Class-based Classical Propositional Logic

Matthew Doty

June 17, 2024

Abstract

We formulate classical propositional logic as an axiom class. Our class represents a Hilbert-style proof system with the axioms $\vdash \varphi \to \psi \to \varphi$, $\vdash (\varphi \to \psi \to \chi) \to (\varphi \to \psi) \to \varphi \to \chi$, and $\vdash ((\varphi \to \bot) \to \bot) \to \varphi$ along with the rule *modus ponens* $\vdash \varphi \to \psi \Longrightarrow \vdash \varphi \Longrightarrow \vdash \psi$. In this axiom class we provide lemmas to obtain *Maximally Consistent Sets* via Zorn's lemma. We define the concrete classical propositional calculus inductively and show it instantiates our axiom class. We formulate the usual semantics for the propositional calculus and show strong soundness and completeness. We provide conventional definitions of the other logical connectives and prove various common identities. Finally, we show that the propositional calculus *embeds* into any logic in our axiom class.

Contents

Chapter 1

Implication Logic

theory *Implication-Logic*
  imports *Main*
begin

This theory presents the pure implicational fragment of intuitionistic logic. That is to say, this is the fragment of intuitionistic logic containing *implication only*, and no other connectives nor *falsum* (i.e., $\bot$). We shall refer to this logic as *implication logic* in future discussion.

For further reference see [7].

1.1 Axiomatization

Implication logic can be given by the a Hilbert-style axiom system, following Troelstra and Schwichtenberg [6, §1.3.9, pg. 33].

class *implication-logic* =
  fixes *deduction* :: $'a \Rightarrow bool$ ($\vdash$ - [*60*] *55*)
  fixes *implication* :: $'a \Rightarrow {'a} \Rightarrow {'a}$ (**infixr** $\rightarrow$ *70*)
  assumes *axiom-k*: $\vdash \varphi \rightarrow \psi \rightarrow \varphi$
  assumes *axiom-s*: $\vdash (\varphi \rightarrow \psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \psi) \rightarrow \varphi \rightarrow \chi$
  assumes *modus-ponens*: $\vdash \varphi \rightarrow \psi \Longrightarrow \vdash \varphi \Longrightarrow \vdash \psi$

1.2 Common Rules

lemma (**in** *implication-logic*) *trivial-implication*:
  $\vdash \varphi \rightarrow \varphi$
  ⟨*proof*⟩

lemma (**in** *implication-logic*) *flip-implication*:
  $\vdash (\varphi \rightarrow \psi \rightarrow \chi) \rightarrow \psi \rightarrow \varphi \rightarrow \chi$
  ⟨*proof*⟩

lemma (**in** *implication-logic*) *hypothetical-syllogism*:
$\vdash (\psi \to \chi) \to (\varphi \to \psi) \to \varphi \to \chi$
$\langle proof \rangle$

lemma (**in** *implication-logic*) *flip-hypothetical-syllogism*:
$\vdash (\psi \to \varphi) \to (\varphi \to \chi) \to (\psi \to \chi)$
$\langle proof \rangle$

lemma (**in** *implication-logic*) *implication-absorption*:
$\vdash (\varphi \to \varphi \to \psi) \to \varphi \to \psi$
$\langle proof \rangle$

1.3 Lists of Assumptions

1.3.1 List Implication

Implication given a list of assumptions can be expressed recursively

primrec (**in** *implication-logic*)
list-implication :: *'a list* ⇒ *'a* ⇒ *'a* (**infix** :→ *80*) **where**
$[] :\to \varphi = \varphi$
$| \ (\psi \ \# \ \Psi) :\to \varphi = \psi \to \Psi :\to \varphi$

1.3.2 Deduction From a List of Assumptions

Deduction from a list of assumptions can be expressed in terms of $(:\to)$.

definition (**in** *implication-logic*) *list-deduction* :: *'a list* ⇒ *'a* ⇒ *bool* (**infix** :⊢ *60*)
where
$\Gamma :\vdash \varphi \equiv \vdash \Gamma :\to \varphi$

1.3.3 List Deduction as Implication Logic

The relation $(:\vdash)$ may naturally be interpreted as a *deduction* predicate for an instance of implication logic for a fixed list of assumptions Γ.

Analogues of the two axioms of implication logic can be naturally stated using list implication.

lemma (**in** *implication-logic*) *list-implication-axiom-k*:
$\vdash \varphi \to \Gamma :\to \varphi$
$\langle proof \rangle$

lemma (**in** *implication-logic*) *list-implication-axiom-s*:
$\vdash \Gamma :\to (\varphi \to \psi) \to \Gamma :\to \varphi \to \Gamma :\to \psi$
$\langle proof \rangle$

The lemmas $\vdash \varphi \to \Gamma :\to \varphi$ and $\vdash \Gamma :\to (\varphi \to \psi) \to \Gamma :\to \varphi \to \Gamma :\to \psi$ jointly give rise to an interpretation of implication logic, where a list of assumptions Γ play the role of a *background theory* of $(:\vdash)$.

context *implication-logic* **begin**
interpretation *list-deduction-logic*:
implication-logic $\lambda\ \varphi.\ \Gamma :\vdash \varphi\ (\to)$
⟨*proof*⟩
end

The following *weakening* rule can also be derived.

lemma (**in** *implication-logic*) *list-deduction-weaken*:
$\vdash \varphi \Longrightarrow \Gamma :\vdash \varphi$
⟨*proof*⟩

In the case of the empty list, the converse may be established.

lemma (**in** *implication-logic*) *list-deduction-base-theory* [*simp*]:
$[] :\vdash \varphi \equiv \vdash \varphi$
⟨*proof*⟩

lemma (**in** *implication-logic*) *list-deduction-modus-ponens*:
$\Gamma :\vdash \varphi \to \psi \Longrightarrow \Gamma :\vdash \varphi \Longrightarrow \Gamma :\vdash \psi$
⟨*proof*⟩

1.4 The Deduction Theorem

One result in the meta-theory of implication logic is the *deduction theorem*, which is a mechanism for moving antecedents back and forth from collections of assumptions.

To develop the deduction theorem, the following two lemmas generalize $\vdash (\varphi \to \psi \to \chi) \to \psi \to \varphi \to \chi$.

lemma (**in** *implication-logic*) *list-flip-implication1*:
$\vdash (\varphi \# \Gamma) :\to \chi \to \Gamma :\to (\varphi \to \chi)$
⟨*proof*⟩

lemma (**in** *implication-logic*) *list-flip-implication2*:
$\vdash \Gamma :\to (\varphi \to \chi) \to (\varphi \# \Gamma) :\to \chi$
⟨*proof*⟩

Together the two lemmas above suffice to prove a form of the deduction theorem:

theorem (**in** *implication-logic*) *list-deduction-theorem*:
$(\varphi \# \Gamma) :\vdash \psi = \Gamma :\vdash \varphi \to \psi$
⟨*proof*⟩

1.5 Monotonic Growth in Deductive Power

In logic, for two sets of assumptions Φ and Ψ, if $\Psi \subseteq \Phi$ then the latter theory Φ is said to be *stronger* than former theory Ψ. In principle, anything a weaker theory can prove a stronger theory can prove. One way of saying this is that deductive power increases monotonically with as the set of underlying assumptions grow.

The monotonic growth of deductive power can be expressed as a meta-theorem in implication logic.

