'
\]
\[
b \mapsto \neg'f(\neg b)
\]

**Lemma 10.25.** Consider a Kripke frame \(K = (S, \text{Act}, \rightarrow)\) and let \(\alpha \in \text{Act}\). Then
\[
[[\alpha]] = [[\langle \alpha \rangle]]^\partial
\]
\[
\langle [\alpha] \rangle = [\langle \alpha \rangle]^\partial
\]

**Proof.** Exercise!

**Lemma 10.26.** Given \(\alpha \in \text{Act}\), in \((\mathcal{L}(\text{HML}), \leq)\) we have
\[
[\alpha] = \langle \alpha \rangle^\partial
\]
\[
\langle [\alpha] \rangle = [\alpha]^\partial
\]

**Proof.** Exercise!

**Lemma 10.27.** Let \(B, B'\) be Boolean algebras, and consider a function \(f : B \rightarrow B'\).

1. We have \(f^{\partial^\partial} = f\).

2. If \(f\) is a map of join (resp. meet) semilattices, then \(f^\partial\) is a map of meet (resp. join) semilattices.

3. If \(f\) is a map of lattices, then \(f^\partial = f\).

**Proof.** Exercise!

There are two equivalent presentations of Boolean Algebra with Operators (BAO) in the literature. The first one consists of a Boolean algebra \(B\) together with maps of join semilattices \(B \rightarrow B\). The second one consists of a Boolean algebra \(B\) together with maps of meet semilattices \(B \rightarrow B\). These two notions are equivalent by Lem. 10.27. We choose the first option as it is the one adopted in [BdRV02]. In the context of HML, this leads to the following notion.

**Definition 10.28** (Boolean Algebra with Operators). A **Boolean algebra with operators** (BAO) \(B^+\) of type \(\text{Act}\) is a Boolean algebra \(B\) equipped with a family \((f_\alpha)_{\alpha \in \text{Act}}\) of join semilattice morphisms \(f_\alpha : B \rightarrow B\).
Example 10.29. \( \mathcal{L}(\text{HML})^+ := (\mathcal{L}(\text{HML}), (\langle \alpha \rangle)_{\alpha \in \text{Act}}) \) is a BAO of type \( \text{Act} \).

Example 10.30. Given a Kripke frame \( K = (S, \text{Act}, \rightarrow) \), \( K^+ := (\mathcal{P}(S), (\langle \alpha \rangle)_{\alpha \in \text{Act}}) \) is a BAO of type \( \text{Act} \).

The crux of the algebraic approach to modal logic is that one can go the other way around. The following is the adaptation of [BdRV02, Def. 5.40, §5.3] to HML.

Definition 10.31 (Ultrafilter Frames). Given a BAO \( B^+ = (B, (f_\alpha)_{\alpha \in \text{Act}}) \), the ultrafilter frame \( \mathfrak{U}(B) \) is defined as

\[
\mathfrak{U}(B^+) := (\mathcal{Sp}(B), \text{Act}, \rightarrow)
\]

where:

- \( \mathcal{Sp}(B) \) is the set of ultrafilters (or equivalently prime filters) over \( B \) (see §??),
- given \( F, H \in \mathcal{Sp}(B) \) and \( \alpha \in \text{Act} \), we have

\[
F \overset{\alpha}{\rightarrow} H \iff \forall b \in B, b \in H \implies f_\alpha(b) \in F
\]

Lemma 10.32. Consider a BAO \( B^+ = (B, (f_\alpha)_{\alpha \in \text{Act}}) \). In the ultrafilter frame \( \mathfrak{U}(B^+) \), given \( \alpha \in \text{Act} \) and \( F, H \in \mathcal{Sp}(B) \) we have

\[
F \overset{\alpha}{\rightarrow} H \iff \forall b \in B, f_\alpha^0(b) \in F \implies b \in H
\]

Proof. Exercise!

