Lecture Notes

Proof Theory

Foundations and Applications

David Baelde
LSV, ENS Cachan

June 2013, UAIC, Iasi, Romania

Contents

Introduction

Born about a century ago from the foundational crisis of mathematics, proof theory has evolved into a rich, independent field of study with its own motivations. The meeting of proof theory and computer science has been particularly fruitful: logic can provide foundations for computing, and computers can automate (part of) the reasoning process. As a result of this interaction, we now have tools for building and checking fully formal proofs, whose success spans from software verification to formalized mathematics.

In this course, we will provide an introduction to the basic concepts of proof theory and show how they provide a foundation for complex applications in formal reasoning. In the first part, we will study sequent calculus for propositional logics. We will highlight the common features of sequent calculi, and the central role of the cut rule and the cut elimination theorem. This will give us the opportunity to mention numerous connections between proof theory and computer science, and the associated research questions. In the second part of the course, we will show how these foundations give us a firm ground on which to build much richer proof systems. We will move from propositional to first-order logic, and then enrich our logic with fixed point definitions, a.k.a. (co)inductive specifications, to obtain a system in which we can prove useful properties of discrete mathematics, programs, and more broadly computer science.

Chapter 1

Basic Sequent Calculus

We start with a common and simple logic: classical propositional logic. After reviewing its boolean truth semantics, we introduce the sequent calculus $\mathbf{LK}^0$ for it, discuss its structure, and show central results such as completeness and cut elimination. On the way, we shall mention some connections between proof theory and computer science. Finally, we introduce intuitionistic logic which makes some of these connections more evident.

1.1 Classical Propositional Logic

We define the syntax of propositional logic and its truth semantics. This section is intentionally quite rough. We only include it to be able to ground the following developments in some intuitive notions.

Definition 1. *Let $\mathcal{P} = \{\ p, q, r, \dots\ \}$ be a countable set of propositional constants. Formulas of propositional logic are then given by the following grammar:*

$$P, Q ::= p \mid \bot \mid \top \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \supset Q$$

Definition 2. *Let $\mathbb{B} = \{\ 0, 1\ \}$ and $+, \times$ be the maximum and minimum operations on that set. Given an interpretation for propositional constants $\mathcal{I} : \mathcal{P} \to \mathbb{B}$, we define the interpretation of a formula by induction on the structure of the formula:*

$$\begin{array}{rcl}
[p]^{\mathcal{I}} & = & \mathcal{I}(p) \\
[\bot]^{\mathcal{I}} & = & 0 \\
[\top]^{\mathcal{I}} & = & 1 \\
[\neg P]^{\mathcal{I}} & = & 1 - [P]^{\mathcal{I}} \\
[P \vee Q]^{\mathcal{I}} & = & [P]^{\mathcal{I}} + [Q]^{\mathcal{I}} \\
[P \wedge Q]^{\mathcal{I}} & = & [P]^{\mathcal{I}} \times [Q]^{\mathcal{I}} \\
[P \supset Q]^{\mathcal{I}} & = & (1 - [P]^{\mathcal{I}}) + [Q]^{\mathcal{I}}
\end{array}$$

Definition 3. *We say that a formula P is* satisfiable *if there exists an $\mathcal{I}$ such that $[P]^{\mathcal{I}} = 1$. We say that the formula is* valid*, or simply* true*, if for all $\mathcal{I}$ we have $[P]^{\mathcal{I}} = 1$.*

Example 1. *Let $p, q \in \mathcal{P}$. The formula $p \supset q$ is satisfiable: it suffices to take an interpretation such that $\mathcal{I}(p) = 0$, or one such that $\mathcal{I}(q) = 1$. However that formula is not valid when $p \neq q$: consider $\mathcal{I}(p) = 1$ and $\mathcal{I}(q) = 0$.*

Proposition 1 (Basic identities)**.** *Let P and Q be arbitrary formulas, then we have the following equivalences with respect to our semantics:*

$$\neg(P \wedge Q) \equiv (\neg P \vee \neg Q) \qquad \neg(P \vee Q) \equiv (\neg P \wedge \neg Q)$$

$$\neg\neg P \equiv P \qquad P \supset Q \equiv \neg P \vee Q$$

The first two identities are called de Morgan duality between conjunction and disjunction.

Exercise 1. *Which of the following facts is true? (1) If $P \wedge Q$ is valid then both P and Q are valid. (2) If $P \vee Q$ is valid then one of P and Q is valid.*

Exercise 2. *Let P and Q be arbitrary formulas such that $P \supset Q$ and $Q \supset P$ are both valid. Let $p \in \mathcal{P}$. Show that $R[P/p]$ is valid iff $R[Q/p]$ is valid, where $[P/p]$ denotes the substitution of P for p.*

1.2 Sequent Calculus

In the previous section, we have defined a *syntax* for our logic, and a truth *semantics* based on the interpretation of our syntax in the *model* $\mathbb{B}$. In the case of propositional logic, the construction is quite satisfying, because the syntax is explained by means of a semantics that is somewhat simpler. With richer logics, for example systems containing arithmetic, the situation would not be that simple: we would typically presuppose the existence of natural numbers in order to obtain a model of arithmetic. We are now going to switch to a different approach based on the notion of proof. This is appealing because we can design simple, syntactical proof systems for arbitrarily rich logics. Of course, foundational issues come back one way or the other: here, it will be when we try to prove that our logic is consistent. In the end, both approaches are used for various purposes, and it is important to be able to relate the two approaches. We shall do it for classical logic, but after this our main focus will be on understanding the general structure of sequent calculi.

Definition 4. *A* sequent *is a pair of multisets of formulas, written $\Gamma \vdash \Delta$. An* $\mathbf{LK}^0$ derivation *(or* proof*) is a tree-like derivation built using the inference rules of Figure 1.1.*

Intuitively, a sequent $\Gamma \vdash \Delta$ should be read as "the conjunction of formulas in Γ implies the disjunction of the formulas in Δ." One can check that this

Identity group

$$\frac{}{P \vdash P}\ \text{axiom} \qquad \frac{\Gamma \vdash \Delta, P \quad \Gamma', P \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \text{cut}$$

Logical group

$$\frac{}{\Gamma, \bot \vdash \Delta}\ \bot \qquad \frac{}{\Gamma \vdash \top, \Delta}\ \top$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ {\wedge}L \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'}\ {\wedge}R$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \vee B \vdash \Delta, \Delta'}\ {\vee}L \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}\ {\vee}L$$

$$\frac{\Gamma \vdash P, \Delta \quad \Gamma', Q \vdash \Delta'}{\Gamma, \Gamma', P \supset Q \vdash \Delta, \Delta'}\ {\supset}L \qquad \frac{\Gamma, P \vdash Q, \Delta}{\Gamma \vdash P \supset Q, \Delta}\ {\supset}R$$

Structural group

$$\frac{\Gamma \vdash \Delta}{\Gamma, P \vdash \Delta}\ WL \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash P, \Delta}\ WR$$

$$\frac{\Gamma, P, P \vdash \Delta}{\Gamma, P \vdash \Delta}\ CL \qquad \frac{\Gamma \vdash P, P, \Delta}{\Gamma \vdash P, \Delta}\ CR$$

Figure 1.1: Inference rules for $\mathbf{LK}^0$

intuition is respected by the rules of inference of our calculus, and this will be made formal in the final theorem of the section.

Inference rules are organized in three groups. The *structural group* is made of the rules WL and WR, called *weakening* rules, and CL and CR which are *contractions*. We do not include exchange rules to move formulas at different places in sequents. Instead our rules should be understood as applying regardless of the position of a formula in the sequent. The *logical group* is the core of reasoning. The applicability of its rules only depends on the outermost connective of one formula in the concluding sequent, which creates a tight connection between the notions of connective and logical rule. That formula is called *principal*. The *identity group* contains the only rules that require a notion of equality on formulas: axiom and cut. To check a proof starting with one of these rules, one has to check that two formulas are the same. We say that a formula is *active* in a rule application when it is singled out in the rule: it is the formula being contracted or weakened, the principal formula in a logical rule, or a formula in the axiom rule.