The lemma $\vdash \Gamma :\rightarrow (\varphi \rightarrow \chi) \rightarrow (\varphi \# \Gamma) :\rightarrow \chi$ presents a means of *introducing* assumptions into a list of assumptions when those assumptions have been arrived at by an implication. The next lemma presents a means of *discharging* those assumptions, which can be used in the monotonic growth theorem to be proved.

lemma (**in** *implication-logic*) *list-implication-removeAll*:
$\vdash \Gamma :\rightarrow \psi \rightarrow (\mathit{removeAll}\ \varphi\ \Gamma) :\rightarrow (\varphi \rightarrow \psi)$
⟨*proof*⟩

From lemma above presents what is needed to prove that deductive power for lists is monotonic.

theorem (**in** *implication-logic*) *list-implication-monotonic*:
$\mathit{set}\ \Sigma \subseteq \mathit{set}\ \Gamma \Longrightarrow \vdash \Sigma :\rightarrow \varphi \rightarrow \Gamma :\rightarrow \varphi$
⟨*proof*⟩

A direct consequence is that deduction from lists of assumptions is monotonic as well:

theorem (**in** *implication-logic*) *list-deduction-monotonic*:
$\mathit{set}\ \Sigma \subseteq \mathit{set}\ \Gamma \Longrightarrow \Sigma :\vdash \varphi \Longrightarrow \Gamma :\vdash \varphi$
⟨*proof*⟩

1.6 The Deduction Theorem Revisited

The monotonic nature of deduction allows us to prove another form of the deduction theorem, where the assumption being discharged is completely removed from the list of assumptions.

theorem (**in** *implication-logic*) *alternate-list-deduction-theorem*:
$(\varphi \# \Gamma) :\vdash \psi = (\mathit{removeAll}\ \varphi\ \Gamma) :\vdash \varphi \rightarrow \psi$
⟨*proof*⟩

1.7 Reflection

In logic the *reflection* principle sometimes refers to when a collection of assumptions can deduce any of its members. It is automatically derivable from $[\![\mathit{set}\ \Sigma \subseteq \mathit{set}\ \Gamma;\ \Sigma :\vdash \varphi]\!] \Longrightarrow \Gamma :\vdash \varphi$ among the other rules provided.

lemma (**in** *implication-logic*) *list-deduction-reflection*:
$\varphi \in \mathit{set}\ \Gamma \Longrightarrow \Gamma :\vdash \varphi$
⟨*proof*⟩

1.8 The Cut Rule

Cut is a rule commonly presented in sequent calculi, dating back to Gerhard Gentzen's *Investigations in Logical Deduction* (1935) [4]

The cut rule is not generally necessary in sequent calculi. It can often be shown that the rule can be eliminated without reducing the power of the underlying logic. However, as demonstrated by George Boolos' *Don't Eliminate Cut* (1984) [3], removing the rule can often lead to very inefficient proof systems.

Here the rule is presented just as a meta theorem.

theorem (**in** *implication-logic*) *list-deduction-cut-rule*:
$(\varphi \# \Gamma) :\vdash \psi \Longrightarrow \Delta :\vdash \varphi \Longrightarrow \Gamma @ \Delta :\vdash \psi$
⟨*proof*⟩

The cut rule can also be strengthened to entire lists of propositions.

theorem (**in** *implication-logic*) *strong-list-deduction-cut-rule*:
$(\Phi @ \Gamma) :\vdash \psi \Longrightarrow \forall \ \varphi \in \mathit{set}\ \Phi.\ \Delta :\vdash \varphi \Longrightarrow \Gamma @ \Delta :\vdash \psi$
⟨*proof*⟩

1.9 Sets of Assumptions

While deduction in terms of lists of assumptions is straight-forward to define, deduction (and the *deduction theorem*) is commonly given in terms of *sets* of propositions. This formulation is suited to establishing strong completeness theorems and compactness theorems.

The presentation of deduction from a set follows the presentation of list deduction given for $(:\vdash)$.

1.10 Definition of Deduction

Just as deduction from a list $(:\vdash)$ can be defined in terms of $(:\rightarrow)$, deduction from a *set* of assumptions can be expressed in terms of $(:\vdash)$.

definition (**in** *implication-logic*) *set-deduction* :: *'a set* ⇒ *'a* ⇒ *bool* (**infix** ⊩ *60*)
where
$\Gamma \Vdash \varphi \equiv \exists \ \Psi.\ \mathit{set}\ \Psi \ \subseteq \Gamma \wedge \Psi :\vdash \varphi$

1.10.1 Interpretation as Implication Logic

As in the case of $(:\vdash)$, the relation $(\Vdash)$ may be interpreted as *deduction* predicate for a fixed set of assumptions Γ.

The following lemma is given in order to establish this, which asserts that every implication logic tautology $\vdash \varphi$ is also a tautology for $\Gamma \Vdash \varphi$.

lemma (**in** *implication-logic*) *set-deduction-weaken*:
$\vdash \varphi \Longrightarrow \Gamma \Vdash \varphi$
⟨*proof*⟩

In the case of the empty set, the converse may be established.

lemma (**in** *implication-logic*) *set-deduction-base-theory*:
$\{\} \Vdash \varphi \equiv \vdash \varphi$
⟨*proof*⟩

Next, a form of *modus ponens* is provided for $(\Vdash)$.

lemma (**in** *implication-logic*) *set-deduction-modus-ponens*:
$\Gamma \Vdash \varphi \to \psi \Longrightarrow \Gamma \Vdash \varphi \Longrightarrow \Gamma \Vdash \psi$
⟨*proof*⟩

context *implication-logic* **begin**
interpretation *set-deduction-logic*:
implication-logic $\lambda\ \varphi.\ \Gamma \Vdash \varphi\ (\to)$
⟨*proof*⟩
end

1.11 The Deduction Theorem

The next result gives the deduction theorem for $(\Vdash)$.

theorem (**in** *implication-logic*) *set-deduction-theorem*:
insert $\varphi\ \Gamma \Vdash \psi = \Gamma \Vdash \varphi \to \psi$
⟨*proof*⟩

1.12 Monotonic Growth in Deductive Power

In contrast to the $(:\vdash)$ relation, the proof that the deductive power of $(\Vdash)$ grows monotonically with its assumptions may be fully automated.

theorem *set-deduction-monotonic*:
$\Sigma \subseteq \Gamma \Longrightarrow \Sigma \Vdash \varphi \Longrightarrow \Gamma \Vdash \varphi$
⟨*proof*⟩

1.13 The Deduction Theorem Revisited

As a consequence of the fact that $[\![\Sigma \subseteq \Gamma; \Sigma \Vdash \varphi]\!] \Longrightarrow \Gamma \Vdash \varphi$ is automatically provable, an alternate *deduction theorem* where the discharged assumption is completely removed from the set of assumptions is just a consequence of the more conventional *insert* $\varphi\ \Gamma \Vdash \psi = \Gamma \Vdash \varphi \to \psi$ rule and some basic set identities.

theorem (**in** *implication-logic*) *alternate-set-deduction-theorem*:
  $\mathit{insert}\ \varphi\ \Gamma \Vdash \psi = \Gamma - \{\varphi\} \Vdash \varphi \to \psi$
  $\langle \mathit{proof} \rangle$

1.14 Reflection

Just as in the case of $(:\vdash)$, deduction from sets of assumptions makes true the *reflection principle* and is automatically provable.

theorem (**in** *implication-logic*) *set-deduction-reflection*:
  $\varphi \in \Gamma \Longrightarrow \Gamma \Vdash \varphi$
  $\langle \mathit{proof} \rangle$

1.15 The Cut Rule

The final principle of $(\Vdash)$ presented is the *cut rule.*

First, the weak form of the rule is established.

theorem (**in** *implication-logic*) *set-deduction-cut-rule*:
  $\mathit{insert}\ \varphi\ \Gamma \Vdash \psi \Longrightarrow \Delta \Vdash \varphi \Longrightarrow \Gamma \cup \Delta \Vdash \psi$
$\langle \mathit{proof} \rangle$

Another lemma is shown next in order to establish the strong form of the cut rule. The lemma shows the existence of a *covering list* of assumptions Ψ in the event some set of assumptions Δ proves everything in a finite set of assumptions Φ.

lemma (**in** *implication-logic*) *finite-set-deduction-list-deduction*:
  assumes *finite* Φ
  and $\forall\ \varphi \in \Phi.\ \Delta \Vdash \varphi$
  shows $\exists\,\Psi.\ \mathit{set}\ \Psi \subseteq \Delta \wedge (\forall\varphi \in \Phi.\ \Psi :\vdash \varphi)$
  $\langle \mathit{proof} \rangle$

With $[\![\mathit{finite}\ \Phi;\ \forall\varphi{\in}\Phi.\ \Delta \Vdash \varphi]\!] \Longrightarrow \exists\,\Psi.\ \mathit{set}\ \Psi \subseteq \Delta \wedge (\forall\varphi{\in}\Phi.\ \Psi :\vdash \varphi)$ the strengthened form of the cut rule can be given.

theorem (**in** *implication-logic*) *strong-set-deduction-cut-rule*:
  assumes $\Phi \cup \Gamma \Vdash \psi$
  and $\forall\ \varphi \in \Phi.\ \Delta \Vdash \varphi$
  shows $\Gamma \cup \Delta \Vdash \psi$
$\langle \mathit{proof} \rangle$

1.16 Maximally Consistent Sets For Implication Logic

Maximally Consistent Sets are a common construction for proving completeness of logical calculi. For a classic presentation, see Dirk van Dalen's *Logic and Structure* (2013, §1.5, pgs. 42–45) [8].

