Introduction to Proof Theory

Lecturers: Marianna Girlando, Sonia Marin
Scribes: Yi Cai, Nathan Guermond, Abhiram Kothapalli,
Selene Linares Arévalo, Youning Xia

Oregon Programming Languages Summer School (OPLSS) 2023

Contents

1 Introduction

We can talk about proofs at three different levels, the social level, the object level, and the meta level. At the social level, proofs are *informal* arguments, or evidence, of the truth of an informal statement. The role of proofs at the social level is to communicate arguments. At the object level, proofs are *formal* arguments of the truth of formal statements. The role of proofs at the object level is to have a systematic way of checking the validity of proofs. At the meta level, proofs are treated as *mathematical objects of study*, and we wish to reason about the properties that proofs and proof systems may have. For example, we may wish to say that a proof system is sound, complete, etc.

A brief timeline of the development of proof theory is as follows:

1879 Frege - the structure of proofs should be formalized as objects (object level)

1890 Peano - arithmetic should be formalized (object level)

1906 Russel & Whitehead - *Principia Mathematica*

1928 Hilbert's Program - proofs systems should be studied and reasoned about as mathematical objects:

- (a) Are proof systems consistent (ie. false is not derivable)?
- (b) Can proof systems be complete (ie. is every formula either provable or disprovable)?
- (c) Is provability decidable (ie. algorithmic proof procedure)?

1928 Hilbert & Ackermann - classical predicate logic

1929 Gödel's completeness theorem (syntactic provability corresponds to semantic provability)

1930 Gödel's incompleteness theorem (proof systems for arithmetic cannot prove or disprove any statement in the language)

1935 Gentzen - consistency of arithmetic (cut elimination)

1936 Church - undecidability of classical predicate logic

In these notes we will attempt to describe logic completely syntactically, and without reference to semantics.

2 Propositional Logic $\mathcal{L}_p$

Propositional logic is typically what comes to mind when we think of formal logic. Propositional logic consists of basic statements such as *A and B*, *A or B*, and *A implies B*. as well as more complex statements that can be stitched together from these basic statements such as

A and B implies B,

and

$$A \text{ implies } B \text{ implies that not } B \text{ implies not } A.$$

Unfortunately, expressing such statements in plain English introduces various ambiguities. Our goal in this section is to define a mathematical syntax for unambiguously expressing such statements.

Of course, our formal syntax will also be able to express false statements such as

$$A \text{ or } B \text{ implies } B.$$

To use our syntax in a meaningful way, we would like to define a set of *axioms* which we semantically know to be true (for example, *A and B implies B*), and a set of *inference rules* for what statements we can derive from a set of true statements (for example, *Given A and A implies B we have B*), and use this to decide whether a statement is true or false. We study a natural set of axioms and inference rules for propositional logic in Section 4.

Syntax We start by defining the syntax of propositional logic $\mathcal{L}_p$. The basic terms of the language are defined as follows.

1. **Propositional Variables:** A countable set of variables

$$\mathsf{Var}_p = \{p, q, r, \ldots\}.$$

2. **Propositional Constants:** $\bot$ which represents false.

3. **Connectives:** $\lor$ (disjunction), $\land$ (conjunction), $\rightarrow$ (implication)

Using these basic terms, we can inductively define the set of formulas Form_p as follows.

- Propositional variables and constants are formulas.
- If A and B are formulas then $A \lor B$ (read A or B), $A \land B$ (read A and B), and $A \rightarrow B$ (read A implies B) are formulas.

Now, we have a formal language for expressing our previous statements. For example, to formally state *A and B implies B* we write

$$A \land B \rightarrow B.$$

Encoding Negation and True Instead of introducing additional terms, we can encode negation and true using the language we have built so far. In particular, we can define negation as

$$\neg A := A \rightarrow \mathsf{false}.$$

Intuitively, if $\neg A$ is true, (that is, A is false) we have that A vacuously implies anything, including false. As an example, to formally state *A implies B implies that not B implies not A*, we write

$$(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A).$$

Similarly, we can define true as

$$\top := p \vee \neg p$$

where p is a propositional variable. Intuitively, regardless of the assignment of p we have that either p is true or false (classically that is).

In fact, we can even design an even more minimal language by encoding $\vee$ and $\wedge$ using just $\rightarrow$ and $\bot$. However, this is beyond the scope of this lecture.

Interpreting Propositional Formulas Thus far, we have defined a language that is populated with basic terms and the formulas that inductively arise from these terms. We would like to semantically interpret these formulas in a meaningful way. Naturally, we can assign propositional variables in Var_p true (denoted with 1) or false (denoted with 0). We can represent this assignment as a function α of type $\mathsf{Var}_p \rightarrow \{0, 1\}$ Given an assignment for propositional variables, we can inductively derive meaningful assignments for formulas as follows. Given assignments for propositional variables A and B we derive an assignment for $A \wedge B$ according to the following table.

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

Likewise, we derive an assignment for $A \vee B$ according to the following table.

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

Finally, we derive an assignment for $A \rightarrow B$ according to the following table.

A	B	$A \rightarrow B$
1	1	1
1	0	0
0	1	1
0	0	1

We say an assignment α satisfies a formula A (denoted $\alpha \models A$) if the the formula A is assigned 1 according to the above rules. Equivalently, we can compute an assignment for a formula A according to α inductively as follows.

$$\begin{array}{lll}
\alpha \not\models \bot & & \\
\alpha \models p & \text{iff} & \alpha(p) = 1 \\
\alpha \models A \wedge B & \text{iff} & \alpha \models A \text{ and } \alpha \models B \\
\alpha \models A \vee B & \text{iff} & \alpha \models A \text{ or } \alpha \models B \\
\alpha \models A \rightarrow B & \text{iff} & \alpha \not\models A \text{ or } \alpha \models B
\end{array}$$

3 Predicate Logic

Predicate logic enhance the expressiveness of propositional logic by adding functions, predicates and quantifiers, as well as rules to recursively generate the sets of terms and formulas. We define the syntax of predicate language $\mathcal{L}^=$ as follows.