In order to familiarize ourselves with the calculus, we are going to make a few elementary observations before the big results.

Exercise 3. *Derive the following sequents in* $\mathbf{LK}^0$*:* $\vdash A \vee \neg A$*,* $A \wedge B \vdash B \wedge A$*,* $A \vee B \vdash B \vee A$*,* $A \vee (B \wedge C) \vdash (A \vee B) \wedge (A \vee C)$*, and the equivalences of Proposition 1.*

Proposition 2. *The general axiom rule from Figure 1.1 can be derived from the other rules and the axiom restricted to atomic formulas.*

Proof. By induction on P, we introduce one by one the logical connectives of P until we reach an axiom. □

Proposition 3. *The rule* $\wedge L$ *is invertible, meaning that if its conclusion is derivable, then so is its premise. From the viewpoint of somebody trying to build a proof of the conclusion, this means that applying the rule will never loose provability.*

Proof. Let us first observe that not all rules are invertible. To see that, consider the following application of $\wedge R$:

$$\frac{A \wedge B \vdash B \qquad \vdash A}{A \wedge B \vdash B \wedge A}$$

While the bottom sequent is certainly provable, this is definitely not the case of the right premise.

Coming back to $\wedge L$, there is an easy proof of its invertibility: assuming a derivation Π, we create a new one that ends with $\wedge L$, then uses cut to "undo" the left conjunction rule and get back to the original sequent.

$$\cfrac{\cfrac{\cfrac{\cfrac{}{A \vdash A} \quad \cfrac{}{B \vdash B}}{A, B \vdash A \wedge B} \qquad \cfrac{\Pi}{\Gamma, A \wedge B \vdash \Delta}}{\Gamma, A, B \vdash \Delta}\ \text{cut}}{\Gamma, A \wedge B \vdash \Delta}$$

This version of the result is a priori not useful for proof search: we would like to know not only that the application does not loose provability but also that it gets us closer to completing a proof. To get some insight in that respect, we present an argument based on proof transformations, or more precisely rule permutations.

Let Π be a derivation of $\Gamma, (P \wedge Q)^n \vdash \Delta$. We prove by induction on Π that there is a derivation of $\Gamma, P^n, Q^n \vdash \Delta$.

- If the last rule does not apply to one of the $P \wedge Q$ formulas, we apply the induction hypothesis were applicable and conclude by permuting the last rule with $\wedge L$ as needed.

- Otherwise, the rule can be an axiom, $\wedge L$, or a structural rule. We have seen that the axiom can be expanded in a way that allows us to conclude. There is nothing to do if the rule is $\wedge L$. If one of our formulas is weakened away, we produce the required derivation by weakening the corresponding subformulas P and Q. If one of the $P \wedge Q$ is contracted then by induction hypothesis we have a derivation of $\Gamma, P^{n+1}, Q^{n+1} \vdash \Delta$, and by applying two contraction rules we obtain the expected result.

□

Exercise 4. *Define the negation normal form of a formula by $\psi(P)$, using the identities of Proposition 1 to simplify any formula into a form without implication and where negation is restricted to atoms. The* one-sided *sequent calculus for classical propositional logic is obtained by removing all left rules as well as logical rules for implication and negation from $\mathbf{LK}^0$, and adapting axiom and cut as follows:*

$$\frac{}{\vdash p, \neg p} \qquad \frac{\vdash \Gamma, P \quad \vdash \Gamma', \psi(\neg P)}{\vdash \Gamma, \Gamma'}$$

Show that this system is equivalent, i.e., *$\vdash P$ is provable in $\mathbf{LK}^0$ iff it is provable in the one-sided system. Hint: generalize to $\vdash \psi(\neg \wedge \Gamma), \psi(\Delta)$.*

We conclude the section by establishing that our proof system is sound and complete. This requires introducing a set of alternative rules first – the reasons for not working with these rules from the beginning will become apparent in the next section.

Proposition 4. *The inference rules of Figure 1.2 are all admissible, meaning that if their premises are derivable in $\mathbf{LK}^0$ then so is their conclusion.*

Proof. This is easily checked. Each rule is derived from the corresponding rule from Figure 1.1 and structural rules. □

Theorem 1. *The calculus $\mathbf{LK}^0$ is sound and complete for propositional classical logic,* i.e., *P is valid iff $\vdash P$ is provable.*

$$\overline{\Gamma, P \vdash P, \Delta}$$

$$\overline{\Gamma, \bot \vdash \Delta} \qquad \overline{\Gamma \vdash \top, \Delta}$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta} \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}$$

$$\frac{\Gamma \vdash P, \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \supset Q \vdash \Delta} \qquad \frac{\Gamma, P \vdash Q, \Delta}{\Gamma \vdash P \supset Q, \Delta}$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta} \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}$$

Figure 1.2: A complete set of invertible rules for propositional classical logic

Proof. Given a derivation Π of $\Gamma \vdash \Delta$ we prove by induction on Π that the sequent is valid. This is straightforward: it suffices to check for each rule that if the premises are valid then so is the conclusion. For completeness, we use the invertible rules of Figure 1.2. By induction on the total number of logical connectives in a valid sequent $\Gamma \vdash \Delta$, we build a derivation of it. □

The careful reader will have noted that the precise choice of rules in Figure 1.1 is what makes Exercise 4 work smoothly. Indeed, we have chosen pairs of rules for $\wedge$, $\vee$ and $\supset$ which closely reflect the basic identities of Proposition 1 at the level of proofs. The cut reductions of the next section also dictate some symmetries in our design. The choice of rules in Figure 1.2 does not enjoy those symmetries, although the resulting system proves exactly the same theorems. If we follow symmetries between left and right rules rather than invertibility as a guideline, we obtain yet another presentation.

Definition 5. *The* additive *connectives* & *and* $\oplus$ *are defined by the following pairs of rules.*

$$\frac{\Gamma, A_i \vdash \Delta}{\Gamma, A_0 \& A_1 \vdash \Delta}\ \&L \qquad \frac{\Gamma \vdash A_0, \Delta \quad \Gamma \vdash A_1, \Delta}{\Gamma \vdash A_0 \& A_1, \Delta}\ \&R$$

$$\frac{\Gamma, A_0 \vdash \Delta \quad \Gamma, A_1 \vdash \Delta}{\Gamma, A_0 \oplus A_1 \vdash \Delta}\ \oplus L \qquad \frac{\Gamma \vdash A_i, \Delta}{\Gamma \vdash A_0 \oplus A_1, \Delta}\ \oplus R$$

The connectives $\wedge$ *and* $\vee$ *are called* multiplicative*. Notations vary a lot here, so the terminology should be understood as referring to the rules, not the symbols used.*

Exercise 5. *Show that $A \wedge B \equiv A\&B$ and $A \vee B \equiv A \oplus B$ for all A and B. Show that $\neg(A \oplus B) \equiv \neg A \& \neg B$. Check that this duality is present at the level of proofs: the transformation from Exercise 4 turns $\&L$ into $\oplus R$, etc.*

1.3 Cut Elimination

Gentzen concentrated in the cut rule all the inventive, indirect part of reasoning. Then he observed that this rule could be eliminated: any proof with cut can be transformed into a cut-free proof. This is a very powerful result because cut-free proofs have a simple structure that is easy to analyze.

Proposition 5. *There is no cut-free derivation of $\vdash \bot$. Hence the logic is consistent if it eliminates cut.*

Proof. By cut elimination it suffices to show that there is no cut-free derivation of $\vdash \bot$. This is obvious because the only applicable rules on that sequent are structural rules, and they do not allow to conclude. □

A stronger observation about cut-free proofs is that they form an *analytic* proof system: deduction does not rely on any invention, but only involves inspecting sub-components of the goal formulas.