Maximally consistent sets will form the foundation of all of the model theory we will employ in this text. In fact, apart from classical logic semantics, conventional model theory will not be used at all.

The models we are centrally concerned are derived from maximally consistent sets. These include probability measures used in completeness theorems of probability logic found in §**??**, as well as arbitrage protection and trading strategies stipulated by our formulation of the *Dutch Book Theorem* we present in §**??**.

Since implication logic does not have *falsum*, consistency is defined relative to a formula φ.

definition (**in** *implication-logic*)
formula-consistent :: $'a \Rightarrow {}'a\ set \Rightarrow bool$ (*-–consistent* - [*100*] *100*)
where
[*simp*]: $\varphi-consistent\ \Gamma \equiv \neg\ (\Gamma \Vdash \varphi)$

Since consistency is defined relative to some φ, *maximal consistency* is presented as asserting that either ψ or $\psi \to \varphi$ is in the consistent set Γ, for all ψ. This coincides with the traditional definition in classical logic when φ is *falsum*.

definition (**in** *implication-logic*)
formula-maximally-consistent-set-def :: $'a \Rightarrow {}'a\ set \Rightarrow bool$ (*-–MCS* - [*100*] *100*)
where
[*simp*]: $\varphi-MCS\ \Gamma \equiv (\varphi-consistent\ \Gamma) \wedge (\forall\ \psi.\ \psi \in \Gamma \vee (\psi \to \varphi) \in \Gamma)$

Every consistent set Γ may be extended to a maximally consistent set.

However, no assumption is made regarding the cardinality of the types of an instance of *implication-logic*.

As a result, typical proofs that assume a countable domain are not suitable. Our proof leverages *Zorn's lemma*.

lemma (**in** *implication-logic*) *formula-consistent-extension*:
assumes $\varphi-consistent\ \Gamma$
shows $(\varphi-consistent\ (insert\ \psi\ \Gamma)) \vee (\varphi-consistent\ (insert\ (\psi \to \varphi)\ \Gamma))$
⟨*proof*⟩

theorem (**in** *implication-logic*) *formula-maximally-consistent-extension*:
assumes $\varphi-consistent\ \Gamma$
shows $\exists\ \Omega.\ (\varphi-MCS\ \Omega) \wedge \Gamma \subseteq \Omega$
⟨*proof*⟩

Finally, maximally consistent sets contain anything that can be deduced from them, and model a form of *modus ponens*.

lemma (**in** *implication-logic*) *formula-maximally-consistent-set-def-reflection*:
$\varphi-MCS\ \Gamma \Longrightarrow \psi \in \Gamma = \Gamma \Vdash \psi$

$\langle proof \rangle$

theorem (**in** *implication-logic*) *formula-maximally-consistent-set-def-implication-elimination*:
assumes $\varphi\text{−}MCS\ \Omega$
shows $(\psi \rightarrow \chi) \in \Omega \Longrightarrow \psi \in \Omega \Longrightarrow \chi \in \Omega$
$\langle proof \rangle$

This concludes our introduction to implication logic.

end

Chapter 2

Classical Propositional Logic

theory *Classical-Logic*
  imports *Implication-Logic*
begin

This theory presents *classical propositional logic*, which is classical logic without quantifiers.

2.1 Axiomatization

Classical propositional logic can be given by the following Hilbert-style axiom system. It is *implication-logic* extended with *falsum* and double negation.

class *classical-logic* = *implication-logic* +
  fixes *falsum* :: $'a$ ($\bot$)
  assumes *double-negation*: $\vdash (((\varphi \to \bot) \to \bot) \to \varphi)$

In some cases it is useful to assume consistency as an axiom:

class *consistent-classical-logic* = *classical-logic* +
  assumes *consistency*: $\neg \vdash \bot$

2.2 Common Rules

There are many common tautologies in classical logic. Once we have established *completeness* in §3, we will be able to leverage Isabelle/HOL's automation for proving these elementary results.

In order to bootstrap completeness, we develop some common lemmas using classical deduction alone.

lemma (**in** *classical-logic*)
  ex-falso-quodlibet: $\vdash \bot \to \varphi$
  ⟨*proof*⟩

lemma (**in** *classical-logic*)
Contraposition: $\vdash ((\varphi \to \bot) \to (\psi \to \bot)) \to \psi \to \varphi$
⟨*proof*⟩

lemma (**in** *classical-logic*)
double-negation-converse: $\vdash \varphi \to (\varphi \to \bot) \to \bot$
⟨*proof*⟩

The following lemma is sometimes referred to as *The Principle of Pseudo-Scotus*[2].

lemma (**in** *classical-logic*)
pseudo-scotus: $\vdash (\varphi \to \bot) \to \varphi \to \psi$
⟨*proof*⟩

Another popular lemma is attributed to Charles Sanders Peirce, and has come to be known as *Peirces Law*[5].

lemma (**in** *classical-logic*) *Peirces-law*:
$\vdash ((\varphi \to \psi) \to \varphi) \to \varphi$
⟨*proof*⟩

lemma (**in** *classical-logic*) *excluded-middle-elimination*:
$\vdash (\varphi \to \psi) \to ((\varphi \to \bot) \to \psi) \to \psi$
⟨*proof*⟩

2.3 Maximally Consistent Sets For Classical Logic

Relativized maximally consistent sets were introduced in §1.16. Often this is exactly what we want in a proof. A completeness theorem typically starts by assuming φ is not provable, then finding a $\varphi{-}MCS\ \Gamma$ which gives rise to a model which does not make φ true.

A more conventional presentation says that Γ is maximally consistent if and only if $\neg\, \Gamma \Vdash \bot$ and $\forall\, \psi.\ \psi \in \Gamma \lor \psi \to \varphi \in \Gamma$. This conventional presentation will come up when formulating MaxSAT in §**??**. This in turn allows us to formulate MaxSAT completeness for probability inequalities in §**??**, and reduce checking if a strategy will always lose money or if it will always make money if matched to bounded MaxSAT as part of our proof of the *Dutch Book Theorem* in §**??** and §**??** respectively.

definition (**in** *classical-logic*)
consistent :: *'a set* ⇒ *bool* **where**
[*simp*]: *consistent* $\Gamma \equiv \bot{-}$*consistent* Γ

definition (**in** *classical-logic*)
maximally-consistent-set :: *'a set* ⇒ *bool* (*MCS*) **where**

$[simp]$: $MCS\ \Gamma \equiv \bot{-}MCS\ \Gamma$

lemma (**in** *classical-logic*)
formula-maximally-consistent-set-def-negation: $\varphi{-}MCS\ \Gamma \Longrightarrow \varphi \to \bot \in \Gamma$
$\langle proof \rangle$

Relative maximal consistency and conventional maximal consistency in fact coincide in classical logic.

lemma (**in** *classical-logic*)
formula-maximal-consistency: $(\exists\,\varphi.\ \varphi{-}MCS\ \Gamma) = MCS\ \Gamma$
$\langle proof \rangle$

Finally, classical logic allows us to strengthen $[\![\varphi{-}MCS\ \Omega;\ \psi \to \chi \in \Omega;\ \psi \in \Omega]\!] \Longrightarrow \chi \in \Omega$ to a biconditional.

lemma (**in** *classical-logic*)
formula-maximally-consistent-set-def-implication:
assumes $\varphi{-}MCS\ \Gamma$
shows $\psi \to \chi \in \Gamma = (\psi \in \Gamma \longrightarrow \chi \in \Gamma)$
$\langle proof \rangle$

end

Chapter 3

Classical Soundness and Completeness

theory *Classical-Logic-Completeness*
  imports *Classical-Logic*
begin

The following presents soundness completeness of the classical propositional calculus for propositional semantics. The classical propositional calculus is sometimes referred to as the *sentential calculus*. We give a concrete algebraic data type for propositional formulae in §3.1. We inductively define a logical judgement $\vdash_{prop}$ for these formulae. We also define the Tarski truth relation $\models_{prop}$ inductively, which we present in §3.3.

The most significant results here are the *embedding theorems*. These theorems show that the propositional calculus can be embedded in any logic extending *classical-logic*. These theorems are proved in §3.5.