Symbols

- Variables: $\mathsf{Var} = \{x, y, z, \ldots\}$
- Propositional Variables: $\mathsf{PropVar} = \{p, q, r, \ldots\}$
- Function Symbols: $\mathsf{Fun} = \{f, g, h, \ldots\}$
  - each function symbol has a fixed number of arguments it takes (in other words, a fixed *arity*). For example, $\mathsf{succ}(x)$ is a function symbol with arity 1.
  - *Constants* are special function symbols that take zero arguments. For example, zero is a constant.
- Predicate Symbols: $\mathsf{Pred} = \{P, Q, R, \ldots\}$
  - each predicate symbol has a fixed arity. For example, $\mathsf{Even}(x)$, $\mathsf{Odd}(y)$ are predicate symbols with arity 1.
- Equality Symbol: =
  - equality symbol is in effect predicate symbol but since it has an important role in our language, we define it separately.
- Connectives: $\vee, \wedge, \rightarrow, \bot$
- Quantifiers: $\exists$ (existential), $\forall$ (universal)

Formation Rules

- Terms are used to represent individual entities, i.e elements in the domain we are working with and they are denoted by $s, t, u, ...$
  The set of terms TERMS is inductively generated as follows:
  * Variables are terms.
  * If $f \in$ FUN is a k-ary function symbol and $t_1, t_2, ...t_k$ are terms, then $f(t_1, t_2, ...t_k)$ is also a term.

  Examples of terms are: z, f, $f(z, z)$, and $g(f(z, z), x)$, where
- Atomic formulas are generated as follows:
  * If $P \in$ Pred is a k-ary predicate symbol and $t_1, t_2, ...t_k$ are terms, then $P(t_1, t_2, ...t_k)$ is an atomic formula.

* If s and t are terms then $s = t$ is an atomic formula.
  Note that this is a particular case of the first rule where P corresponds to $=$ (the equality symbol), whose arity is 2.

Predicates take one or more terms and decide if the terms satisfy a property, in other words, $P : TERM^k \to Bool$.
Examples of atomic formulas are: $P(x)$, $f(z) = f(g(y))$ and $R(f(x), y, z, w)$.

- The set of formulas, FORM, is inductively generated as follows:
  * Atomic formulas are formulas.
  * $\bot$ is a formula.
  * If A and B are formulas then $A \vee B, A \wedge B$ and $, A \to B$ are formulas.
  * If A is a formula and x is a variable then $\exists x A$ and $\forall x A$ is a formula.

4 Proof Systems

4.1 Proof Systems, informally

A **proof system** consists of:

- Set of axioms (formulas of the language)
- Set of inference rules. (how to compose formulas of the language)

A **proof** of a formula A is constructed by chaining together axioms, inference rules, and *objects* (intermediate steps) generated from axioms and inference rules, until A is reached.

A **logic** can be *identified* with the set of provable formulas.

4.2 Various kinds of proof systems

- Hilbert-Frege proof systems, or axiom systems, or reductive systems (Prawitz, 1971)
- Gentzen-style proof systems (our focus)

5 Hilbert-Frege system for classical propositional logic $\mathcal{H}_{cp}$

Let A, B and C be formulas of $\mathcal{L}_p$, then the following are axiom *schemas* for the logic. This means we have an instance of each axiom for each instance of A, B, and C.

PL1. $A \to (B \to A)$
PL2. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
PL3. $(A \land B) \to A$
PL4. $(A \land B) \to B$
PL5. $A \to (B \to (A \land B))$
PL6. $A \to (A \lor B)$
PL7. $B \to (A \lor B)$
PL8. $(A \to C) \to ((B \to C) \to ((A \lor B) \to C))$
PL9. $\bot \to A$
PL10. $A \lor (A \to \bot)$

One inference rule for implication elimination (modus ponens):

$$\text{MP}\ \frac{A \qquad A \to B}{B}$$

Note that PL1.−PL2. are rules for implication, PL3.−PL5. are rules for conjunction, PL6. − PL8. are rules for disjunction, PL9. is the principal of explosion, and PL10. is the law of excluded middle.

5.1 Proofs in $\mathcal{H}_{cp}$

Given a formula A in $\mathcal{L}_p$, and Γ a set of formulas of $\mathcal{L}_p$, an $\mathcal{H}_{cp}$ *derivation* of A from assumptions Γ is a list of $\mathcal{L}_p$ formulas

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}$$

such that $A_n = A$ and for each A_i we have either

1. A_i is an axiom of $\mathcal{H}_{cp}$
2. $A_i \in \Gamma$
3. A_i is obtained by applying mp to formulas in $A_1 \dots A_{i-1}$.

We write $\Gamma \vdash_{\mathcal{H}\ cp} A$ if there is a derivation of A from formulas in Γ. A *proof* of A is a derivation of A from the empty set of premises $\emptyset$. We write $\vdash_{\mathcal{H}_{cp}} A$ if there is a proof of A. The classical propositional logic

CPL

is defined as $\{A \mid\ \vdash_{\mathcal{H}\ cp} A\}$.