Proposition 6 (Subformula property)**.** *All formulas occurring in a cut free derivation of $\Gamma \vdash \Delta$ are subformulas of formulas occurring in Γ or Δ.*

Proof. This follows from a simple inspection of the rules. □

We shall now prove that cut is admissible, *i.e.*, that it can be eliminated. In the above proof of completeness, we have actually built a cut-free derivation for any valid formula. Thus we have an indirect, *semantic* cut elimination proof for our logic: if a sequent is provable then it is valid, and by completeness it has a cut-free derivation.

Gentzen showed cut elimination in a more direct way. His approach is interesting because it does not require a notion of model, but relies only on proofs. We shall define a set of proof transformations, called *cut reductions*, such that irreducible proofs are cut free, and show that any proof can be reduced in a finite number of steps into an irreducible, and thus cut-free proof. There are a lot of syntactical details to consider! To get the essential idea behind all these symbols, it is useful to keep in mind some intuitions — that Gentzen probably did not have. One should not think too much of sequents as sets or even multisets, as it is useful to follow the "path" of a given formula through the proof. This way a proof appears as a recipe for manipulating formulas, working on one and then on the other. Following individual formulas in a proof, we can "see" wires, and it becomes obvious that what happens on one wire is quite independent from what happens on another: the order of rule applications on distinct formulas is irrelevant. This kind of intuition is behind the first set of reductions.

Definition 6 (Auxiliary cut reductions)**.** *These reductions apply to any cut which has a subderivation where the cut formula is not immediately active. There are many rules with lots of symmetric cases, but the essential idea is always the same: permuting the cut above the other rule.*
Permuting cut and left conjunction rules:

$$\dfrac{\Gamma \vdash C, \Delta \qquad \dfrac{\Gamma', A, B, C \vdash \Delta'}{\Gamma', A \wedge B, C \vdash \Delta'}}{\Gamma, \Gamma', A \wedge B \vdash \Delta, \Delta'}\,\text{cut} \qquad \longrightarrow \qquad \dfrac{\dfrac{\Gamma \vdash C, \Delta \quad \Gamma', A, B, C \vdash \Delta'}{\Gamma, \Gamma', A, B \vdash \Delta, \Delta'}\,\text{cut}}{\Gamma, \Gamma', A \wedge B \vdash \Delta, \Delta'}$$

Permuting cut and right conjunction rules:

$$\dfrac{\Gamma \vdash C, \Delta \qquad \dfrac{\Gamma', C \vdash A, \Delta' \quad \Gamma'' \vdash B, \Delta''}{\Gamma', \Gamma'', C \vdash A \wedge B, \Delta', \Delta''}}{\Gamma, \Gamma', \Gamma'' \vdash A \wedge B, \Delta, \Delta', \Delta''}\,\text{cut}$$

$$\downarrow$$

$$\dfrac{\dfrac{\Gamma \vdash C, \Delta \quad \Gamma', C \vdash A, \Delta'}{\Gamma, \Gamma' \vdash A, \Delta, \Delta'}\,\text{cut} \qquad \Gamma'' \vdash B, \Delta''}{\Gamma, \Gamma', \Gamma'' \vdash A \wedge B, \Delta, \Delta', \Delta''}$$

Permuting cut and weakening:

$$\dfrac{\dfrac{\Gamma \vdash C, \Delta}{\Gamma \vdash A, C, \Delta} \qquad \Gamma', C \vdash \Delta'}{\Gamma, \Gamma' \vdash A, \Delta, \Delta'}\,\text{cut} \qquad \longrightarrow \qquad \dfrac{\dfrac{\Gamma \vdash C, \Delta \quad \Gamma', C \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\,\text{cut}}{\Gamma, \Gamma' \vdash A, \Delta, \Delta'}$$

Permuting cut and contraction:

$$\dfrac{\dfrac{\Gamma \vdash A, A, C, \Delta}{\Gamma \vdash A, C, \Delta} \qquad \Gamma', C \vdash \Delta'}{\Gamma, \Gamma' \vdash A, \Delta, \Delta'}\,\text{cut} \qquad \longrightarrow \qquad \dfrac{\dfrac{\Gamma \vdash A, A, C, \Delta \quad \Gamma', C \vdash \Delta'}{\Gamma, \Gamma' \vdash A, A, \Delta, \Delta'}\,\text{cut}}{\Gamma, \Gamma' \vdash A, \Delta, \Delta'}$$

We also allow the permutation of two cuts:

$$\dfrac{\Gamma \vdash A, \Delta \qquad \dfrac{\Gamma', A \vdash B, \Delta' \quad \Gamma'', B \vdash \Delta''}{\Gamma', \Gamma'', A \vdash \Delta', \Delta''}\,\text{cut}(B)}{\Gamma, \Gamma', \Gamma'' \vdash \Delta, \Delta', \Delta''}\,\text{cut}(A)$$

$$\downarrow$$

$$\dfrac{\dfrac{\Gamma \vdash A, \Delta \quad \Gamma', A \vdash B, \Delta'}{\Gamma, \Gamma' \vdash B, \Delta, \Delta'}\,\text{cut}(A) \qquad \Gamma'', B \vdash \Delta''}{\Gamma, \Gamma', \Gamma'' \vdash \Delta, \Delta', \Delta''}\,\text{cut}(B)$$

And several other similar cases. . .

Definition 7 (Principal cut reductions)**.** *These reductions cover all cases where the cut formula is active in both subderivations.*

If one subderivation is an axiom, then the cut reduces to the other subderivation. Here is one such case, where the cut formula is underlined to help keeping track of things:

$$\dfrac{\Gamma \vdash \underline{C}, \Delta \qquad \dfrac{}{\underline{C} \vdash C}\,\text{axiom}}{\Gamma \vdash C, \Delta}\,\text{cut} \qquad \longrightarrow \qquad \Gamma \vdash C, \Delta$$

When the cut formula is a conjunction, the cut is reduced into two cuts on the immediate subformulas of the original cut formula:

$$\dfrac{\dfrac{\Gamma' \vdash P, \Delta' \quad \Gamma'' \vdash Q, \Delta''}{\Gamma', \Gamma'' \vdash P \wedge Q, \Delta', \Delta''} \quad \dfrac{\Gamma, P, Q \vdash \Delta}{\Gamma, P \wedge Q \vdash \Delta}}{\Gamma', \Gamma'', \Gamma \vdash \Delta', \Delta'', \Delta}\,\text{cut}$$

$$\downarrow$$

$$\dfrac{\Gamma' \vdash P, \Delta' \qquad \dfrac{\Gamma'' \vdash Q, \Delta'' \quad \Gamma, P, Q \vdash \Delta}{\Gamma'', \Gamma, P \vdash \Delta'', \Delta}\,\text{cut}}{\Gamma', \Gamma'', \Gamma \vdash \Delta', \Delta'', \Delta}\,\text{cut}$$

If the cut formula is weakened away, then the corresponding parts of the sequent are weakened away and the cut disappears:

$$\dfrac{\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash C, \Delta} \quad \Gamma', C \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\,\text{cut} \qquad \longrightarrow \qquad \dfrac{\Gamma \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

If the cut formula is contracted, the cut is reduced into two cuts, with one subderivation being duplicated:

$$\dfrac{\dfrac{\Gamma \vdash C, C, \Delta}{\Gamma \vdash C, \Delta} \quad \Gamma', C \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\,\text{cut}$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash C, \underline{C}, \Delta \quad \Gamma', \underline{C} \vdash \Delta'}{\Gamma, \Gamma' \vdash \underline{C}, \Delta, \Delta'}\,\text{cut} \qquad \Gamma', \underline{C} \vdash \Delta'}{\Gamma, \Gamma', \Gamma' \vdash \Delta, \Delta', \Delta'}\,\text{cut}}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