3.1 Syntax

Here we provide the usual language for formulae in the propositional calculus. It contains *falsum* $\perp$, implication $(\rightarrow)$, and a way of constructing *atomic* propositions $\lambda\ \varphi\ .\ \langle\ \varphi\ \rangle$. Defining the language is straight-forward using an algebraic data type.

datatype *'a classical-propositional-formula* =
    Falsum $(\perp)$
  | *Proposition* *'a* $(\langle\ \text{-}\ \rangle\ [45])$
  | *Implication*
      'a classical-propositional-formula
      'a classical-propositional-formula (**infixr** $\rightarrow$ *70*)

3.2 Propositional Calculus

In this section we recursively define what a proof is in the classical propositional calculus. We provide the familiar K and S axioms, as well as *double negation* and *modus ponens*.

named-theorems *classical-propositional-calculus*
Rules for the Propositional Calculus

inductive *classical-propositional-calculus* ::
$'a$ *classical-propositional-formula* $\Rightarrow$ *bool* ($\vdash_{prop}$ - [*60*] *55*)
where
axiom-k [*classical-propositional-calculus*]:
$\vdash_{prop} \varphi \to \psi \to \varphi$
| *axiom-s* [*classical-propositional-calculus*]:
$\vdash_{prop} (\varphi \to \psi \to \chi) \to (\varphi \to \psi) \to \varphi \to \chi$
| *double-negation* [*classical-propositional-calculus*]:
$\vdash_{prop} ((\varphi \to \bot) \to \bot) \to \varphi$
| *modus-ponens* [*classical-propositional-calculus*]:
$\vdash_{prop} \varphi \to \psi \Longrightarrow \vdash_{prop} \varphi \Longrightarrow \vdash_{prop} \psi$

Our proof system for our propositional calculus is trivially an instance of *classical-logic*. The introduction rules for $\vdash_{prop}$ naturally reflect the axioms of the classical logic axiom class.

instantiation *classical-propositional-formula*
:: (*type*) *classical-logic*
begin
definition [*simp*]: $\bot = \bot$
definition [*simp*]: $\vdash \varphi = \vdash_{prop} \varphi$
definition [*simp*]: $\varphi \to \psi = \varphi \to \psi$
instance $\langle$*proof*$\rangle$
end

3.3 Propositional Semantics

Below we give the typical definition of the Tarski truth relation $\models_{prop}$.

primrec *classical-propositional-semantics* ::
$'a$ *set* $\Rightarrow$ $'a$ *classical-propositional-formula* $\Rightarrow$ *bool*
(**infix** $\models_{prop}$ *65*)
where
$\mathfrak{M} \models_{prop} \langle\ p\ \rangle = (p \in \mathfrak{M})$
| $\mathfrak{M} \models_{prop} \varphi \to \psi = (\mathfrak{M} \models_{prop} \varphi \longrightarrow \mathfrak{M} \models_{prop} \psi)$
| $\mathfrak{M} \models_{prop} \bot = False$

Soundness of our calculus for these semantics is trivial.

theorem *classical-propositional-calculus-soundness*:
$\vdash_{prop} \varphi \Longrightarrow \mathfrak{M} \models_{prop} \varphi$
$\langle$*proof*$\rangle$

3.4 Soundness and Completeness Proofs

definition *strong-classical-propositional-deduction* ::
'a classical-propositional-formula set
$\Rightarrow$ *'a classical-propositional-formula* $\Rightarrow$ *bool*
(**infix** $\Vdash_{prop}$ *65*)
where
[*simp*]: $\Gamma \Vdash_{prop} \varphi \equiv \Gamma \Vdash \varphi$

definition *strong-classical-propositional-tarski-truth* ::
'a classical-propositional-formula set
$\Rightarrow$ *'a classical-propositional-formula* $\Rightarrow$ *bool*
(**infix** $\Vvdash_{prop}$ *65*)
where
[*simp*]: $\Gamma \Vvdash_{prop} \varphi \equiv \forall\ \mathfrak{M}.(\forall\ \gamma \in \Gamma.\ \mathfrak{M} \models_{prop} \gamma) \longrightarrow \mathfrak{M} \models_{prop} \varphi$

definition *theory-propositions* ::
'a classical-propositional-formula set $\Rightarrow$ *'a set* ($\{\!|$ - $|\!\}$ [*50*])
where
[*simp*]: $\{\!|\ \Gamma\ |\!\} = \{p\ .\ \Gamma \Vdash_{prop} \langle\ p\ \rangle\}$

Below we give the main lemma for completeness: the *truth lemma*. This proof connects the maximally consistent sets developed in §1.16 and §2.3 with the semantics given in §3.3.

All together, the technique we are using essentially follows the approach by Blackburn et al. [1, §4.2, pgs. 196-201].

lemma *truth-lemma*:
assumes *MCS* Γ
shows $\Gamma \Vdash_{prop} \varphi \equiv \{\!|\ \Gamma\ |\!\} \models_{prop} \varphi$
⟨*proof*⟩

Here the truth lemma above is combined with $\varphi-$*consistent* $\Gamma \Longrightarrow \exists\,\Omega.$ $\varphi-$*MCS* $\Omega \wedge \Gamma \subseteq \Omega$ proven in §3.3. These theorems together give rise to strong completeness for the propositional calculus.

theorem *classical-propositional-calculus-strong-soundness-and-completeness*:
$\Gamma \Vdash_{prop} \varphi = \Gamma \Vvdash_{prop} \varphi$
⟨*proof*⟩

For our applications in §sec:propositional-embedding, we will only need a weaker form of soundness and completeness rather than the stronger form proved above.

theorem *classical-propositional-calculus-soundness-and-completeness*:
$\vdash_{prop} \varphi = (\forall\mathfrak{M}.\ \mathfrak{M} \models_{prop} \varphi)$
⟨*proof*⟩

instantiation *classical-propositional-formula*
:: (*type*) *consistent-classical-logic*

begin
instance ⟨*proof*⟩
end

3.5 Embedding Theorem For the Propositional Calculus

A recurring technique to prove theorems in logic moving forward is *embed* our theorem into the classical propositional calculus.

Using our embedding, we can leverage completeness to turn our problem into semantics and dispatch to Isabelle/HOL's classical theorem provers.

In future work we may make a tactic for this, but for now we just manually leverage the technique throughout our subsequent proofs.

primrec (**in** *classical-logic*)
  classical-propositional-formula-embedding
  :: $'a$ *classical-propositional-formula* $\Rightarrow$ $'a$ ($(\!|$ - $|\!)$ [*50*]) **where**
    $(\!|\ \langle\ p\ \rangle\ |\!) = p$
  $|\ (\!|\ \varphi \rightarrow \psi\ |\!) = (\!|\ \varphi\ |\!) \rightarrow (\!|\ \psi\ |\!)$
  $|\ (\!|\ \bot\ |\!) = \bot$

theorem (**in** *classical-logic*) *propositional-calculus*:
 $\vdash_{prop} \varphi \Longrightarrow \vdash (\!|\ \varphi\ |\!)$
 ⟨*proof*⟩

The following theorem in particular shows that it suffices to prove theorems using classical semantics to prove theorems about the logic under investigation.

theorem (**in** *classical-logic*) *propositional-semantics*:
 $\forall \mathfrak{M}.\ \mathfrak{M} \models_{prop} \varphi \Longrightarrow \vdash (\!|\ \varphi\ |\!)$
 ⟨*proof*⟩

end

Chapter 4

List Utility Theorems

theory *List-Utilities*
  imports
    HOL−Combinatorics.List-Permutation
begin

Throughout our work it will be necessary to reuse common lemmas regarding lists and multisets. These results are proved in the following section and reused by subsequent lemmas and theorems.