Remark 1. *Note that a derivation in $\mathcal{H}_{cp}$ can also be viewed as a tree. However, the list representation is more compact since hypotheses may be reused.*

5.1.1 Examples

Example 1. $\vdash_{\mathcal{H}_{cp}} A \to A$

1. $A \to ((B \to A) \to A)$ *(*PL1.*)*
2. $(A \to ((B \to A) \to A)) \to (A \to ((B \to A) \to (A \to A)))$ *(*PL2.*)*
3. $(A \to (B \to A)) \to (A \to A))$ *(*mp *1, 2)*
4. $A \to (B \to A)$ *(*PL1.*)*
5. $A \to A$ *(*mp *3, 4)*

Example 2 (Exercise)**.** $\vdash_{\mathcal{H}_{cp}} (A \to B) \to (B \to C) \to (A \to C)$

5.2 Deduction Theorem

Theorem 5.1 (Deduction)**.** *For a formula A and a set of hypotheses* Γ *of* $\mathcal{L}_p$*,*

$$\Gamma \vdash_{\mathcal{H}_{cp}} A \to B \quad \textit{iff} \quad \Gamma \cup \{A\} \vdash_{\mathcal{H}_{cp}} B$$

This means to prove an implication $A \to B$, it is sufficient to assume A as a hypothesis and then proceed to derive B, and it is also true the other way around. This means the following two derivations are equivalent.

$$\begin{array}{ccc} \Gamma & & \Gamma \cup \{A\} \\ \vdots & \Longleftrightarrow & \vdots \\ A \to B & & B \end{array}$$

6 Peano Arithmetic

6.1 First-order logic $\mathcal{H}_{fo}$

In section 2, we have introduced propositional language $\mathcal{L}_p$ that can be used to describe facts like "lemon is a fruit", i.e. $\mathsf{Fruit}(\mathit{lemon})$. However, it is not powerful, *expressive* enough to describe a complicated world in a concise way. For example, in a world (domain) where we describe family relationships, we want to express a parent of one's parent (a.k.a one's grandparent). Using propositional logic, we can describe this knowledge as

$$\exists p, \mathsf{Parent}(g, p) \land \mathsf{Parent}(p, c)$$

where g is p's grandparent. This can be more concisely expressed using first-order logic as

$$\forall g, c, \mathsf{Grandparent}(g, c)$$

Intuitively, being able to express complex knowledge more concisely serves as our motivation to expand propositional logic to first-order logic. Now, let us formally define the axiom system of first-order logic $\mathcal{H}_{fo}$.

Axioms Axioms of first-order logic are defined as follows

1. $\forall x(A(x)) \to A(t)$
2. $A(t) \to \exists x(A(x))$
3. $\forall x(A \to B(x)) \to (A \to \forall x(B(x)))$ where $x \notin FV(A)$
4. $\forall x(A(x) \to B) \to (\exists x(A(x)) \to B)$ where $x \notin FV(B)$
5. $\forall x(x = x)$
6. $\forall x \forall y(x = x \to (A(x) \to A(y)))$

Inference Rules Inference rule of first-order logic is defined as

$$\mathsf{gen}\ \frac{\forall x(A)}{A}$$

6.2 Proofs in $\mathcal{H}_{fo}$

We denote the A as formula, and Γ as a set of formula. (We have defined formula when we talked about predicate logic in previous section)

We define *derivation* of A from Γ as a list of formula,

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}$$

where $A_n = A$ and for each A_i, for $i \leq n$, we have that either:

- A_i is an axiom of $\mathcal{H}_{fo}$;
- $A_i \in \Gamma$;
- A_i is obtained by applying (mp) or (gen) to $A_1, \ldots, A_{i-1}$.

A proof of A is a derivation of A from $\emptyset$. We write $\Gamma \vdash_{\mathcal{H}_{fo}} A$ if there is a proof of A.

6.3 Peano Arithmetic

Perhaps numbers are the most vivid example of a first-order logic where we start from a small set of axioms and build up the large theory of natural numbers. These axioms are known as Peano axioms that define natural numbers and addition.

More formally,

7 Intuitionistic Logic

7.1 Constructive proofs

Motivation for constructive logic. Let's consider the following theorem:

Theorem 7.1. *There exists irrational numbers a and b such that a^b is irrational.*

Proof 1. Let's assume that $\sqrt{2}^2$ is either rational or irrational by the axiom of excluded middle.

- *Case $\sqrt{2}^2$ is a rational number:* let $a = b = \sqrt{2}$, then a and b are irrational numbers such that $a^b = \sqrt{2}^2$ is rational and the statement is proved.
- *Case $\sqrt{2}^2$ is an irrational number:* Let $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$.
  Then $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = 2$, which is rational and the statement is proved.

□

Note that the above is a non-constructive proof because it assumes that $\sqrt{2}^2$ is either rational or irrational, but it does not say which case holds true.

Proof 2. Take $a = \sqrt{2}$ and $b = log_2 9$. Then $a^b = 3$, which is rational. □

On the other hand, *Proof 2* is a constructive proof because it provides evidence of two elements a and b satisfying that a^b is irrational.

7.2 An axiom system for intuitionistic propositional logic: $\mathcal{H}_{ip}$

The syntax used in the axiom system for intuitionistic propositional logic $\mathcal{H}_{ip}$ is the same syntax from $\mathcal{L}_p$. Additionally, $\mathcal{H}_{ip}$ consist of the following axioms:

$$
\begin{array}{ll}
PL1 & B \to (A \to B) \\
PL2 & (A \to (B \to C)) \to ((A \to B) \to (A \to C)) \\
PL3 & (A \land B) \to A \\
PL4 & (A \land B) \to B \\
PL5 & A \to (B \to (A \land B)) \\
PL6 & A \to (A \lor B) \\
PL7 & B \to (A \lor B) \\
PL8 & (A \to C) \to ((B \to C) \to ((A \lor B) \to C)) \\
PL9 & \bot \to A
\end{array}
$$

On top of that, $\mathcal{H}_{ip}$ has one single inference rule, namely *modus ponens*:

$$\frac{A \quad A \to B}{B}\ \text{MP}$$

which can be read as if we have a proof of A and a proof of $A \to B$, then we can conclude that we have a proof of B.