And several other similar cases…

Example 2. *Consider two derivations: first, a derivation of $\vdash (a \wedge a) \vee \neg a$; second, a derivation of $(a \wedge a) \vee \neg a \vdash a \vee \neg a$ which performs a case analysis on the disjunction, instead of proving $a \vee \neg a$ directly without using the hypothesis. Cut these two derivations to obtain a proof of $\vdash a \vee \neg a$, and apply the above rule to reduce it. Eventually it becomes a cut of $\vdash a \wedge a, \neg a$ against an axiom on $\neg a$ and a derivation of $a \wedge a \vdash a$. The axiom goes away, the contraction on $\neg a$ passes below the cut, then a principal reduction can occur on the conjunction.*

At this point we have two cuts on a but one is weakened away. Finally, axioms are simplified and we end up with:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a \vdash a}}{\vdash a, \neg a}}{\vdash a, \neg a, \neg a}}{\vdash a, \neg a}}{\vdash a \vee \neg a}$$

Exercise 6. *Write cut reduction (auxiliary and principal) for the additive connectives of Definition 5. What happens if you create a "Frankenstein" connective,* e.g.*, with the left rule of multiplication conjunction and the right rule of additive conjunction?*

Theorem 2. *If $\Gamma \vdash \Delta$ has a derivation then it has a cut-free derivation.*

To prove this theorem, Gentzen showed that there always exists a strategy for applying the above rules to obtain a cut-free proof. To do so in the style of Gentzen, one needs additional devices that are a bit heavy. To see why, let us observe the effect of cut reductions, omitting the permutation of two cuts which is not necessary to eliminate cuts. When performing an auxiliary reduction on a cut, the cut is either removed (in the case of units) or it is changed into a new cut with one strictly smaller subderivation. Except for case of contraction, principal reductions either remove a cut or change it into cuts on smaller cut formulas. The case of contraction is more complex, as it does not reduce the size of the cut formula nor that of the cut's subderivations. If we reduce the deepest cut, there is no hope to control the size of the resulting derivation. What can save us, though, is that the subderivations of the new cuts are subderivations of the original one. Gentzen's solution is to package cuts and contractions together in a new rule called *mix*, as a way to present a special reduction strategy in which the subderivations that are duplicated by contractions are not treated separately but kept together as long as possible. With this device, he could show that a particular reduction strategy terminates using a lexicographic ordering. The introduction of the *mix* rule makes the presentation of reduction rules more tedious, with no gain for our main purpose, so we will see a different proof.

Our proof technique is more abstract than a termination argument based on a simple ordering, but it has the advantage of working on the bare reductions presented above, and of scaling well to richer logics or stronger results. In fact, it is an adaptation of techniques that were first used to proof *strong* normalization arguments for typed λ-calculus. More specifically, the following definitions are adapted from Girard's proof of strong normalization for linear logic proof nets.

Definition 8. *A derivation of* type *A (resp. $A^\perp$) is a derivation with a distinguished formula A in the right (resp. left) side of its conclusion sequent. We shall denote types by T, and we define $T^\perp$ by $(A)^\perp = A^\perp$ and $(A^\perp)^\perp = A$. This way we can talk of left and right types in a symmetrical way. We say that two derivations are* compatible *if they are of respective types T and $T^\perp$ for some T.*

Definition 9. *Given two normalizing and compatible derivations* Π *and* Π'*, we say that* $\Pi \perp \Pi'$ *when* $\mathrm{cut}(\Pi, \Pi')$ *normalizes. For a set of normalizing derivations* $\mathcal{X}$ *of type* T *and for a normalizing derivation* Π *of type* $T^{\perp}$*, we write* $\Pi \perp \mathcal{X}$ *if* $\Pi \perp \Pi'$ *for all* $\Pi' \in \mathcal{X}$*. Finally we define* $\mathcal{X}^{\perp}$ *as* $\{\ \Pi\ :\ \Pi \perp \mathcal{X}\ \}$*.*

Proposition 7. *For any set of normalizing proofs* $\mathcal{X}$ *of type* T*, we have* $\mathcal{X} \subseteq \mathcal{X}^{\perp\perp}$*, and* $\mathcal{X}^{\perp} = \mathcal{X}^{\perp\perp\perp}$*.*

Proof. The first part follows simply by unfolding twice the definition. This already gives us one direction of the second part. For other direction it suffices to observe that if $X \subseteq Y$ then $Y^{\perp} \subseteq X^{\perp}$: this gives us $X^{\perp\perp\perp} \subseteq X^{\perp}$ from the first fact. □

Proposition 8. *For any set* $X = Y^{\perp}$ *(in practice we will always take* $X = X^{\perp\perp}$*) we have* axiom $\in X$ *and, if* Π *reduces to* $\Pi' \in X$*, then* $\Pi \in X$*.*

Proof. For any normalizing proof, $\mathrm{cut}(\Pi, \mathrm{axiom})$ reduces to Π and hence normalizes. This gives us the first part. For the second part, we have $\Pi' \in Y^{\perp}$, so $\Pi' \perp \Theta$ for each $\Theta \in Y$, and we seek to obtain $\Pi \perp \Theta$ for each such Θ. This follows simply from the fact that $\mathrm{cut}(\Pi, \Theta)$ reduces to $\mathrm{cut}(\Pi', \Theta)$, and hence normalizes. □

For concision, we use the formula constructors as a notation for proofs and proof constructions. For instance, $\top$ stands for the $\top$ rule, and $\Pi \wedge \Pi'$ stands for the proof ending with a right conjunction rule and whose subderivations are Π and Π'. We also write things like $\mathrm{cut}(\Pi, \Pi')$ or even $\mathrm{cut}(\Pi, \Pi_1, \Pi_2 \ldots)$ when the precise structure of the cuts is irrelevant or given by the context — note that since we allow cut permutations, the order of cuts does not matter, we only need to know on which formula each cut is performed and how the contexts are split.

Definition 10. *For a formula* A*, we define* $[A]$ *as a set of proofs of type* A*, by induction on* A*:*

$$\begin{array}{rcl}
[\perp] & = & \emptyset^{\perp\perp} \\
[\top] & = & \{\top\}^{\perp\perp} \\
[\neg A] & = & [A]^{\perp} \\
[A \wedge B] & = & \left\{ \dfrac{\dfrac{\Pi_A}{\Gamma \vdash A, \Delta} \quad \dfrac{\Pi_B}{\Gamma' \vdash B, \Delta'}}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'} \ \textit{where}\ \Pi_A \in [A], \Pi_B \in [B] \right\}^{\perp\perp} \\
[A \vee B] & = & \left\{ \dfrac{\dfrac{\Pi_A}{\Gamma, A \vdash \Delta} \quad \dfrac{\Pi_B}{\Gamma', B \vdash \Delta'}}{\Gamma, \Gamma', A \vee B \vdash \Delta, \Delta'} \ \textit{where}\ \Pi_A \in [A]^{\perp}, \Pi_B \in [B]^{\perp} \right\}^{\perp} \\
[A \supset B] & = & \left\{ \dfrac{\dfrac{\Pi_A}{\Gamma \vdash A, \Delta} \quad \dfrac{\Pi_B}{\Gamma', B \vdash \Delta'}}{\Gamma, \Gamma', A \supset B \vdash \Delta, \Delta'} \ \textit{where}\ \Pi_A \in [A], \Pi_B \in [B]^{\perp} \right\}^{\perp}
\end{array}$$

We extend the definition to contexts in a natural way: if $\Gamma = (A_1, \ldots, A_n)$ *then* $[\Gamma] = [A_1] \times \ldots \times [A_n]$ *and* $[\Gamma]^\perp = [A_1]^\perp \times \ldots \times [A_n]^\perp$*.*