We refer to e.g. [BdRV02] (and in particular to [BdRV02, Chap. 5]) for uses of this construction (and in particular in the context of Stone duality). We shall just see in §10.6 how this construction, applied to the BAO \( (\mathcal{P}(S), (\langle \alpha \rangle)_{\alpha \in \text{Act}}) \) of a Kripke model \( M = (S, \text{Act}, \rightarrow, \text{AP}, L) \), induces a Kripke model with the Hennessy-Milner property.

10.6 Ultrafilter Extensions of Kripke Models

The ultrafilter frame construction of Def. 10.31 turns a BAO into a frame. If one starts from the BAO \( K^+ \) induced by the frame structure \( K \) of a Kripke model \( M \), we can extend \( \mathfrak{U}(K^+) \) to a Kripke model \( \mathfrak{U}(M) \), the ultrafilter extension of \( M \), which is modally saturated (and in particular satisfies the Hennessy-Milner property). We essentially follow [BdRV02, §2.5].

We take the material of §?? for granted. We begin by specializing it to ultrafilters over powerset algebras.

Definition 10.33. Let \( X \) be a set.

1. A (proper) filter on \( X \) is a (proper) filter on \( (\mathcal{P}(X), \subseteq) \).

2. An ultrafilter on \( X \) is an ultrafilter (or equivalently a prime filter) on \( (\mathcal{P}(X), \subseteq) \).

We write \( \mathfrak{U}(X) \) for the set of ultrafilters on \( X \).
10 On Modal Logics of Transition Systems

Hence $\mathcal{U}(X) = \text{Sp}(\mathcal{P}(X), \subseteq)$.

Lemma 10.34. Let $X$ be a set. If $G \subseteq \mathcal{P}(X)$ has the finite intersection property, then

$$F := \bigcap \{E \mid E \text{ is a proper filter } \supseteq G\}$$

is a proper filter.

Note that if there is some $G \subseteq \mathcal{P}(X)$ with the finite intersection property, we necessarily have $X$ non empty, since otherwise the intersection of the empty family, which is the top element of $(\mathcal{P}(X), \subseteq)$ (i.e. $X$), would be empty.

Proof. Exercise!

Example 10.35.

(1) For each $x \in X$, the principal ultrafilter on $x$ is the ultrafilter

$$\pi(x) := \{A \in \mathcal{P}(X) \mid x \in A\}$$

(2) If $X$ is a finite set, then the ultrafilters on $X$ are exactly the principal filters on $X$. In particular, $\mathcal{U}(X)$ is in bijection with $X$.

Proof. Exercise!

(3) It follows from the Ultrafilter Lemma ?? that every family $G \subseteq \mathcal{P}(X)$ with the finite intersection property is contained in an ultrafilter.

(4) This in particular gives ultrafilters of co-finite sets (for $X$ infinite), namely ultrafilters $\mathcal{F}$ containing all $A \subseteq X$ such that $X \setminus A$ is finite.

We shall now use the ultrafilter extension of a Kripke model $M$ in order to produce modally saturated models. In the following, we assume that the labelings $L : S \to \mathcal{P}(\text{AP})$ are described by their transpose $V : \text{AP} \to \mathcal{P}(S)$ (where $s \in V(a)$ iff $a \in L(s)$).

Definition 10.36 (Ultrafilter Extension of a Kripke Model). Consider a Kripke model $M = (S, \text{Act}, \to, \text{AP}, L)$. The ultrafilter extension of $M$ is the Kripke model $\mathcal{U}(M)$ over $\text{AP}$ and $\text{Act}$ with

- as state set the set $\mathcal{U}(S)$ of ultrafilters on $S$,
- as transition relation, $F \xrightarrow{a} H$ iff $[\langle a \rangle](A) \in \mathcal{F}$ whenever $A \in H$,
- as state labelling, the map taking $a \in \text{AP}$ to the set of ultrafilters $\mathcal{F}$ such that $V(a) \in \mathcal{F}$,

In the case of as t.s. $TS = (S, \text{Act}, \to, I, \text{AP}, L)$, $\mathcal{U}(TS)$ has underlying Kripke model $\mathcal{U}(S, \text{Act}, \to, \text{AP}, L)$ and initial states $\{\pi(s) \mid s \in I\}$.
Hence the Kripke frame part of $\mathcal{Uf}(M)$ is the ultrafilter frame $\mathcal{Uf}(S, \text{Act}, \rightarrow)$ in the sense of Def. 10.31. In particular, Lem. 10.32 specializes to the following.