7.3 Proofs in $\mathcal{H}_{ip}$

Definition 1. *Let A be a formula in $\mathcal{L}_p$, Γ a set of formulas in $\mathcal{L}_p$. A $\mathcal{H}_{ip}$ derivation of A from assumptions in Γ is a list of $\mathcal{L}_p$ formulas*

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}$$

where $A_1 = A$ and for each A_i such that $i \leq n$, we have that either:

- *A_i is an axiom of $\mathcal{H}_{ip}$*
- *$A - i \in \Gamma$*
- *A_i is obtained by applying (mp) to formulas in $A_1, A_2, \ldots, A_{n-1}$.*

We write $\Gamma \vdash_{\mathcal{H}_{ip}} A$ if there is a derivation of A from formulas in Γ.

Definition 2. *A proof of A is a derivation of A from $\emptyset$, denoted by*

$$\vdash_{\mathcal{H}_{ip}} A$$

.

Definition 3. *Intuitionistic propositional logic IPL is define as all the propositional formulas A such that there is a proof of A, i.e, there is a derivation of A from $\emptyset$; in other words:*

$$IPL = \{A \mid \vdash_{\mathcal{H}_{ip}} A\}$$

7.3.1 Examples

Examples of formulas that are provable in $\mathcal{H}_{ip}$:

- $(\neg A \vee B) \to A \to B$
- $(A \vee B) \to \neg(\neg A \wedge \neg B)$

Examples of formulas that are not provable in $\mathcal{H}_{ip}$:

- $A \vee \neg A$ (excluded middle)
- $\neg\neg A \to A$ (double negation)
- $((A \to B) \to A) \to A$ (Pierce's law)
- $(A \to B) \to (\neg A \vee B)$

8 Intuitionistic Natural Deduction $\mathcal{NJ}$

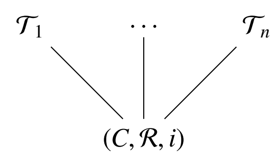
A pre-derivation in natural deduction ($\mathcal{NJ}$) is a tree with nodes labelled with formulas in $\mathcal{L}_p$ as well as a deduction rule (see below). A leaf is a formula which is tagged as either open (live) or closed (dead/discharged) with an index i. For a set of hypotheses Γ, a *derivation* of C from Γ is a well-formed pre-derivation with root C (defined below) whose set of open leaves all lie in Γ. Given a pre-derivation $\mathcal{T}$, denote by $\mathcal{T}[i]$ the same tree where the closed leaves tagged with index i are now marked as open.

A deduction rule $\mathcal{R}$ is an n-ary constructor of the form

$$\dfrac{A_1 \quad \dots \quad A_k \quad \begin{array}{c}[B_1]_i\\ \vdots \\ A_{k+1}\end{array} \quad \dots \quad \begin{array}{c}[B_{n-k}]_i\\ \vdots \\ A_n\end{array}}{C}\ \mathcal{R}^i$$

A leaf A is a well formed derivation of A from hypotheses $\{A\} \cup \Gamma$ if it is open.

A pre-derivation of C from assumptions Γ with root node labelled with formula C, rule $\mathcal{R}$, and index i seen as follows,



is *well-formed* if for each child derivation $\mathcal{T}_j$, then $\mathcal{T}_j$ is a derivation of A_j from hypotheses Γ when $j \leq k$, and $\mathcal{T}_j[i]$ is a derivation of A_j from hypotheses Γ, B_{j-k} when $j > k$.

The rules are categorized either as *introduction* or *elimination* rules. Namely, an introduction rule specifies how to construct a proof of a formula, and an elimination rule specifies how to use a proof of this formula to prove another formula.

The rules for $\mathcal{NJ}$ are as follows:

Introduction rules

$$\dfrac{A}{A \vee B}\ \vee\mathcal{I}1 \qquad \dfrac{B}{A \vee B}\ \vee\mathcal{I}2 \qquad \dfrac{A \quad B}{A \wedge B}\ \wedge\mathcal{I} \qquad \dfrac{\begin{array}{c}[A]_i\\ \vdots \\ B\end{array}}{A \to B}\ \to\mathcal{I}^i$$

Elimination rules

$$\frac{A \wedge B}{A}\wedge\mathcal{E}1 \qquad \frac{A \wedge B}{B}\wedge\mathcal{E}2 \qquad \frac{A \vee B \quad \begin{matrix}[A]_i \\ \vdots \\ C\end{matrix} \quad \begin{matrix}[B]_i \\ \vdots \\ C\end{matrix}}{C}\vee\mathcal{E}^i$$

$$\frac{A \to B \qquad A}{B}\to\mathcal{E}$$

As can be remarked from this definition, the property of a leaf being open or closed is not a static property of the proof since it changes depending on the subtree we are verifying. Hence a natural deduction proof is not a genuine tree. One may wonder why not present the natural deduction in the style of the sequent calculus with judgements of the form $\Gamma \vdash A$. However, the proofs in this calculus are not one to one with proofs in natural deduction. Consider for example the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A, A \vdash A}\,?}{A \vdash A \to A}\to\mathcal{I} \qquad A \vdash A}{A \vdash A}\to\mathcal{E}}{\vdash A \to A}\to\mathcal{I}$$

This proof corresponds to two different natural deduction proofs, since we have two ways to prove $A, A \vdash A$. Contrast this with the $\mathcal{NJ}$ proof

$$\dfrac{\dfrac{\dfrac{[A]_2}{A \to A}\to\mathcal{I}^2 \qquad [A]_1}{A}\to\mathcal{E}}{A \to A}\to\mathcal{I}^1$$

where this ambiguity is resolved.

8.1 Completeness and Soundness

We can now show that $\mathcal{NJ}$ is relatively sound and complete with respect to $\mathcal{H}_{ip}$. That is, the completeness theorem says that if $\mathcal{H}_{ip}$ is complete then $\mathcal{NJ}$ is complete, and the soundness theorem says that if $\mathcal{H}_{ip}$ is sound then $\mathcal{NJ}$ is sound (recall that a proof system is sound if $\bot$ is not derivable).