Definition 11. *A derivation* Π *of* $\Gamma \vdash \Delta$ *is said to be reducible if for all* $\vec{\gamma} \in [\Gamma]$, $\vec{\delta} \in [\Delta]^\perp$, $\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta})$ *normalizes.*

Proposition 9. *The derivation* Π *is* $(\Gamma, A \vdash \Delta)$*-reducible iff for all* $\vec{\gamma} \in [\Gamma], \vec{\delta} \in [\Delta]^\perp$ *we have* $\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta}) \in [A]^\perp$*.*

Proof. We show only the first half of the proposition, when A is on the left. Given $\Theta \in [A]$, we first show that $\mathrm{cut}(\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta}), \Theta)$ normalizes. This derivation reduces to $\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta}, \Theta)$, which normalizes by hypothesis on Π. Conversely, assuming $\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta}) \in [A]^\perp$, we need to show that $\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta}, \Theta)$ for $\Theta \in [A]$. This is obvious, we only have to re-arrange cuts to reduce to $\mathrm{cut}(\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta}), \Theta)$ and conclude by hypothesis. □

Theorem 3. *All derivations of* $\Gamma \vdash \Delta$ *are reducible.*

Proof. By induction on the structure of the derivation.

- The case of the axiom is obvious. A cut against a normalizable derivation Θ reduces to Θ which is normalizable.

- Right conjunction rule. We consider $\Pi = \Pi_A \wedge \Pi_B$, that is

$$\dfrac{\dfrac{\Pi_A}{\Gamma \vdash A, \Delta} \quad \dfrac{\Pi_B}{\Gamma' \vdash B, \Delta'}}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'}$$

  By induction hypothesis Π_A is $(\Gamma \vdash A, \Delta)$-reducible and Π_B is $(\Gamma' \vdash B, \Delta')$-reducible. We need to show that $\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\gamma'}, \vec{\delta}, \vec{\delta'}, \Theta)$ normalizes for $\vec{\gamma} \in [\Gamma]$, $\vec{\delta} \in [\Delta]^\perp$ and $\Theta \in [A \wedge B]^\perp$. We observe that this derivations reduces to $\mathrm{cut}(\mathrm{cut}(\Pi_A, \vec{\gamma}, \vec{\delta}) \wedge \mathrm{cut}(\Pi_B, \vec{\gamma'}, \vec{\delta'}), \Theta)$, using auxiliary reductions. We conclude by observing that this reduct normalizes since the left derivation belongs to $[A] \wedge [B]$ which is included in $[A \wedge B]$, and thus it normalizes when cut against Θ.

- Left conjunction rule. We consider the following derivation Π:

$$\dfrac{\dfrac{\Pi'}{\Gamma, A, B \vdash \Delta}}{\Gamma, A \wedge B \vdash \Delta}$$

  By induction hypothesis the derivation Π' is $(\Gamma, A, B \vdash \Delta)$-reducible. We need to show that $\mathrm{cut}(\Pi, \Theta, \vec{\gamma}, \vec{\delta})$ normalizes for $\vec{\gamma} \in [\Gamma]$, $\vec{\delta} \in [\Delta]^\perp$ and $\Theta \in [A \wedge B]$. Our derivation reduces to $\mathrm{cut}(\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta}), \Theta)$ and thus it suffices to show that $\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta})$ belongs to $[A \wedge B]^\perp = ([A] \wedge [B])^\perp$ — we used $X^{\perp\perp\perp} = X^\perp$ and the definition of $[A \wedge B]$. That is obvious because $\mathrm{cut}(\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta}), \Pi_A \wedge \Pi_B)$ reduces to $\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta}, \Pi_A, \Pi_B)$ which normalizes because Π is reducible.

- We leave the case of other logical rules as an exercise. The essential argument for the right conjunction rule can be used for the left disjunction and implication rules: the proof is in the "core" of the interpretation. The argument for the left conjunction rule applies for the right disjunction and implication rules: it suffices to check normalization against that "core".

- The case of a cut is simply done by combining the two induction hypotheses, using the fact that if Π' is $(\Gamma \vdash \Delta, A)$-reducible then $\mathrm{cut}(\Pi', \vec{\delta}, \vec{\gamma})$ belongs to $[A]$.

- Structural rules are handled easily. We show contraction:

$$\dfrac{\dfrac{\Pi'}{\Gamma, A, A \vdash \Delta}}{\Gamma, A \vdash \Delta}$$

  We consider $\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta}, \Theta)$ for $\Theta \in [A]$, and reduce it to a derivation ending with a bunch of contractions, followed by $\mathrm{cut}(\Pi, \vec{\gamma}, \vec{\delta}, \Theta, \Theta)$ which normalizes by induction hypothesis.

□

Exercise 7. *Complete the missing cases in the proof: start with units, then negation, then disjunction and implication.*

Exercise 8. *Propose an interpretation for the additive connectives (& and $\oplus$) and extend the previous theorem to handle them.*

Corollary 1. *All derivations normalize and thus cut is admissible.*

Proof. Given a proof Π, the previous theorem gives us that $\mathrm{cut}(\Pi, \mathrm{id}, \mathrm{id}, \ldots)$ normalizes, because identities belong to all interpretations.

We now establish more generally that if Π is an *id-simplification* of Θ and Θ normalizes, then Π also normalizes. Here, id-simplification means that the former proof is obtained from the latter by reducing cuts against an axiom.

This is done by induction on the reduction sequence of Θ. If the reduction simplifies a cut against an axiom, we conclude immediately by induction hypothesis. In all other cases the reduction can be mimicked to reduce Π, and conclude by induction hypothesis. □

Corollary 2. *The logic is consistent.*

This simply follows from cut elimination and the fact that there is no cut-free derivation of $\vdash \bot$. And so we have managed to prove consistency, by manipulating only syntactical objects. This is not as simple as it seems: showing the termination of cut reductions required a complex mathematical argument, necessarily more complex than what could be done in the logic which we are proving consistent.

1.4 Proofs as Programs

By showing cut elimination, Gentzen actually gave an algorithm for transforming any proof into a cut free proof. In other words, he gave a method for computing the composition of cut-free proofs. Now, if we take this idea to the next level, we may see proofs as programs, or processes, that may evolve in themselves or by interacting when cut against other processes.

So far we have proved that among possible evolutions of a proof, at least one terminates, leading to a cut-free proof. That property is called *weak normalization* of the cut reduction relation. If we take more seriously the idea of a proof as a computing process, we may ask more precise questions about their behavior in cut reductions. Do all possible evolutions terminate? This is *strong normalization.* Can we characterize when two proofs behave the same in all possible interactions? This is called behavioral or contextual equivalence.

Such questions might seem a bit abstract, or even irrelevant when talking about proofs. Even if we may look at $\mathbf{LK}^0$ proofs as programs, they are certainly weird programs. Indeed, the idea of proofs-as-programs started to receive some attention only when people identified a logic for which proofs corresponded to a pre-existing, recognized notion of program. Around the 70's, people realized that there was a deep connection (an isomorphism) between typed λ-calculus and intuitionistic logic, a constructive restriction of classical logic. This connection became known as the Curry-Howard(-Lambek) isomorphism. It was later extended to lots of other logics and abstract programming languages, establishing proof theory as a logical foundation for programming languages, mostly purely functional ones. For instance, classical proofs can be seen as functional programs with exceptions, sequent calculus proofs correspond to λ-terms that are executed by means of explicit substitutions, and linear logic yields calculi with explicit ressource/memory management.