4.1 Multisets

lemma *length-sub-mset*:
  assumes *mset* $\Psi \subseteq\#$ *mset* Γ
    and *length* Ψ >= *length* Γ
   shows *mset* Ψ = *mset* Γ
  ⟨*proof*⟩

lemma *set-exclusion-mset-simplify*:
  assumes $\neg$ ($\exists$ $\psi \in$ *set* Ψ. $\psi \in$ *set* Σ)
    and *mset* $\Psi \subseteq\#$ *mset* (Σ @ Γ)
   shows *mset* $\Psi \subseteq\#$ *mset* Γ
⟨*proof*⟩

lemma *image-mset-cons-homomorphism*:
 image-mset mset (*image-mset* ((#) φ) Φ) = *image-mset* ((+) {# φ #}) (*image-mset mset* Φ)
  ⟨*proof*⟩

lemma *image-mset-append-homomorphism*:
 image-mset mset (*image-mset* ((@) Δ) Φ) = *image-mset* ((+) (*mset* Δ)) (*image-mset mset* Φ)
  ⟨*proof*⟩

lemma *image-mset-add-collapse*:

fixes $A\ B :: {}'a\ multiset$
shows $image\text{-}mset\ ((+)\ A)\ (image\text{-}mset\ ((+)\ B)\ X) = image\text{-}mset\ ((+)\ (A + B))\ X$
$\langle proof \rangle$

lemma *remove1-remdups-removeAll*: $remove1\ x\ (remdups\ A) = remdups\ (removeAll\ x\ A)$
$\langle proof \rangle$

lemma *mset-remdups*:
assumes $mset\ A = mset\ B$
shows $mset\ (remdups\ A) = mset\ (remdups\ B)$
$\langle proof \rangle$

lemma *mset-mset-map-snd-remdups*:
assumes $mset\ (map\ mset\ A) = mset\ (map\ mset\ B)$
shows $mset\ (map\ (mset \circ (map\ snd) \circ remdups)\ A) = mset\ (map\ (mset \circ (map\ snd) \circ remdups)\ B)$
$\langle proof \rangle$

lemma *mset-remdups-append-msub*:
$mset\ (remdups\ A) \subseteq\#\ mset\ (remdups\ (B\ @\ A))$
$\langle proof \rangle$

4.2 List Mapping

The following notation for permutations is slightly nicer when formatted in LaTeX.

notation *perm* (**infix** $\rightleftharpoons$ *50*)

lemma *map-monotonic*:
assumes $mset\ A \subseteq\#\ mset\ B$
shows $mset\ (map\ f\ A) \subseteq\#\ mset\ (map\ f\ B)$
$\langle proof \rangle$

lemma *perm-map-perm-list-exists*:
assumes $A \rightleftharpoons map\ f\ B$
shows $\exists\ B'.\ A = map\ f\ B' \wedge B' \rightleftharpoons B$
$\langle proof \rangle$

lemma *mset-sub-map-list-exists*:
assumes $mset\ \Phi \subseteq\#\ mset\ (map\ f\ \Gamma)$
shows $\exists\ \Phi'.\ mset\ \Phi' \subseteq\#\ mset\ \Gamma \wedge \Phi = (map\ f\ \Phi')$
$\langle proof \rangle$

4.3 Laws for Searching a List

lemma *find-Some-predicate*:
assumes $\mathit{find}\ P\ \Psi = \mathit{Some}\ \psi$
shows $P\ \psi$
$\langle \mathit{proof} \rangle$

lemma *find-Some-set-membership*:
assumes $\mathit{find}\ P\ \Psi = \mathit{Some}\ \psi$
shows $\psi \in \mathit{set}\ \Psi$
$\langle \mathit{proof} \rangle$

4.4 Permutations

lemma *perm-count-list*:
assumes $\Phi \rightleftharpoons \Psi$
shows $\mathit{count\text{-}list}\ \Phi\ \varphi = \mathit{count\text{-}list}\ \Psi\ \varphi$
$\langle \mathit{proof} \rangle$

lemma *count-list-append*:
$\mathit{count\text{-}list}\ (A\ @\ B)\ a = \mathit{count\text{-}list}\ A\ a + \mathit{count\text{-}list}\ B\ a$
$\langle \mathit{proof} \rangle$

lemma *concat-remove1*:
assumes $\Psi \in \mathit{set}\ \mathcal{L}$
shows $\mathit{concat}\ \mathcal{L} \rightleftharpoons \Psi\ @\ \mathit{concat}\ (\mathit{remove1}\ \Psi\ \mathcal{L})$
$\langle \mathit{proof} \rangle$

lemma *concat-set-membership-mset-containment*:
assumes $\mathit{concat}\ \Gamma \rightleftharpoons \Lambda$
and $\Phi \in \mathit{set}\ \Gamma$
shows $\mathit{mset}\ \Phi \subseteq\#\ \mathit{mset}\ \Lambda$
$\langle \mathit{proof} \rangle$

lemma (**in** *comm-monoid-add*) *perm-list-summation*:
assumes $\Psi \rightleftharpoons \Phi$
shows $(\sum \psi' \leftarrow \Psi.\ f\ \psi') = (\sum \varphi' \leftarrow \Phi.\ f\ \varphi')$
$\langle \mathit{proof} \rangle$

4.5 List Duplicates

primrec $\mathit{duplicates} :: 'a\ \mathit{list} \Rightarrow 'a\ \mathit{set}$
where
$\mathit{duplicates}\ [] = \{\}$
$\mid \mathit{duplicates}\ (x\ \#\ xs) =$
$(\mathit{if}\ (x \in \mathit{set}\ xs)$
$\mathit{then}\ \mathit{insert}\ x\ (\mathit{duplicates}\ xs)$
$\mathit{else}\ \mathit{duplicates}\ xs)$

lemma *duplicates-subset*:
  $duplicates\ \Phi \subseteq set\ \Phi$
  $\langle proof \rangle$

lemma *duplicates-alt-def*:
  $duplicates\ xs = \{x.\ count\text{-}list\ xs\ x \geq 2\}$
$\langle proof \rangle$

4.6 List Subtraction

primrec *list-subtract* :: $'a\ list \Rightarrow 'a\ list \Rightarrow 'a\ list$ (**infixl** $\ominus$ *70*)
  where
    $xs \ominus [] = xs$
  $|\ xs \ominus (y\ \#\ ys) = (remove1\ y\ (xs \ominus ys))$

lemma *list-subtract-mset-homomorphism* [*simp*]:
  $mset\ (A \ominus B) = mset\ A - mset\ B$
  $\langle proof \rangle$

lemma *list-subtract-empty* [*simp*]:
  $[] \ominus \Phi = []$
  $\langle proof \rangle$

lemma *list-subtract-remove1-cons-perm*:
  $\Phi \ominus (\varphi\ \#\ \Lambda) \rightleftharpoons (remove1\ \varphi\ \Phi) \ominus \Lambda$
  $\langle proof \rangle$

lemma *list-subtract-cons*:
  assumes $\varphi \notin set\ \Lambda$
  shows $(\varphi\ \#\ \Phi) \ominus \Lambda = \varphi\ \#\ (\Phi \ominus \Lambda)$
  $\langle proof \rangle$

lemma *list-subtract-cons-absorb*:
  assumes $count\text{-}list\ \Phi\ \varphi \geq count\text{-}list\ \Lambda\ \varphi$
  shows $\varphi\ \#\ (\Phi \ominus \Lambda) \rightleftharpoons (\varphi\ \#\ \Phi) \ominus \Lambda$
  $\langle proof \rangle$

lemma *list-subtract-cons-remove1-perm*:
  assumes $\varphi \in set\ \Lambda$
  shows $(\varphi\ \#\ \Phi) \ominus \Lambda \rightleftharpoons \Phi \ominus (remove1\ \varphi\ \Lambda)$
  $\langle proof \rangle$

lemma *list-subtract-removeAll-perm*:
  assumes $count\text{-}list\ \Phi\ \varphi \leq count\text{-}list\ \Lambda\ \varphi$
  shows $\Phi \ominus \Lambda \rightleftharpoons (removeAll\ \varphi\ \Phi) \ominus (removeAll\ \varphi\ \Lambda)$
  $\langle proof \rangle$

lemma *list-subtract-permute*:

assumes $\Phi \rightleftharpoons \Psi$
shows $\Phi \ominus \Lambda \rightleftharpoons \Psi \ominus \Lambda$
⟨*proof*⟩

lemma *append-perm-list-subtract-intro*:
assumes $A \rightleftharpoons B\ @\ C$
shows $A \ominus C \rightleftharpoons B$
⟨*proof*⟩

lemma *list-subtract-concat*:
assumes $\Psi \in \mathit{set}\ \mathcal{L}$
shows $\mathit{concat}\ (\mathcal{L} \ominus [\Psi]) \rightleftharpoons (\mathit{concat}\ \mathcal{L}) \ominus \Psi$
⟨*proof*⟩

lemma (**in** *comm-monoid-add*) *listSubstract-multisubset-list-summation*:
assumes $\mathit{mset}\ \Psi \subseteq\#\ \mathit{mset}\ \Phi$
shows $(\sum \psi \leftarrow \Psi.\ f\ \psi) + (\sum \varphi' \leftarrow (\Phi \ominus \Psi).\ f\ \varphi') = (\sum \varphi' \leftarrow \Phi.\ f\ \varphi')$
⟨*proof*⟩

lemma *list-subtract-set-difference-lower-bound*:
$\mathit{set}\ \Gamma - \mathit{set}\ \Phi \subseteq \mathit{set}\ (\Gamma \ominus \Phi)$
⟨*proof*⟩