Theorem 8.1 (Completeness of $\mathcal{NJ}$)**.** *If* $\vdash_{\mathcal{H}_{ip}} A$ *then* $\vdash_{\mathcal{NJ}} A$.

Proof. By induction on the structure of the derivation of $\vdash_{\mathcal{H}_{ip}} A$, seen as a tree (see remark 1).

Base case: We must show that each of the axioms PL. 1-9 are derivable in $\mathcal{NJ}$ (exercise).

Inductive step: The only inductive case is the mp rule, that is if we have $\vdash_{\mathcal{NJ}} A \to B$ and $\vdash_{\mathcal{NJ}} A$, then to derive $\vdash_{\mathcal{NJ}} B$ we use the $\to \mathcal{E}$ rule. □

Theorem 8.2 (Soundness of $\mathcal{NJ}$)**.** *If* $\vdash_{\mathcal{NJ}} A$ *then* $\vdash_{\mathcal{H}_{ip}}$.

Proof. By induction on the structure of the derivation.

Base case: If $\vdash_{\mathcal{NJ}} A$ where A is an open leaf (recall this means that A belongs to the set of hypotheses), then recall from example 1 that $\vdash_{\mathcal{H}_{ip}} A \to A$, and by the deduction theorem we obtain $A \vdash_{\mathcal{H}_{ip}} A$.

Inductive step: We must consider every rule of $\mathcal{NJ}$.

$\to \mathcal{I}$ In the case that a derivation of $\vdash_{\mathcal{NJ}} A \to B$ ends with the rule $\to \mathcal{I}$, we know from the induction hypothesis that $A \vdash_{\mathcal{H}_{ip}} B$, and thus $\vdash_{\mathcal{H}_{ip}} A \to B$ follows from the deduction theorem.

The other cases are left as exercise. □

9 The Curry-Howard Correspondence

Over a series of exchanges between the 1930's and 1950's Curry and Howard together discovered a remarkable correspondence between the structure of proofs and the structure of programs. Just as programs map a set of input types to a set of output types, proofs can be viewed as mapping a set of input propositions (expressed as formulas) to a set of output propositions. The formal realization of this symmetry is known as the Curry-Howard correspondence.

To see this, we can introduce a new typing judgement

$$M : A$$

to indicate that M is a proof of A. Then, by definition, there exists M such that $M : A$ if and only if $\vdash A$, that is A is true in the proof system. Thus, under the Curry-Howard paradigm, to prove that a proposition A is true, we seek to show that type A is inhibited by some term M which represents the proof.

Of course, the type system will change based on the logical system that we use. In this section, we first introduce the simply-typed lambda calculus with *products* and *sums*. We then argue that a formula is true in $\mathcal{NJ}$ if and only if the corresponding type in this language inhibited.

SLTC with Products and Sums We define the syntax of the simply typed lambda calculus with products (i.e. pairs) and sums (i.e. disjunctions) as follows.

$$\begin{aligned} L, M, N := & x \mid \lambda x.M \mid MN \mid \\ & \langle M, N \rangle \mid \pi_1(M) \mid \pi_2(M) \mid \\ & \mathsf{in}_1(M) \mid \mathsf{in}_2(M) \mid \mathsf{case}(L, x \Rightarrow M, y \Rightarrow N) \mid \\ & \varepsilon(M) \end{aligned}$$

Semantically, $\langle M, N\rangle$ represents a pair of elements. Then, π_1 (respectively π_2) is intended to represent projecting the first element of a pair (respectively second). Similarly, in_1 (respectively in_2) is intended to represent injecting the first (respectively second) element into a disjoint pair. case is intended to represent picking different terms based on whether the input is a the first or second element of a disjunctive pair. Finally $\varepsilon(M)$ is intended to represent an error.

Proofs as λ-terms We now show how to view a proof of a formula in $\mathcal{NJ}$ as a term in the simply typed lambda calculus with products and sums. We do so inductively: In particular, we will show how to construct such terms for each of the introduction and elimination rules in $\mathcal{NJ}$.

First, given a proof M of A (respectively N of B), we can derive a proof $\mathsf{in}_1(M)$ (respectively $\mathsf{in}_2(N)$) of $A \vee B$.

$$\frac{M : A}{\mathsf{in}_1(M) : A \vee B}\ \vee_{I_1} \qquad\qquad \frac{N : B}{\mathsf{in}_2(N) : A \vee B}\ \vee_{I_2}$$

Given a proof M of A and a proof N of B, we have that the pair of proofs $\langle M, N\rangle$ represents a proof for $A \wedge B$.

$$\frac{M : A \qquad N : B}{\langle M, N\rangle : A \wedge B}\ \wedge_I$$

Suppose that we can derive a proof M of B assuming a proof x of A. Then, with lambda abstraction we are afforded a function a proof of type $A \to B$.

$$\frac{\begin{array}{c}[x : A]\\ \vdots\\ M : B\end{array}}{\lambda x.M : A \to B}\ \to_I$$

Given a proof M for $A \wedge B$ we can project out proofs for A and B respectively.

$$\frac{M : A \wedge B}{\pi_1(M) : A}\ \wedge_{E_1} \qquad\qquad \frac{A \wedge B}{\pi_2(M) : B}\ \wedge_{E_2}$$

Now, suppose we have a proof of $A \vee B$ and assuming either a proof of A or a proof of B we can construct a proof for C. Then, we have a proof for C. This is captured by the following rule.

$$\frac{L : A \vee B \qquad \begin{array}{c}[x : A]\\ \vdots\\ M : C\end{array} \qquad \begin{array}{c}[y : B]\\ \vdots\\ N : C\end{array}}{\mathsf{case}(L, x \Rightarrow M, y \Rightarrow N) : C}\ \vee_E$$

Similarly, given a proof M of $A \to B$ and a proof N of A, we obtain a proof

$$\dfrac{M : A \to B \qquad N : A}{\mathsf{app}(M, N) : B} \to_E$$

which is precisely application in STLC.