1.5 Intuitionistic Logic

Definition 12. *The system* $\mathbf{LJ}^0$ *is obtained from* $\mathbf{LK}^0$ *by removing right structural rules, and changing the disjunction rule so that the right-hand side of a sequent contains always exactly one formula:*

$$\frac{\Gamma, A_0 \vdash P \quad \Gamma, A_0 \vdash P}{\Gamma, A_0 \vee A_1 \vdash P} \qquad \frac{\Gamma \vdash A_i}{\Gamma \vdash A_0 \vee A_1}$$

Cut reductions and the cut elimination theorem can be adapted to intuitionistic logic with no major difference.

Theorem 4. *The intuitionistic sequent calculus enjoys cut elimination.*

Proof. We can adapt the previous proof, we just have to make sure that we are never forced to escape the intuitionistic fragment. □

Exercise 9. *Try to derive the two directions of each de Morgan duality. Derive $A \supset A$, show that $\neg A \vee A$ is not derivable. What does it means for the classical decomposition of $A \supset B$ into $\neg A \vee B$?*

In the rest of the course, we shall restrict to intuitionistic logic for simplicity, although most of what we say is valid in classical logic as well.

1.6 References

I recommend the book *Proofs and Types* by Girard, Taylor and Lafont, 1989[1]. It is a must read on the connection between typed λ-calculi and logic, and more advanced topics.

In honor of Gentzen's centenary, Jan von Plato wrote an essay on Gentzen's motivations and impact, titled *Gentzen's Proof Systems: Byproducts in a Work of Genius*[2].

Finally, Gentzen's 1934 paper is surprisingly readable. Its English translation, *Investigations into Logical Deduction*, can be found in the book *The Collected Papers of Gerhard Gentzen*, edited by M. E. Szabo at North Holland in 1969.

[1] `http://www.paultaylor.eu/stable/Proofs+Types.html`

[2] `http://www.math.ucla.edu/~asl/bsl/1803/1803-001.ps`

Chapter 2

A logic for reasoning about (co)inductive specifications

This chapter will be covered in a single lecture, so a lot of details will have to be skipped. The lecture notes are a bit rough; don't hesitate to contact me if you'd like to know how/where things are done cleanly in detail.

2.1 First-Order Logic

Definition 13. *In first-order logic, formulas express properties of objects denoted by* terms*. In general, there is a lot of flexibility regarding the particular choice of a language of terms (denoted by t, u, v). Terms should come with a notion of term variable (denoted by x, y, z) and a notion of substitution (replacing x by u in t is written $t[u/x]$) satisfying usual properties.*

Terms are usually first-order, meaning that they are made of constructors taking n terms and building a new one: for example, natural numbers are generated from the constructors 0 and s, 3 being written $s(s(s(0)))$; binary trees could generated from $leaf$ and $node$, allowing to write things like $node(leaf, node(leaf, leaf))$. Such terms can be multi-sorted, meaning that we separate terms into several families, such as natural numbers and trees, with constraints on what sort of term goes into a given position of a constructor.

Although we won't use it, terms can be many other things. For example, they could be arithmetic expressions considered up to computation of arithmetic operations, allowing to identify $s(x)+y$ and $s(x+y)$. They could also be higher-order, for example typed λ-calculus can be taken as our term language.

Definition 14. *Atoms are now predicates and take a number of terms as arguments. Formulas are extended with constructs for quantifies: if P is a formula then $\forall x.P$ and $\exists x.P$ are formulas. The variable x is* bound *by the quantification constructs.*

We shall always consider formulas up-to the (capture avoiding) renaming of variables that do not occur *free* (*i.e.*, only bound) in a formula. For instance, $\forall x.\ p(x,y)$ is the same as $\forall z.\ p(z,y)$ but not the same as $\forall y.\ p(y,y)$ (because this captures the free occurrence of y); finally, it is also not the same as $\forall x.\ p(x,z)$ because that is a renaming of the free variable y.

Definition 15 ($\mathbf{LK}^1$)**.** *Sequents for first-order classical logic are obtained by extending sequents with a* signature Σ *containing distinct first-order variables, including at least all variables occurring in the rest of the sequent. First-order sequents are written* $\Sigma\ ;\ \Gamma \vdash \Delta$.

In practice, we often omit to write the signature when it does not play an important role or when it is obvious. In particular, all rules from $\mathbf{LK}^0$ *are trivially adapted into* $\mathbf{LK}^1$ *that have no effect (nor constraints) on the signature. For instance, we have:*

$$\frac{\Sigma;\ \Gamma, A, B \vdash \Delta}{\Sigma;\ \Gamma, A \wedge B \vdash \Delta}$$

In addition to the (adapted) $\mathbf{LK}^0$ *rules, we add logical rules for the new connectives, that is the quantifiers:*

$$\frac{\Sigma; \Gamma, P[t/x] \vdash \Delta}{\Sigma; \Gamma, \forall x.P \vdash \Delta}\ \forall L \qquad \frac{\Sigma, x\ ;\ \Gamma \vdash P, \Delta}{\Sigma; \Gamma \vdash \forall x.P, \Delta}\ \forall R$$

$$\frac{\Sigma, x\ ;\ \Gamma, P \vdash \Delta}{\Sigma; \Gamma, \exists x.P \vdash \Delta}\ \exists L \qquad \frac{\Sigma; \Gamma \vdash P[t/x], \Delta}{\Sigma; \Gamma \vdash \exists x.P, \Delta}\ \exists R$$

Theorem 5. *First-order sequent calculus (both classical and intuitionistic) enjoys cut-elimination and is thus consistent.*

Proof. As an exercise, try to define cut reductions for the quantifiers. The key is to realize that if we have a derivation of $\Sigma, x;\ \Gamma \vdash \Delta$, then we can substitute x by any t everywhere in the derivation to obtain a derivation of $\Sigma;\ \Gamma[t/x] \vdash \Delta[t/x]$. We shall admit that the extended system of reduction rules terminate. □

Exercise 10. *The drinker paradox: in any (non-empty) bar, there is a person such that if this person drinks, then everybody in the bar drinks. The proof is non-constructive, this is why this fact is a bit counter-intuitive. Find an informal proof, then do it in* $\boldsymbol{LK}^1$, *where it is formalized as* $\exists x.\ d(x) \supset \forall y.\ d(y)$ *— the bar is implicit in the formalization, terms denote persons in the bar, and* $d(x)$ *stands for "x drinks". To reflect the non-emptiness of the bar, we can assume that we have one constant c, a term denoting one person in the bar.*

2.2 Equality

Although first-order logic is standard and widely understood, equality is less frequently considered within proof-theory, and its treatment is subtle. We detail here our approach to equality, which dates back to proposals by Girard and then Schroeder-Heister in the early 1990. Historically, this notion of equality is a byproduct of the introduction of fixed points. But it gains to be introduced separately. We should point out that there are other approaches to equality, for example Leibniz' equality is most common in higher-order logics.

What is clear about equality is its right rule: reflexivity. But there is no clear cut for the design of the left rule. We shall consider the following rules:

$$\frac{\{\Sigma\theta; \Gamma\theta \vdash G\theta : u\theta = v\theta\}}{\Sigma; \Gamma, u = v \vdash G} =L \qquad \frac{}{\Sigma; \Gamma \vdash u = u} =R$$

The left rule has one premise for each unifier θ of $u \doteq v$. The application of θ to terms is standard and naturally extended to formulas and to the left hand-side of the sequent. Its application to the signature ($\Sigma\theta$) denotes the signature obtained by removing from Σ the variables that are in the domain of θ, and adding those that are in its range.

