Proof terms for generalized natural deduction

Herman Geuvers[1] and Tonny Hurkens[2]

1 **Radboud University & Technical University Eindhoven, The Netherlands**
herman@cs.ru.nl

Abstract

In previous work we have shown how to generate natural deduction rules for propositional connectives from truth tables, both for classical and constructive logic. In the present work we extend this work for the constructive case with proof-terms, thereby extending the Curry-Howard isomorphism to these new connectives. We define generalized notions of cut-elimination as term-reductions and we show how the well-known rules for natural deduction (Gentzen, Prawitz) and generalized elimination rules (Schroeder-Heister, von Plato and others) can be found as instances. In more technical terms: we define the notions of direct cut, an introduction rule followed immediately by an elimination rule, and of indirect cut, an elimination rule nested inside an elimination rule. Both are defined in the very general setting and we prove that all cuts can be removed from a derivation. As a consequence we derive weak normalization of cut-elimination for the well-known sets of rules of intuitionistic proposition logic.

Keywords and phrases constructive logic, natural deduction, cut-elimination, Curry-Howard isomorphism

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Natural deduction rules come in various forms, where the tree format is the most well-known. One either puts formulas A as the nodes and leaves of the tree, or sequents $\Gamma \vdash A$, where Γ is a sequence or a finite set of formulas. Other formalisms use a linear format, using flags or boxes to explicitly manage the open and discharged assumptions.

We use a tree format with sequents in the nodes and leaves, where all rules have a special form:

$$\frac{\dots \quad \Gamma \vdash A_i \quad \dots \qquad \dots \quad \Gamma, A_j \vdash D \quad \dots}{\Gamma \vdash D}$$

So if the conclusion of a rule is $\Gamma \vdash D$, then the hypotheses of the rule can be of one of two forms:

1. $\Gamma, A_j \vdash D$: we still need to prove D from Γ, but we are now also allowed to use A_j as additional assumption. We call A_j a casus.
2. $\Gamma \vdash A_i$: in stead of proving D from Γ, we now need to prove A_i from Γ. We call A_i a lemma.

One obvious advantage is that we don't have to give the Γ explicitly, as it can be retrieved from the other information in a deduction. So, we will present the deduction rules without the Γ in the format

$$\frac{\dots \quad \vdash A_i \quad \dots \qquad \dots \quad A_j \vdash D \quad \dots}{\vdash D}$$

In [7] we have shown how to give natural deduction rules for every connective that is defined by a truth table, both for the classical and the intuitionistic case. In that paper, we have shown that the intuitionistic rules are constructive indeed by providing a Kripke

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

semantics. In the present paper we provide a proof-theoretic study of the natural deduction rules for the intuitionistic case. We define cut-elimination for the general connectives, which we analyze by interpreting deductions as proof-terms. So we extend the Curry-Howard isomorphism, that interprets formulas as types and deductions as terms, to include all these new intuitionistic connectives.

It turns out that our standard format for the deduction rules we have chosen (as described above) is very suitable for defining cut-elimination in general, for giving a term interpretation to deductions and for defining a reduction on these proof-terms that corresponds with cut-elimination. The format of our rules also allows the transformation of other formalisms, like the very well-known ones by [6, 14] but also more recent ones by [22], in terms of ours. This transformation we will define on the proof-term level and we will show how the elimination of a *direct cut* (that is: an introduction rule immediately followed by an elimination rule) is preserved by the translation.

Standard questions about logic are consistency and decidability. We prove that both hold (in general for our connectives) by proving *weak normalization* for the process of eliminating direct cuts and *indirect cuts*. An indirect cut arises when an elimination rule blocks a direct cut for another connective; in that case one has to permute one elimination rule over another. Weak normalization states that for any deduction (proof term) we can eliminate cuts in such a way that eventually not cuts are left. Using this one can prove the sub-formula property and consistency and decidability.

The interest does not lie in the fact that our logic is consistent and decidable, but in the fact that the natural deduction rules can be defined and analyzed in such a generic way, capturing very many known instances of natural deduction rules for intuitionistic logic. The key concepts that make this work are our general rule format (described above) and the fact that our system provides natural deduction rules for each connective *in isolation*: rules for one connective do not use another connective. (This is true in general, also for the classical case, see [7].)