Finally, if we assume a proof term for false, then we should be able to derive proof terms for any other formula.

$$\dfrac{M : \bot}{\varepsilon(M) : A} \text{ FAIL}$$

10 Intuitionistic Sequent Calculus $\mathcal{LJ}$

Sequent calculus was defined by Getzen as a syntax to reflect the natural process of deductive reasoning. In particular, a *sequent* is of the form

$$\Gamma \Rightarrow A. \tag{1}$$

Here, Γ is a sequence of formulas in $\mathcal{NJ}$ and A is a formula in $\mathcal{NJ}$. We can formally interpret a sequent as a formula in $\mathcal{NJ}$ via function fm as follows

$$\mathsf{fm}(\emptyset \Rightarrow A) := \top \to A$$
$$\mathsf{fm}(B_1, \ldots, B_n \Rightarrow A) := (B_1, \ldots, B_n) \to A$$

We now define the inference rules of Intuitionistic Sequent Calculus. We start with two *initial* sequents, which can be viewed as the axioms of the system

$$\mathsf{init}\,\dfrac{}{p, \Gamma \Rightarrow p} \qquad \bot\,\dfrac{}{\bot, \Gamma \Rightarrow C}$$

We also require inference rules for conjunction, disjunction, and implication.

$$\wedge\mathsf{L}\,\dfrac{A, B, \Gamma \Rightarrow C}{A \wedge B, \Gamma \Rightarrow C} \qquad \wedge\mathsf{R}\,\dfrac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B}$$

$$\vee\mathsf{L}\,\dfrac{A, \Gamma \Rightarrow C \quad B, \Gamma \Rightarrow C}{A \vee B, \Gamma \Rightarrow C} \qquad \vee\mathsf{R}\,\dfrac{\Gamma \Rightarrow A_i}{\Gamma \Rightarrow A_0 \vee A_1}\ i \in \{0, 1\}$$

$$\to\mathsf{L}\,\dfrac{A \to B, \Gamma \Rightarrow A \quad B, \Gamma \Rightarrow C}{A \to B, \Gamma \Rightarrow C} \qquad \to\mathsf{R}\,\dfrac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \to B}$$

Next, we introduce a weakening rule, which states that any provable statement can always take additional (unused) hypothesis. We additionally introduce a contraction rule, which states that we only need to assume a hypothesis once to use it.

$$\text{weak}\ \frac{\Gamma \Rightarrow C}{A, \Gamma \Rightarrow C} \qquad \text{cont}\ \frac{A, A, \Gamma \Rightarrow B}{A, \Gamma \Rightarrow B}$$

Finally, we introduce the cut rule which states that if formula A is provable in some context Γ and A, Γ together prove C, then we must have that C is provable under just Γ.

$$\text{cut}\ \frac{\Gamma \Rightarrow A \quad A, \Gamma \Rightarrow C}{\Gamma \Rightarrow C}$$

A Derivation in $\mathcal{LJ}$ is a (rooted) tree where nodes are labeled by sequents such that

(1) The leaves of the tree are initial sequents,

(2) The sequents occupying intermediate nodes in the tree are obtained from the sequents occupying the nodes directly above them by means of a correct application of an inference rule, and

(3) The root of the tree is the conclusion of the derivation (the endsequent).

We say that $\Gamma \Rightarrow A$ is derivable in $\mathcal{LJ}$ if it appears as the endsequent of a derivation. We denote this with $\vdash_{\mathcal{LJ}} \Gamma \Rightarrow A$.

As an example, we derive the sequent $\Rightarrow p \to (q \to (p \wedge q))$ as follows

$$\to_R \frac{\to_R \dfrac{\wedge_R \dfrac{\text{init}\ \dfrac{}{p, q \Rightarrow p} \qquad \text{init}\ \dfrac{}{p, q \Rightarrow p}}{p, q \Rightarrow p \wedge q}}{p \Rightarrow q \to (p \wedge q)}}{\Rightarrow p \to (q \to (p \wedge q))}$$

11 Cut Elimination

One of the most complex rules in intuitionistic sequent calculus is the cut rule which, recall, is defined as follows.

$$\text{cut}\ \frac{\Gamma \Rightarrow A \quad A, \Gamma \Rightarrow C}{\Gamma \Rightarrow C}$$

The key challenge is that the cut rule is not local: We can derive a proof of A further up the deduction tree and still apply the cut rule much further below. This makes it challenging for automated proof search techniques to decide when or if the cut rule is applicable. Our goal in this section is to show that the cut rule is strictly not necessary. In particular, we overview in this section a proof for the following theorem.

Theorem 11.1 (Hauptsatz, Getzen 1935)**.** *Every theorem in $\mathcal{LJ}$ has a proof that does not use the cut rule.*

As a consequence, we have the following corollary.

Corollary 1 (Analyticity)**.** *Every theorem $\mathcal{LJ}$ has a proof that only contains subformulas of it.*

The general strategy is to keep pushing the cut upwards in the derivation tree into smaller and smaller derivations until they disappear. Of course, to ensure termination, we need a "measure" on derivations and formulas, and need to ensure that we can keep pushing the cuts into derivations of smaller measure. The actual proof is quite involved so we only intuit the proof in this section. Several proofs of cut-elimination exist in the literature, using slightly different procedures and for slightly different systems [4, 3, 2].

We begin by introducing several preliminary definitions and lemmas.