In a sense, the left rule is the naive dual of the right one: it enumerates all cases for which the right one might have been proved. Indeed, this design supports cut-elimination. The principal case only consists in permuting $=R$ below the cut. The interesting phenomenon occur when reducing a cut on first-order quantifiers: this results in the instantiation of the universal variable by the witness of the existential quantification. That instantiation has to be performed in a subderivation, preserving its validity — this is a common simple result. The new case here is $=L$: as a variable gets instantiated, some unifiers might be simply updated, but others might disappear. If the equality eventually becomes absurd, the corresponding instance of $=L$ has no more premise. It is also easy to expand the axiom on equality, and along the same lines we obtain commutativity and canonicity of equality:

$$\frac{\ldots \quad \dfrac{}{\vdash u\theta = v\theta} =R \quad \ldots}{u = v \vdash u = v} =L$$

Infinitary rules are often convenient, but can be rightfully criticized. Indeed, a proof should always be finitely presentable, so that its validity can be decided. It is also a practical issue that proofs can be built in a finite amount of time. Hence, we usually consider a specialized version of the left rule, relying on a *complete set of unifiers* (*csu*), *i.e.*, a set S of unifiers such that all unifiers of $u \doteq v$ are specializations $\theta\theta'$ of some $\theta \in S$:

$$\frac{\{\Sigma\theta; \Gamma\theta \vdash G\theta : \theta \in csu(u \doteq v\theta)\}}{\Sigma; \Gamma, u = v \vdash G}$$

That rule is equivalent to the previous one: in one direction it is because the complete set of unifiers is a subset of all unifiers, in the other because the difference between the two can be obtained by specializing substitutions, and proofs accordingly. In the case of first-order terms, the *csu* can in fact be a most general unifier. However, that rule is still not effective in general in the case of higher-order terms; in practice it can often be managed by using higher-order pattern unification — a procedure for computing most general unifiers in a fragment of typed λ-calculus.

Example 3. *The csu-based rule is natural to work with, as it only requires the essential information. For example, with first-order terms:*

$$\dfrac{\dfrac{}{x; Px \vdash Px}}{x, y; x = y, Px \vdash Py}$$

With higher-order terms, in the higher-order pattern fragment:

$$\dfrac{\dfrac{\vdots}{z; \vdash \exists z'.\ (\lambda a.\ z) = (\lambda a.\ z') \wedge (\lambda b.\ z) = (\lambda b.\ z')}}{x, y; (\lambda a \lambda b.\ x\ a) = (\lambda a \lambda b.\ y\ b) \vdash \exists z'.\ x = (\lambda a.\ z') \wedge y = (\lambda b.\ z')}$$

As is clear from these examples, universal variables are not constants. Hence, we avoid to call them eigenvariables as Gentzen did.

2.3 Fixed Points

The content for this section comes from my thesis, A linear approach to the proof theory of least and greatest fixed points*, available online*[1]*. You may refer to it for more details, related work, or if you're missing some context.*

We present the logic μLJ, our intuitionistic system of reference supporting least and greatest fixed points. Its language of formulas is extended not only with the connective μ, now representing *least* fixed points, but also ν, of the same type, representing greatest fixed points. The rules of μLJ are presented in Figure 2.1. We present and discuss below the treatment of fixed points, starting with least fixed points.

2.3.1 Least fixed points

In proof-theory, least fixed points are characterized by the ability to reason about them by *induction.* It is interesting to justify that characterization, and its formalization in μLJ, from other presentations of least fixed points. It also shows that we are considering a natural notion, and not an exotic connective or an ad-hoc increment of expressiveness.

Definition 16 (Fixed, prefixed and postfixed point)**.** *Let ϕ be a mapping from sets to sets*[2]*. The set S is said to be a* fixed point *of ϕ when $\phi(S) = S$; a* prefixed point *of ϕ when $\phi(S) \subseteq S$; a* postfixed point *of ϕ when $S \subseteq \phi(S)$.*

Example 4. *The predicate operator B_{nat} can be read as the following function:*

$$N \mapsto \{0\} \cup \{s\ y : y \in N\}$$

[1] `http://www.lix.polytechnique.fr/d̃baelde/thesis/`

[2] One can more generally consider a mapping on a complete lattice, but we seek the most intuitive presentation.

Propositional intuitionistic logic

$$\frac{}{\Sigma; \Gamma, \bot \vdash P} \qquad \frac{}{\Sigma; \Gamma \vdash \top}$$

$$\frac{\Sigma; \Gamma, P, P' \vdash Q}{\Sigma; \Gamma, P \wedge P' \vdash Q} \qquad \frac{\Sigma; \Gamma \vdash P \quad \Sigma; \Gamma \vdash Q}{\Sigma; \Gamma \vdash P \wedge Q}$$

$$\frac{\Sigma; \Gamma, P_0 \vdash Q \quad \Sigma; \Gamma, P_1 \vdash Q}{\Sigma; \Gamma, P_0 \vee P_1 \vdash Q} \qquad \frac{\Sigma; \Gamma \vdash P_i}{\Sigma; \Gamma \vdash P_0 \vee P_1}$$

$$\frac{\Sigma; \Gamma \vdash P \quad \Sigma; \Gamma, P' \vdash Q}{\Sigma; \Gamma, P \supset P' \vdash Q} \qquad \frac{\Sigma; \Gamma, P \vdash Q}{\Sigma; \Gamma \vdash P \supset Q}$$

First-order structure

$$\frac{\Sigma, x; \Gamma, P \vdash Q}{\Sigma; \Gamma, \exists x.P \vdash Q} \qquad \frac{\Sigma; \Gamma \vdash P[t/x]}{\Sigma; \Gamma \vdash \exists x.P}$$

$$\frac{\Sigma; \Gamma, P[t/x] \vdash Q}{\Sigma; \Gamma, \forall x.P \vdash Q} \qquad \frac{\Sigma, x; \Gamma \vdash P}{\Sigma; \Gamma \vdash \forall x.P}$$

$$\frac{\{(\Sigma; \Gamma \vdash Q)\theta : t\theta \doteq t'\theta\}}{\Sigma; \Gamma, t = t' \vdash Q} \qquad \frac{}{\Sigma; \Gamma \vdash t = t}$$

Fixed points

$$\frac{\Sigma; \Gamma, St \vdash P \quad x; BSx \vdash Sx}{\Sigma; \Gamma, \mu Bt \vdash P} \qquad \frac{\Sigma; \Gamma \vdash B(\mu B)t}{\Sigma; \Gamma \vdash \mu Bt}$$

$$\frac{\Sigma; \Gamma, B(\nu B)t \vdash P}{\Sigma; \Gamma, \nu Bt \vdash P} \qquad \frac{\Sigma; \Gamma \vdash St \quad x; Sx \vdash BSx}{\Sigma; \Gamma \vdash \nu Bt}$$

Identity group

$$\frac{}{\Sigma; P \vdash P} \frac{\Sigma; \Gamma \vdash Q \quad \Sigma; \Gamma', Q \vdash P}{\Sigma; \Gamma, \Gamma' \vdash P}$$

Figure 2.1: Inference rules for μLJ(structural rules missing)

Its prefixed points contain zero and are stable by successor. Its postfixed points do not necessarily contain zero, but each of their elements is either zero or the successor of another. It admits a least fixed point, obtained by iterating from the empty set: it is the usual set of natural numbers. The greatest fixed point, assuming that there exists objects x which are not natural numbers, would contain them as well as their successors $s^n x$; assuming infinite terms, the greatest fixed point would also contain the infinite chain of successors.

Theorem 6 (Knaster-Tarski)**.** *Let ϕ be a monotonic function, then ϕ has a least fixed point, which is the intersection of all its prefixed points.*

The Knaster-Tarski theorem gives us an induction rule, along the common interpretation of implication as an inclusion:

$$\text{“If } B(S) \subseteq S \text{ and } t \in \mu B \text{ then } t \in S\text{.”} \qquad \frac{x; BSx \vdash Sx}{\Sigma; \mu Bt \vdash St}$$

The monotonicity condition of the Knaster-Tarski theorem, ensuring the existence of a (least) fixed point, translates in μLJ to the constraint that fixed point bodies are monotonic.