lemma *list-subtract-set-trivial-upper-bound*:
$\mathit{set}\ (\Gamma \ominus \Phi) \subseteq \mathit{set}\ \Gamma$
⟨*proof*⟩

lemma *list-subtract-msub-eq*:
assumes $\mathit{mset}\ \Phi \subseteq\#\ \mathit{mset}\ \Gamma$
and $\mathit{length}\ (\Gamma \ominus \Phi) = m$
shows $\mathit{length}\ \Gamma = m + \mathit{length}\ \Phi$
⟨*proof*⟩

lemma *list-subtract-not-member*:
assumes $b \notin \mathit{set}\ A$
shows $A \ominus B = A \ominus (\mathit{remove1}\ b\ B)$
⟨*proof*⟩

lemma *list-subtract-monotonic*:
assumes $\mathit{mset}\ A \subseteq\#\ \mathit{mset}\ B$
shows $\mathit{mset}\ (A \ominus C) \subseteq\#\ \mathit{mset}\ (B \ominus C)$
⟨*proof*⟩

lemma *map-list-subtract-mset-containment*:
$\mathit{mset}\ ((\mathit{map}\ f\ A) \ominus (\mathit{map}\ f\ B)) \subseteq\#\ \mathit{mset}\ (\mathit{map}\ f\ (A \ominus B))$
⟨*proof*⟩

lemma *map-list-subtract-mset-equivalence*:
assumes $\mathit{mset}\ B \subseteq\#\ \mathit{mset}\ A$

shows *mset* ((*map f A*) $\ominus$ (*map f B*)) = *mset* (*map f* (*A* $\ominus$ *B*))
⟨*proof*⟩

lemma *msub-list-subtract-elem-cons-msub*:
assumes *mset* Ξ $\subseteq\#$ *mset* Γ
and $\psi \in$ *set* ($\Gamma \ominus \Xi$)
shows *mset* (ψ # Ξ) $\subseteq\#$ *mset* Γ
⟨*proof*⟩

4.7 Tuple Lists

lemma *remove1-pairs-list-projections-fst*:
assumes (γ,σ) $\in\#$ *mset* Φ
shows *mset* (*map fst* (*remove1* (γ, σ) Φ)) = *mset* (*map fst* Φ) − {# γ #}
⟨*proof*⟩

lemma *remove1-pairs-list-projections-snd*:
assumes (γ,σ) $\in\#$ *mset* Φ
shows *mset* (*map snd* (*remove1* (γ, σ) Φ)) = *mset* (*map snd* Φ) − {# σ #}
⟨*proof*⟩

lemma *triple-list-exists*:
assumes *mset* (*map snd* Ψ) $\subseteq\#$ *mset* Σ
and *mset* Σ $\subseteq\#$ *mset* (*map snd* Δ)
shows $\exists$ Ω. *map* (λ (ψ, σ, -). (ψ, σ)) Ω = Ψ $\wedge$
mset (*map* (λ (-, σ, γ). (γ, σ)) Ω) $\subseteq\#$ *mset* Δ
⟨*proof*⟩

4.8 List Intersection

primrec *list-intersect* :: *'a list => 'a list => 'a list* (**infixl** $\cap$ *60*)
where
- $\cap$ [] = []
| *xs* $\cap$ (*y* # *ys*) =
(*if* (*y* $\in$ *set xs*)
then (*y* # (*remove1 y xs* $\cap$ *ys*))
else (*xs* $\cap$ *ys*))

lemma *list-intersect-mset-homomorphism* [*simp*]:
mset ($\Phi \cap \Psi$) = *mset* Φ $\cap\#$ *mset* Ψ
⟨*proof*⟩

lemma *list-intersect-left-empty* [*simp*]: [] $\cap$ Φ = [] ⟨*proof*⟩

lemma *list-diff-intersect-comp*:
mset Φ = *mset* ($\Phi \ominus \Psi$) + *mset* ($\Phi \cap \Psi$)
⟨*proof*⟩

lemma *list-intersect-left-project*: *mset* $(\Phi \cap \Psi) \subseteq\#$ *mset* Φ
⟨*proof*⟩

lemma *list-intersect-right-project*: *mset* $(\Phi \cap \Psi) \subseteq\#$ *mset* Ψ
⟨*proof*⟩

end

Chapter 5

Classical Logic Connectives

theory *Classical-Connectives*
  imports
    Classical-Logic-Completeness
    List-Utilities
begin

Here we define the usual connectives for classical logic.

no-notation *FuncSet.funcset* (**infixr** $\rightarrow$ *60*)

5.1 Verum

definition (**in** *classical-logic*) *verum* :: $'a$ ($\top$)
  where
    $\top = \bot \rightarrow \bot$

lemma (**in** *classical-logic*) *verum-tautology* [*simp*]: $\vdash \top$
  $\langle proof \rangle$

lemma *verum-semantics* [*simp*]:
  $\mathfrak{M} \models_{prop} \top$
  $\langle proof \rangle$

lemma (**in** *classical-logic*) *verum-embedding* [*simp*]:
  $(\!|\ \top\ |\!) = \top$
  $\langle proof \rangle$

5.2 Conjunction

definition (**in** *classical-logic*)
  conjunction :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixr** $\sqcap$ *67*)
  where
    $\varphi \sqcap \psi = (\varphi \rightarrow \psi \rightarrow \bot) \rightarrow \bot$

primrec (**in** *classical-logic*)
arbitrary-conjunction :: $'a$ *list* $\Rightarrow$ $'a$ ($\bigsqcap$)
where
$\bigsqcap\ [] = \top$
$|\ \bigsqcap\ (\varphi \# \Phi) = \varphi \sqcap \bigsqcap\ \Phi$

lemma (**in** *classical-logic*) *conjunction-introduction*:
$\vdash \varphi \to \psi \to (\varphi \sqcap \psi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *conjunction-left-elimination*:
$\vdash (\varphi \sqcap \psi) \to \varphi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *conjunction-right-elimination*:
$\vdash (\varphi \sqcap \psi) \to \psi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *conjunction-embedding* [*simp*]:
$(\!|\ \varphi \sqcap \psi\ |\!) = (\!|\ \varphi\ |\!) \sqcap (\!|\ \psi\ |\!)$
$\langle proof \rangle$

lemma *conjunction-semantics* [*simp*]:
$\mathfrak{M} \models_{prop} \varphi \sqcap \psi = (\mathfrak{M} \models_{prop} \varphi \wedge \mathfrak{M} \models_{prop} \psi)$
$\langle proof \rangle$

5.3 Biconditional

definition (**in** *classical-logic*) *biconditional* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixr** $\leftrightarrow$ *75*)
where
$\varphi \leftrightarrow \psi = (\varphi \to \psi) \sqcap (\psi \to \varphi)$

lemma (**in** *classical-logic*) *biconditional-introduction*:
$\vdash (\varphi \to \psi) \to (\psi \to \varphi) \to (\varphi \leftrightarrow \psi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *biconditional-left-elimination*:
$\vdash (\varphi \leftrightarrow \psi) \to \varphi \to \psi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *biconditional-right-elimination*:
$\vdash (\varphi \leftrightarrow \psi) \to \psi \to \varphi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *biconditional-embedding* [*simp*]:
$(\!|\ \varphi \leftrightarrow \psi\ |\!) = (\!|\ \varphi\ |\!) \leftrightarrow (\!|\ \psi\ |\!)$
$\langle proof \rangle$

lemma *biconditional-semantics* [*simp*]:

$\mathfrak{M} \models_{prop} \varphi \leftrightarrow \psi = (\mathfrak{M} \models_{prop} \varphi \longleftrightarrow \mathfrak{M} \models_{prop} \psi)$
⟨*proof*⟩

5.4 Negation

definition (**in** *classical-logic*) *negation* :: $'a \Rightarrow 'a$ ($\sim$)
where
$\sim \varphi = \varphi \to \bot$

lemma (**in** *classical-logic*) *negation-introduction*:
$\vdash (\varphi \to \bot) \to \sim \varphi$
⟨*proof*⟩

lemma (**in** *classical-logic*) *negation-elimination*:
$\vdash \sim \varphi \to (\varphi \to \bot)$
⟨*proof*⟩

lemma (**in** *classical-logic*) *negation-embedding* [*simp*]:
$(\!| \sim \varphi |\!) = \sim (\!| \varphi |\!)$
⟨*proof*⟩

lemma *negation-semantics* [*simp*]:
$\mathfrak{M} \models_{prop} \sim \varphi = (\neg\, \mathfrak{M} \models_{prop} \varphi)$
⟨*proof*⟩