Definition 4 (Height, Degree, and Rank)**.** *The* height *of a derivation $\mathcal{D}$, denoted* $\mathsf{ht}(\mathcal{D})$ *is the length of the longest branch minus* 1*. The* level *of a cut rule is the sum of the heights of the derivations of the two premises of the cut. For example, suppose we have the following derivation with cut being used as the following rule.*

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma \Rightarrow A\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ A, \Gamma \Rightarrow C\end{array}}{\Gamma \Rightarrow C}\,\mathsf{cut}$$

In particular, we have that the endsequent of $\mathcal{D}_1$ is $\Gamma \Rightarrow A$ and the endsequent of $\mathcal{D}_2$ is $A, \Gamma \Rightarrow C$. Suppose now that $\mathsf{ht}(\mathcal{D}_1) = m$ and $\mathsf{ht}(\mathcal{D}_2) = n$. Then, we have the level of this cut is $m + n$. The degree *of a formula A, denoted* $\mathsf{deg}(A)$ *is the total number of logical connectives occurring in it. In particular we define* deg *inductively as follows*

$$\begin{aligned}
\mathsf{deg}(p) &:= 0\\
\mathsf{deg}(\bot) &:= 0\\
\mathsf{deg}(A \star B) &:= \mathsf{deg}(A) + \mathsf{deg}(B) + 1 \quad \textit{for } \star \in \{\wedge, \vee, \rightarrow\}
\end{aligned}$$

The rank *of a cut rule is the degree of a cut formula plus* 1*. The rank of a derivation tree $\mathcal{D}$, denoted* $\mathsf{rk}(\mathcal{D})$*, is the maximum degree of the cut rules occurring in $\mathcal{D}$.*

We will write
$$\Gamma \Rightarrow^m_p C$$
to mean that there is a derivation of $\Gamma \Rightarrow C$ of height at most m and rank at most p. Additionally, we introduce the following two lemmas

Lemma 1 (Closure under Weakening)**.** *If $\Gamma \Rightarrow^m_p C$ then $\Gamma', \Gamma \Rightarrow^m_p C$ for any Γ'.*

Proof (Intuition). By induction on the height m of the derivation. □

Lemma 2 (Closure under Contraction)**.** *If $A, A, \Gamma \Rightarrow^m_p C$, then $A, \Gamma \Rightarrow^m_p C$.*

Proof (Intuition). By induction on m using weakening (and invertibility). □

11.1 The Cut Elimination Theorem

Our high level plan is to convert a derivation $\mathcal{D}$ in LJ of the sequent $\Gamma \Rightarrow_p C$ to a derivation D^* of the sequent $\Gamma \Rightarrow_0 C$. That is the rank of $\mathcal{D}^*$ is 0 (and therefore necessarily does not have a cut rule). We will first show how to simulate instances of a cut, and then leverate this technique to eliminate all cuts occurring in a derivation.

Lemma 3 (Closure under Cut)**.** *If* $\Gamma \Rightarrow_0^m A$ *and* $A, \Gamma \Rightarrow_0^n C$ *and*

$$\begin{array}{c} \mathcal{D}_1 \qquad\qquad \mathcal{D}_2 \\[4pt] \text{cut}\ \dfrac{\Gamma \Rightarrow_0^m A \qquad A, \Gamma \Rightarrow_0^n C}{\Gamma \Rightarrow_1 C} \end{array}$$

Then, we can construct the following derivation $\mathcal{D}^*$

$$\begin{array}{c} \mathcal{D}^* \\[4pt] \Gamma \Rightarrow_0 C \end{array}$$

Proof (Intuition). We distinguish based on cases. Let R_1 be the last rule required to derive $\Gamma \Rightarrow_0^m A$ and R_2 be the last rule required to derive $A, \Gamma \Rightarrow_0^n C$. First, we consider the setting where R_1 is an initial sequent (symmetrically R_2 is an initial sequent). Next, we consider the setting where A is principal in both R_1 and R_2 (i.e., A is the formula being "acted upon" by R_1 and R_2). Finally, we consider the case where A is not principal in either R_1 and R_2.

Suppose now that $\mathcal{D}_1$ is initial. That is, we have that

$$\mathcal{D}_1 = \text{init}\ \dfrac{}{A, \Gamma' \Rightarrow_p^m A}$$

suppose additionally that we have the final rule of $\mathcal{D}_2$ can be written as follows

$$\begin{array}{c} \mathcal{D}_2 \\[4pt] \text{R}_2\ \dfrac{\Gamma'' \Rightarrow_p^{n-1} C''}{A, \Gamma \Rightarrow_p^n C} \end{array}$$

with $\Gamma = A, \Gamma'$. We construct the following derivation $\mathcal{D}$ of $\Gamma \Rightarrow_p C$:

$$\begin{array}{c} \mathcal{D}_2 \\[4pt] \text{ctr}\ \dfrac{\text{R}_2\ \dfrac{\Gamma'' \Rightarrow C''}{A, A, \Gamma' \Rightarrow C}}{A, \Gamma' \Rightarrow C} \end{array}$$

Suppose instead that A is principal in both R_1 and R_2. We consider the following case. Suppose $R_1 = \to_R$ and $R_2 = \to_L$. That is we have that rule R_1 can be written as follows

$$\begin{matrix}\mathcal{D}_1\\ {\to}\mathsf{R}\ \dfrac{A, \Gamma \Rightarrow_p^{m-1} B}{\Gamma \Rightarrow_p^{m} A \to B}\end{matrix}$$

and R_2 can be written as follows

$$\begin{matrix}\mathcal{D}_2' \qquad\qquad\qquad \mathcal{D}_2''\\ {\to}\mathsf{L}\ \dfrac{A \to B, \Gamma \Rightarrow_p^{n1} A \qquad B, \Gamma \Rightarrow_p^{n2} C}{A \to B, \Gamma \Rightarrow_p^{n} C}\end{matrix}$$

with $n_1, n_2 < n$. Then, we can construct the following derivation $\mathcal{D}$ of $\Gamma \Rightarrow_p C$

$$\mathsf{cut}\ \dfrac{\mathsf{cut}\ \dfrac{\mathsf{cut}\ \dfrac{{\to}\mathsf{R}\ \dfrac{\overset{\mathcal{D}_1}{A, \Gamma \Rightarrow B}}{\Gamma \Rightarrow A \to B} \qquad \overset{\mathcal{D}_2'}{A \to B, \Gamma \Rightarrow A}}{\Gamma \Rightarrow A} \qquad \overset{\mathcal{D}_1}{A, \Gamma \Rightarrow B}}{\Gamma \Rightarrow B} \qquad \overset{\mathcal{D}_2''}{B, \Gamma \Rightarrow C}}{\Gamma \Rightarrow C}$$