Example 5. *In the particular case of nat, the above induction rule yields the usual induction principle:*

$$\cfrac{\cfrac{\vdash P\ 0 \quad Py \vdash P(s\ y)}{\hline (B_{nat}P)x \vdash Px}\ \vee L, \exists L, =L}{nat\ x \vdash Px}$$

The problem with the considered induction rule is that it does not satisfy cut-elimination. We shall see that there is a way to reduce a cut between derivations of $\mu B \vdash S$ and $\vdash \mu B$, obtaining a derivation of the invariant S. But it is impossible to reduce a cut between derivations of $\mu B \vdash S$ and $S \vdash P$, until the invariant becomes active in the former derivation. In other words, the reduction of a cut on the invariant has to be postponed until a cut is reduced on the associated least fixed point. To express this, we consider in μLJ the following left rule for μ, which aggregates the former induction rule with a cut on the invariant, thereby restoring cut-eliminability:

$$\frac{\Sigma; \Gamma, St \vdash P \quad x; BSx \vdash Sx}{\Sigma; \Gamma, \mu Bt \vdash P}$$

As shown in Figure 2.1, the right rule for μ is unchanged, and the axiom on least fixed points is necessary. There is no need to consider a left unfolding rule for μ. Indeed, induction can emulate unfolding in the case of a monotonic fixed point B, by picking the invariant $B(\mu B)$.

2.3.2 Cut-elimination

We have introduced the proof-theoretical treatment of least fixed points in μLJ, by means of semantic intuitions. Although these intuitions are useful, and connections certainly exists, they do not need to be formally established to validate the design of μLJ: the syntactic, internal process of cut-elimination suffices. And a presentation of some of its key points should help understanding the logic.

The principal cut reduction for least fixed points is based on the transformation of a derivation of μB into a derivation of one of its invariants. Given a formula S and a proof Θ of $\forall x.\ BSx \supset Sx$, one can transform a derivation of $\Gamma \vdash \mu Bt$ into one of $\Gamma \vdash St$. This is done by induction on the derivation of the least fixed point, along the following scheme:

$$\begin{array}{ccc}
\dfrac{\dfrac{\dfrac{\vdots}{\Gamma' \vdash B(\mu B)\vec{t'}}}{\Gamma' \vdash \mu B\vec{t'}}}{} & & \dfrac{\dfrac{\dfrac{\vdots}{\Gamma' \vdash BS\vec{t'}} \quad \dfrac{\Theta(\vec{t'})}{BS\vec{t'} \vdash S\vec{t'}}}{\Gamma' \vdash S\vec{t'}}}{} \\
\dfrac{\dfrac{\vdots}{\Gamma \vdash B(\mu B)t}}{\Gamma \vdash \mu Bt} & \rightarrow & \dfrac{\dfrac{\vdots}{\Gamma \vdash BSt} \qquad \dfrac{\Theta(t)}{BSt \vdash St}}{\Gamma \vdash St}
\end{array}$$

The big steps here, represented by dots, consist in traversing the structure of B. For doing so it is crucial that B is positive. If it is strictly positive, then there is no recursive occurrence of μB that will ever occur on the left. Otherwise, some instances may occur on the left, but negatively. In any case these sub-formulas will only occur at toplevel on the right hand-side of the sequent, and can thus only be active in the right unfolding rule. This is essential, as it is the only thing that S can simulate. So, unlike the cut reduction for the self-dual μ, the reduction associated to least fixed point does rely on monotonicity.

It is in fact possible to refine the transformation, and obtain a more precise constraint on fixed points. The traversal of B can be expressed as a functoriality property, *i.e.,* the following rule:

$$\frac{x; Px \vdash Qx}{\Sigma; BP \vdash BQ}\ functo$$

Assuming only the functoriality of B, we can fully formulate the reduction of a principal cut on least fixed points:

$$\dfrac{\dfrac{\dfrac{\Pi}{\Gamma \vdash B(\mu B)t}}{\Gamma \vdash \mu B t}\,\mu R \qquad \dfrac{\dfrac{\Pi'}{\Delta, St \vdash G} \quad \dfrac{\Theta}{BSx \vdash Sx}}{\Delta, \mu B t \vdash G}\,\mu L}{\Gamma, \Delta \vdash G}\,cut$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{\Pi}{\Gamma \vdash B(\mu B)t} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{Sx \vdash Sx} \quad \dfrac{\Theta}{BSy \vdash Sy}}{\mu Bx \vdash Sx}\,\mu L}{B(\mu B)t \vdash BSt}\,functo \quad \dfrac{\Theta(t)}{BSt \vdash St}}{B(\mu B)t \vdash St}\,cut}{\Gamma \vdash St}\,cut \qquad \dfrac{\Pi'}{\Delta, St \vdash G}}{\Gamma, \Delta \vdash G}\,cut$$

Identifying the functoriality property allows for an elegant presentation of the rule, and has proved to help structuring normalization proofs. We do not show a normalization proof for μLJ in this thesis. For cut-elimination proof in sequent calculus, we refer the reader to the work on LINC [MT03, Tiu04], which is closely related to μLJ.

2.3.3 Greatest fixed points

In many settings, least and greatest fixed points are duals of each other: in a complete lattice, reversing the order swaps least and greatest fixed points; this amounts to consider complements in set theory, in other words the complement of a least fixed point is given by the greatest fixed point of the dual operator; this is also observed in category theory [CS02]. Unsurprisingly, the treatment of greatest fixed points in μLJ is obtained by dualizing the rules for least fixed points. The same observations can be made: admissibility of the right unfolding, cut reductions, etc.

2.4 Examples

Formalize and prove basic arithmetic facts: every even number is a natural number, every natural number is even or odd, (truncated) division by two is total.

The most common example of coinductive definition is simulation: we say that a process P simulates Q if whenever Q takes a step α towards some Q' then $P \overset{\alpha}{\to} P'$ and P' simulates Q'. To make this (fixed point) definition mean the "right" thing for infinite processes, we take the greatest fixed point. This can be formalized as a coinductive definition. Show that it is reflexive and transitive. Show that a process looping on one state with a transition α can be simulated by a process having two loops, one with α and one with β.

2.5 Proof of normalization

If time allows, we will outline how we use a fixed point construction to define $[\mu Bt]$ and $[\nu Bt]$, and use the Knaster-Tarski theorem to extend the previous proof of normalization in the case of fixed point operators.

2.6 References

For full normalization proofs of calculi involving fixed points and equality as presented above, see my papers[3] *Least and greatest fixed points in linear logic*, in ACM Transactions on Computational Logic, 2012, and *Combining Deduction Modulo and Logics of Fixed Point Definitions* in the proceedings of LICS 2012. The first paper deals with a sequent calculus for classical linear logic and the second with a natural deduction (λ-calculus) presentation of intuitionistic logic; both calculi feature first-order quantifiers, equality and fixed points.

One great way to learn those logics is to use them: you can have a look at the tools Bedwyr (`http://slimmer.gforge.inria.fr/bedwyr/`) and Abella (`http://abella-prover.org/`), which work on the logic we've seen, plus just one extension, namely the ∇ quantifier for introducing fresh/generic objects. The two websites have examples and docs to get started.

[3] Available on `http://www.lsv.ens-cachan.fr/b̃aelde/mes_publis.php`

Bibliography

[CS02] J. Robin B. Cockett and Luigi Santocanale. Induction, coinduction, and adjoints. *Electr. Notes Theor. Comput. Sci.*, 69, 2002.

[MT03] Alberto Momigliano and Alwen Tiu. Induction and co-induction in sequent calculus. In Mario Coppo, Stefano Berardi, and Ferruccio Damiani, editors, *Post-proceedings of TYPES 2003*, number 3085 in LNCS, pages 293–308, January 2003.

[Tiu04] Alwen Tiu. *A Logical Framework for Reasoning about Logical Specifications*. PhD thesis, Pennsylvania State University, May 2004.