5.5 Disjunction

definition (**in** *classical-logic*) *disjunction* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixr** $\sqcup$ *67*)
where
$\varphi \sqcup \psi = (\varphi \to \bot) \to \psi$

primrec (**in** *classical-logic*) *arbitrary-disjunction* :: $'a$ *list* $\Rightarrow 'a$ ($\bigsqcup$)
where
$\bigsqcup\ [] = \bot$
$|\ \bigsqcup\ (\varphi \# \Phi) = \varphi \sqcup \bigsqcup\ \Phi$

lemma (**in** *classical-logic*) *disjunction-elimination*:
$\vdash (\varphi \to \chi) \to (\psi \to \chi) \to (\varphi \sqcup \psi) \to \chi$
⟨*proof*⟩

lemma (**in** *classical-logic*) *disjunction-left-introduction*:
$\vdash \varphi \to (\varphi \sqcup \psi)$
⟨*proof*⟩

lemma (**in** *classical-logic*) *disjunction-right-introduction*:
$\vdash \psi \to (\varphi \sqcup \psi)$
⟨*proof*⟩

lemma (**in** *classical-logic*) *disjunction-embedding* [*simp*]:
$(\!|\ \varphi \sqcup \psi\ |\!) = (\!|\ \varphi\ |\!) \sqcup (\!|\ \psi\ |\!)$
$\langle proof \rangle$

lemma *disjunction-semantics* [*simp*]:
$\mathfrak{M} \models_{prop} \varphi \sqcup \psi = (\mathfrak{M} \models_{prop} \varphi \vee \mathfrak{M} \models_{prop} \psi)$
$\langle proof \rangle$

5.6 Mutual Exclusion

primrec (**in** *classical-logic*) *exclusive* :: *'a list* $\Rightarrow$ *'a* ($\coprod$)
where
$\coprod\ [] = \top$
$|\ \coprod\ (\varphi\ \#\ \Phi) = \sim (\varphi \sqcap \bigsqcup\ \Phi) \sqcap \coprod\ \Phi$

5.7 Subtraction

definition (**in** *classical-logic*) *subtraction* :: *'a* $\Rightarrow$ *'a* $\Rightarrow$ *'a* (**infixl** $\setminus$ *69*)
where $\varphi \setminus \psi = \varphi \sqcap \sim \psi$

lemma (**in** *classical-logic*) *subtraction-embedding* [*simp*]:
$(\!|\ \varphi \setminus \psi\ |\!) = (\!|\ \varphi\ |\!) \setminus (\!|\ \psi\ |\!)$
$\langle proof \rangle$

5.8 Negated Lists

definition (**in** *classical-logic*) *map-negation* :: *'a list* $\Rightarrow$ *'a list* ($\boldsymbol{\sim}$)
where [*simp*]: $\boldsymbol{\sim}\ \Phi \equiv map \sim \Phi$

5.9 Common (& Uncommon) Identities

5.9.1 Biconditional Equivalence Relation

lemma (**in** *classical-logic*) *biconditional-reflection*:
$\vdash \varphi \leftrightarrow \varphi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *biconditional-symmetry*:
$\vdash (\varphi \leftrightarrow \psi) \leftrightarrow (\psi \leftrightarrow \varphi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *biconditional-symmetry-rule*:
$\vdash \varphi \leftrightarrow \psi \Longrightarrow \vdash \psi \leftrightarrow \varphi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *biconditional-transitivity*:
$\vdash (\varphi \leftrightarrow \psi) \to (\psi \leftrightarrow \chi) \to (\varphi \leftrightarrow \chi)$

$\langle proof \rangle$

lemma (**in** *classical-logic*) *biconditional-transitivity-rule*:
$\vdash \varphi \leftrightarrow \psi \Longrightarrow \vdash \psi \leftrightarrow \chi \Longrightarrow \vdash \varphi \leftrightarrow \chi$
$\langle proof \rangle$

5.9.2 Biconditional Weakening

lemma (**in** *classical-logic*) *biconditional-weaken*:
assumes $\Gamma \Vdash \varphi \leftrightarrow \psi$
shows $\Gamma \Vdash \varphi = \Gamma \Vdash \psi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *list-biconditional-weaken*:
assumes $\Gamma :\vdash \varphi \leftrightarrow \psi$
shows $\Gamma :\vdash \varphi = \Gamma :\vdash \psi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *weak-biconditional-weaken*:
assumes $\vdash \varphi \leftrightarrow \psi$
shows $\vdash \varphi = \vdash \psi$
$\langle proof \rangle$

5.9.3 Conjunction Identities

lemma (**in** *classical-logic*) *conjunction-negation-identity*:
$\vdash \sim (\varphi \sqcap \psi) \leftrightarrow (\varphi \rightarrow \psi \rightarrow \bot)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *conjunction-set-deduction-equivalence* [*simp*]:
$\Gamma \Vdash \varphi \sqcap \psi = (\Gamma \Vdash \varphi \wedge \Gamma \Vdash \psi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *conjunction-list-deduction-equivalence* [*simp*]:
$\Gamma :\vdash \varphi \sqcap \psi = (\Gamma :\vdash \varphi \wedge \Gamma :\vdash \psi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *weak-conjunction-deduction-equivalence* [*simp*]:
$\vdash \varphi \sqcap \psi = (\vdash \varphi \wedge \vdash \psi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *conjunction-set-deduction-arbitrary-equivalence* [*simp*]:
$\Gamma \Vdash \bigsqcap \Phi = (\forall \ \varphi \in set\ \Phi.\ \Gamma \Vdash \varphi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *conjunction-list-deduction-arbitrary-equivalence* [*simp*]:
$\Gamma :\vdash \bigsqcap \Phi = (\forall \ \varphi \in set\ \Phi.\ \Gamma :\vdash \varphi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *weak-conjunction-deduction-arbitrary-equivalence* [*simp*]:

$\vdash \bigsqcap \Phi = (\forall \ \varphi \in set \ \Phi. \vdash \varphi)$
⟨*proof*⟩

lemma (**in** *classical-logic*) *conjunction-commutativity*:
$\vdash (\psi \sqcap \varphi) \leftrightarrow (\varphi \sqcap \psi)$
⟨*proof*⟩

lemma (**in** *classical-logic*) *conjunction-associativity*:
$\vdash ((\varphi \sqcap \psi) \sqcap \chi) \leftrightarrow (\varphi \sqcap (\psi \sqcap \chi))$
⟨*proof*⟩

lemma (**in** *classical-logic*) *arbitrary-conjunction-antitone*:
$set \ \Phi \subseteq set \ \Psi \Longrightarrow \vdash \bigsqcap \Psi \to \bigsqcap \Phi$
⟨*proof*⟩

lemma (**in** *classical-logic*) *arbitrary-conjunction-remdups*:
$\vdash (\bigsqcap \Phi) \leftrightarrow \bigsqcap (remdups \ \Phi)$
⟨*proof*⟩

lemma (**in** *classical-logic*) *curry-uncurry*:
$\vdash (\varphi \to \psi \to \chi) \leftrightarrow ((\varphi \sqcap \psi) \to \chi)$
⟨*proof*⟩

lemma (**in** *classical-logic*) *list-curry-uncurry*:
$\vdash (\Phi :\to \chi) \leftrightarrow (\bigsqcap \Phi \to \chi)$
⟨*proof*⟩

5.9.4 Disjunction Identities

lemma (**in** *classical-logic*) *bivalence*:
$\vdash \sim \varphi \sqcup \varphi$
⟨*proof*⟩

lemma (**in** *classical-logic*) *implication-equivalence*:
$\vdash (\sim \varphi \sqcup \psi) \leftrightarrow (\varphi \to \psi)$
⟨*proof*⟩

lemma (**in** *classical-logic*) *disjunction-commutativity*:
$\vdash (\psi \sqcup \varphi) \leftrightarrow (\varphi \sqcup \psi)$
⟨*proof*⟩

lemma (**in** *classical-logic*) *disjunction-associativity*:
$\vdash ((\varphi \sqcup \psi) \sqcup \chi) \leftrightarrow (\varphi \sqcup (\psi \sqcup \chi))$
⟨*proof*⟩

lemma (**in** *classical-logic*) *arbitrary-disjunction-monotone*:
$set \ \Phi \subseteq set \ \Psi \Longrightarrow \vdash \bigsqcup \Phi \to \bigsqcup \Psi$
⟨*proof*⟩

lemma (**in** *classical-logic*) *arbitrary-disjunction-remdups*:
$\vdash (\bigsqcup\ \Phi) \leftrightarrow \bigsqcup\ (remdups\ \Phi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *arbitrary-disjunction-exclusion-MCS*:
assumes $MCS\ \Omega$
shows $\bigsqcup\ \Psi \notin \Omega \equiv \forall\ \psi \in set\ \Psi.\ \psi \notin \Omega$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *contra-list-curry-uncurry*:
$\vdash (\Phi :\rightarrow \chi) \leftrightarrow (\sim \chi \rightarrow \bigsqcup\ (\sim \Phi))$
$\langle proof \rangle$