**Lemma 10.37.** Consider a Kripke model $M = (S, \text{Act}, \rightarrow, \text{AP}, L)$. Then, in $\mathcal{Uf}(M)$ we have

$$F \models_{\alpha} H \iff \forall A \in \mathcal{P}(S), \llbracket [\alpha] \rrbracket (A) \in F \implies A \in H$$

**Proof.** Exercise!

**Example 10.38.** Consider a Kripke model $M = (S, \text{Act}, \rightarrow, \text{AP}, L)$ with finite set of states $S$. It follows from Ex. 10.35.(2) that $\mathcal{Uf}(M)$ has a finite set of states $\mathcal{Uf}(S) \simeq S$ (via $\pi$). Moreover:

- Given $s \in S$ and $a \in \text{AP}$, we have $a \in L(s)$ in $M$ if and only if $a \in L(\pi(s))$ in $\mathcal{Uf}(M)$.

**Proof.** Exercise!

- Given $s, s' \in S$ and $\alpha \in \text{Act}$, we have $s \stackrel{\alpha}{\rightarrow} s'$ in $M$ if and only if $\pi(s) \stackrel{\alpha}{\rightarrow} \pi(s')$ in $\mathcal{Uf}(M)$.

**Proof.** Exercise!

**Notation 10.39.** In the following, given a transition system $M$ and its ultrafilter extension $\mathcal{Uf}(M)$, with $\llbracket - \rrbracket$ we always refer to the semantics of HML in $M$ rather than in $\mathcal{Uf}(M)$.

Recall the map $\text{ext}$ of Def. ?? (§??). In the case of a Boolean algebra of the form $(\mathcal{P}(X), \subseteq)$ for some set $X$, we have

$$\text{ext} : \mathcal{P}(X) \longrightarrow \mathcal{P}(\mathcal{Uf}(X))$$

$$A \longmapsto \{ F \in \mathcal{Uf}(X) \mid A \in F \}$$

In particular, given a Kripke model $M$ with state set $S$, $\text{ext}(\llbracket \varphi \rrbracket) \in \mathcal{P}(\mathcal{Uf}(S))$ for each HML-formula $\varphi$.

**Proposition 10.40.** Let $M = (S, \text{Act}, \rightarrow, \text{AP}, L)$. Then, for all $F \in \mathcal{Uf}(S)$ and all HML-formula $\varphi$, we have

$$F \models \varphi \iff F \in \text{ext}(\llbracket \varphi \rrbracket)$$

**Proof.** Exercise!

**Corollary 10.41.** Let $M = (S, \text{Act}, \rightarrow, \text{AP}, L)$. Then, for every HML-formula $\varphi$ we have

$$(\forall s \in S)(s \models \varphi \iff \pi(s) \models \varphi)$$

$$M \models \varphi \iff \mathcal{Uf}(M) \models \varphi$$
Remark 10.42. Since the initial states of \( \mathcal{Uf}(TS) \) are exactly the \( \pi(s) \) for \( s \) initial in \( TS \), Cor. 10.41 extends to the notion \( \models^\dagger \) of Rem. 10.4 as

\[
TS \models^\dagger \varphi \iff \mathcal{Uf}(TS) \models^\dagger \varphi
\]

Proposition 10.43. Let \( M = (S, \text{Act}, \rightarrow, \text{AP}, L) \). Then \( \mathcal{Uf}(M) \) is modally saturated.

Proof. Exercise! 

Corollary 10.44. Given Kripke models \( M_0 \) and \( M_1 \), both over \( \text{AP} \) and \( \text{Act} \), for all \( (s_0, s_1) \in S_0 \times S_1 \) we have

\[
s_0 \equiv s_1 \iff \pi(s_0) \sim \pi(s_1)
\]
References


References