Now suppose A is not principal in R_1 and R_1 is a one-premise rule.

$$\begin{matrix}\mathcal{D}_1\\ \mathsf{R}_1\ \dfrac{\Gamma' \Rightarrow_p^{m-1} A}{\Gamma \Rightarrow_p^{m} A}\end{matrix} \qquad\qquad \begin{matrix}\mathcal{D}_2\\ \mathsf{R}_2\ \dfrac{\Gamma'' \Rightarrow_p^{n-1} C''}{A, \Gamma \Rightarrow_p^{n} C}\end{matrix}$$

Then, we construct the following derivation $\mathcal{D}$ of $\Gamma \Rightarrow_p \Delta$:

$$
\dfrac{\dfrac{\begin{matrix}\mathcal{D}_1\\ \Gamma' \Rightarrow , A\end{matrix}}{\Gamma', \Gamma \Rightarrow A}\,\mathsf{wk} \qquad \dfrac{\dfrac{\begin{matrix}\mathcal{D}_2\\ \Gamma'' \Rightarrow C''\end{matrix}}{A, \Gamma \Rightarrow C}\,\mathsf{R}_2}{A, \Gamma', \Gamma \Rightarrow C}\,\mathsf{wk}}{\dfrac{\dfrac{\Gamma', \Gamma \Rightarrow C}{\Gamma, \Gamma \Rightarrow C}\,\mathsf{R}_1}{\Gamma \Rightarrow C}}\,\mathsf{cut}
$$

Much in the same way we can consider the remaining cases. □

Theorem 11.2 (Cut Elimination)**.** *If we have a derivation $\mathcal{D}$ of $\Gamma \Rightarrow_p C$ then we can construct a derivation D^* of $\Gamma \Rightarrow_0 C$, that is, a derivation where cut does not occur.*

Proof (Intuition). We apply the proof transformation detailed in the Lemma to the cuts occurring in $\mathcal{D}$, starting with topmost cuts of maximal rank. The Lemma ensures us that after every proof transformation one instance of cut is eliminated. Therefore, in finitely many steps, we obtain a derivation $\mathcal{D}^*$ of $\Gamma \Rightarrow_0 C$, where the cut rule does not occur. □

12 Normalization

Previously, we saw the correspondence between natural deduction and the simply typed lambda calculus. In this section we extend this correspondence to reduction. In particular, we will see how to define reduction on $\mathcal{NJ}$ proofs and see that they reduce to normal forms, which are in bijection with reductions and normal forms in the lambda calculus.

We begin by defining a *detour* in an $\mathcal{NJ}$ proof, which consists of an introduction rule immediately followed by an elimination rule. The *major premise* of the elimination rule is that which coincides with the conclusion of the introduction rule.

The following detours may occur in our proof:

$$\dfrac{\dfrac{M_1 : A_1 \qquad M_2 : A_2}{\langle M_1, M_2\rangle : A_1 \wedge A_2}\wedge_I}{\pi_i\langle M_1, M_2\rangle : A_i}\wedge_{Ei} \qquad \rightsquigarrow \qquad M_i : A_i$$

$$\dfrac{\dfrac{L : A_i}{\mathsf{in}_i(L) : A_1 \vee A_2}\vee_{Ii} \qquad \begin{matrix}[x_1 : A_1]\\ \vdots \\ M_1 : C\end{matrix} \qquad \begin{matrix}[x_2 : A_2]\\ \vdots \\ M_2 : C\end{matrix}}{\mathsf{case}(\mathsf{in}_i(L), x_1 \Rightarrow M_1, x_2 \Rightarrow M_2) : C}\vee E \qquad \rightsquigarrow \qquad \begin{matrix}L : A_i\\ \vdots \\ M_i[L/x_i] : C\end{matrix}$$

$$\dfrac{\dfrac{\begin{matrix}[x : A]\\ \vdots \\ M : B\end{matrix}}{\lambda x.M : A \to B}{}^{I} \qquad N : A}{\mathsf{app}(\lambda x.M, N) : B}{}_{E} \qquad \rightsquigarrow \qquad \begin{matrix}N : A\\ \vdots \\ M[N/x] : B\end{matrix}$$

The term annotations $\langle\cdot,\cdot\rangle, \pi_i, \mathsf{case}, \mathsf{in}_i, \lambda$ are only there to highlight the correspondence with the simply typed lambda calculus. Here we immediately see that these detours correspond to β-reduction the lambda calculus, and the reductions appear on the right. Whereas in the lambda calculus, variables are substituted for terms, in $\mathcal{NJ}$ we substitute an open leaf (the leaf is open in the subproof, but closed in the global proof) with a proof of the leaf's formula.

13 From $\mathcal{LJ}$ to $\mathcal{NJ}$ and Back

13.1 Back to Normalization

14 Peano Arithmetic with Sequents

References

[1] Morten Heine Sørensen, Pawel Urzyczyn. *Lectures on the Curry-Howard Isomorphism*. 2006.

[2] Negri, S., Von Plato, J., Ranta, A. *Structural Proof Theory.* 2001.

[3] Troelsta, Schwichtenberg *Basic Proof Theory.* 1996.

[4] Buss. *Handbook of proof theory*. 1998.

[5] Jean-Yves Girard, Paul Taylor, Yves Lafont. *Proofs and Types*. 1989.