5.9.5 Monotony of Conjunction and Disjunction

lemma (**in** *classical-logic*) *conjunction-monotonic-identity*:
$\vdash (\varphi \rightarrow \psi) \rightarrow (\varphi \sqcap \chi) \rightarrow (\psi \sqcap \chi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *conjunction-monotonic*:
assumes $\vdash \varphi \rightarrow \psi$
shows $\vdash (\varphi \sqcap \chi) \rightarrow (\psi \sqcap \chi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *disjunction-monotonic-identity*:
$\vdash (\varphi \rightarrow \psi) \rightarrow (\varphi \sqcup \chi) \rightarrow (\psi \sqcup \chi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *disjunction-monotonic*:
assumes $\vdash \varphi \rightarrow \psi$
shows $\vdash (\varphi \sqcup \chi) \rightarrow (\psi \sqcup \chi)$
$\langle proof \rangle$

5.9.6 Distribution Identities

lemma (**in** *classical-logic*) *conjunction-distribution*:
$\vdash ((\psi \sqcup \chi) \sqcap \varphi) \leftrightarrow ((\psi \sqcap \varphi) \sqcup (\chi \sqcap \varphi))$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *subtraction-distribution*:
$\vdash ((\psi \sqcup \chi) \setminus \varphi) \leftrightarrow ((\psi \setminus \varphi) \sqcup (\chi \setminus \varphi))$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *conjunction-arbitrary-distribution*:
$\vdash (\bigsqcup\ \Psi \sqcap \varphi) \leftrightarrow \bigsqcup\ [\psi \sqcap \varphi.\ \psi \leftarrow \Psi]$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *subtraction-arbitrary-distribution*:
$\vdash (\bigsqcup\ \Psi \setminus \varphi) \leftrightarrow \bigsqcup\ [\psi \setminus \varphi.\ \psi \leftarrow \Psi]$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *disjunction-distribution*:
$\vdash (\varphi \sqcup (\psi \sqcap \chi)) \leftrightarrow ((\varphi \sqcup \psi) \sqcap (\varphi \sqcup \chi))$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *implication-distribution*:
$\vdash (\varphi \to (\psi \sqcap \chi)) \leftrightarrow ((\varphi \to \psi) \sqcap (\varphi \to \chi))$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *list-implication-distribution*:
$\vdash (\Phi :\to (\psi \sqcap \chi)) \leftrightarrow ((\Phi :\to \psi) \sqcap (\Phi :\to \chi))$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *biconditional-conjunction-weaken*:
$\vdash (\alpha \leftrightarrow \beta) \to ((\gamma \sqcap \alpha) \leftrightarrow (\gamma \sqcap \beta))$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *biconditional-conjunction-weaken-rule*:
$\vdash (\alpha \leftrightarrow \beta) \Longrightarrow \vdash (\gamma \sqcap \alpha) \leftrightarrow (\gamma \sqcap \beta)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *disjunction-arbitrary-distribution*:
$\vdash (\varphi \sqcup \prod \Psi) \leftrightarrow \prod [\varphi \sqcup \psi.\ \psi \leftarrow \Psi]$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *list-implication-arbitrary-distribution*:
$\vdash (\Phi :\to \prod \Psi) \leftrightarrow \prod [\Phi :\to \psi.\ \psi \leftarrow \Psi]$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *implication-arbitrary-distribution*:
$\vdash (\varphi \to \prod \Psi) \leftrightarrow \prod [\varphi \to \psi.\ \psi \leftarrow \Psi]$
$\langle proof \rangle$

5.9.7 Negation

lemma (**in** *classical-logic*) *double-negation-biconditional*:
$\vdash \sim (\sim \varphi) \leftrightarrow \varphi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *double-negation-elimination* [*simp*]:
$\Gamma \Vdash \sim (\sim \varphi) = \Gamma \Vdash \varphi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *alt-double-negation-elimination* [*simp*]:
$\Gamma \Vdash (\varphi \to \bot) \to \bot \equiv \Gamma \Vdash \varphi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *base-double-negation-elimination* [*simp*]:
$\vdash \sim (\sim \varphi) = \vdash \varphi$

$\langle proof \rangle$

lemma (**in** *classical-logic*) *alt-base-double-negation-elimination* [*simp*]:
$\vdash (\varphi \to \bot) \to \bot \equiv \vdash \varphi$
$\langle proof \rangle$

5.9.8 Mutual Exclusion Identities

lemma (**in** *classical-logic*) *exclusion-contrapositive-equivalence*:
$\vdash (\varphi \to \gamma) \leftrightarrow \sim (\varphi \sqcap \sim \gamma)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *disjuction-exclusion-equivalence*:
$\Gamma \Vdash \sim (\psi \sqcap \bigsqcup \Phi) \equiv \forall \varphi \in set\ \Phi.\ \Gamma \Vdash \sim (\psi \sqcap \varphi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *exclusive-elimination1*:
assumes $\Gamma \Vdash \coprod \Phi$
shows $\forall \varphi \in set\ \Phi.\ \forall \psi \in set\ \Phi.\ (\varphi \neq \psi) \longrightarrow \Gamma \Vdash \sim (\varphi \sqcap \psi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *exclusive-elimination2*:
assumes $\Gamma \Vdash \coprod \Phi$
shows $\forall \varphi \in duplicates\ \Phi.\ \Gamma \Vdash \sim \varphi$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *exclusive-equivalence*:
$\Gamma \Vdash \coprod \Phi =$
$((\forall \varphi{\in}duplicates\ \Phi.\ \Gamma \Vdash \sim \varphi) \wedge$
$(\forall \varphi \in set\ \Phi.\ \forall \psi \in set\ \Phi.\ (\varphi \neq \psi) \longrightarrow \Gamma \Vdash \sim (\varphi \sqcap \psi)))$
$\langle proof \rangle$

5.9.9 Miscellaneous Disjunctive Normal Form Identities

lemma (**in** *classical-logic*) *map-negation-list-implication*:
$\vdash ((\sim \Phi) :\to (\sim \varphi)) \leftrightarrow (\varphi \to \bigsqcup \Phi)$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *conj-dnf-distribute*:
$\vdash \bigsqcup (map\ (\bigsqcap \circ (\lambda\ \varphi s.\ \varphi \# \varphi s))\ \Lambda) \leftrightarrow (\varphi \sqcap \bigsqcup (map \bigsqcap \Lambda))$
$\langle proof \rangle$

lemma (**in** *classical-logic*) *append-dnf-distribute*:
$\vdash \bigsqcup (map\ (\bigsqcap \circ (\lambda\ \Psi.\ \Phi\ @\ \Psi))\ \Lambda) \leftrightarrow (\bigsqcap \Phi \sqcap \bigsqcup (map \bigsqcap \Lambda))$
$\langle proof \rangle$

notation *FuncSet.funcset* (**infixr** $\to$ *60*)

end

Bibliography

[1] P. Blackburn, M. de Rijke, and Y. Venema. Section 4.2 Canonical Models. In *Modal Logic*, pages 196–201.

[2] A. Bobenrieth. The Origins of the Use of the Argument of Trivialization in the Twentieth Century. 31(2):111–121.

[3] G. Boolos. Don't Eliminate Cut. 13(4):373–378.

[4] G. Gentzen. Untersuchungen über das logische schließen. i. 39(1):176–210.

[5] C. S. Peirce. On the Algebra of Logic: A Contribution to the Philosophy of Notation. 7(2):180–196.

[6] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Number 43 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2nd ed edition.

[7] A. Urquhart. Implicational Formulas in Intuitionistic Logic. 39(4):661–664.

[8] D. van Dalen. *Logic and Structure*. Universitext. Springer-Verlag, 5 edition.