1.1 Definitions and Examples

For every connective we have elimination rules and introduction rules, where the introduction rules come in a intuitionistic and a classical variant. In [7] we also give the classical introduction rule, but in the present paper we restrict to the intuitionistic rules. The elimination rules have the following form. Φ is the formula we eliminate. We have $\Phi = c(A_1, \ldots, A_n, B_1, \ldots, B_m)$ where c is a connective of arity $n+m$. The formula D is arbitrary.

$$\frac{\vdash \Phi \quad \vdash A_1 \quad \ldots \quad \vdash A_n \qquad B_1 \vdash D \quad \ldots \quad B_m \vdash D}{\vdash D}\,\mathrm{el}$$

The introduction rules have a classical and an intuitionistic form. In [7] we treat both of them, but in the present paper we focus on intuitionistic logic. The intuitionistic introduction rules have the following form. Again, $\Phi = c(A_1, \ldots, A_n, B_1, \ldots, B_m)$ where c is a connective of arity $n+m$.

$$\frac{\vdash A_1 \quad \ldots \quad \vdash A_n \qquad B_1 \vdash \Phi \quad \ldots \quad B_m \vdash \Phi}{\vdash \Phi}\,\mathrm{in}$$

We extract these rules from the truth table for the connective c, as described in the following Definition.

▶ **Definition 1.** Suppose we have an n-ary connective c with a truth table t_c (with 2^n rows). We write $\varphi = c(p_1, \ldots, p_n)$, where $p_1, \ldots, p_n$ are proposition letters and we write

$\Phi = c(A_1, \ldots, A_n)$, where $A_1, \ldots, A_n$ are arbitrary propositions. Each row of t_c gives rise to an elimination rule or an introduction rule for c in the following way.

$$\begin{array}{ccc|c} p_1 & \ldots & p_n & c(p_1,\ldots,p_n) \\ \hline a_1 & \ldots & a_n & 0 \end{array} \quad \mapsto \quad \frac{\vdash \Phi \quad \ldots \vdash A_j (\text{if } a_j = 1) \ldots \quad \ldots A_i \vdash D (\text{if } a_i = 0) \ldots}{\vdash D}\,\text{el}$$

$$\begin{array}{ccc|c} p_1 & \ldots & p_n & c(p_1,\ldots,p_n) \\ \hline b_1 & \ldots & b_n & 1 \end{array} \quad \mapsto \quad \frac{\ldots \vdash A_j (\text{if } b_j = 1) \ldots \quad \ldots A_i \vdash \Phi (\text{if } b_i = 0) \ldots}{\vdash \Phi}\,\text{in}$$

If $a_j = 1$ in t_c, then A_j occurs as a lemma in the rule; if $a_i = 0$ in t_c, then A_i occurs as a casus. The rules are given in abbreviated form and it should be understood that all judgments can be used with an extended hypotheses set Γ. So the elimination rule in full reads as follows (where Γ is a set of propositions).

$$\frac{\Gamma \vdash \Phi \quad \ldots \Gamma \vdash A_j \text{ (if } a_j = 1) \ldots \quad \ldots \Gamma, A_i \vdash D \text{ (if } a_i = 0) \ldots}{\Gamma \vdash D}\,\text{el}$$

In an elimination rule, we call $\vdash \Phi$ the *major premise* and the other hypotheses of the rule we call the *minor premises*.

▶ **Definition 2.** Given a set of connectives $\mathcal{C} := \{c_1, \ldots, c_n\}$, we define the *intuitionistic* natural deduction systems for $\mathcal{C}$, $\mathsf{IPC}_{\mathcal{C}}$, by the following derivation rules.

- The *axiom rule*

$$\frac{}{\Gamma \vdash A}\,\text{axiom(if } A \in \Gamma)$$

- The elimination rules for the connectives in $\mathcal{C}$ and the intuitionistic introduction rules for the connectives in $\mathcal{C}$, as defined in Definition 1.

We write $\Gamma \vdash A$ if $\Gamma \vdash A$ is derivable using the derivation rules of $\mathsf{IPC}_{\mathcal{C}}$.

▶ **Example 3.**

A	B	$A \vee B$	$A \wedge B$	$A \to B$	$\neg A$
0	0	0	0	1	1
0	1	1	1	1	1
1	0	1	1	0	0
1	1	1	1	1	0

1. From the truth table for $\vee$ we derive the following intuitionistic rules for $\vee$.

$$\frac{\vdash A \vee B \quad A \vdash D \quad B \vdash D}{\vdash D}\,\vee\text{-el} \qquad \frac{A \vdash A \vee B \quad \vdash B}{\vdash A \vee B}\,\vee\text{-in}_a$$

$$\frac{\vdash A \quad B \vdash A \vee B}{\vdash A \vee B}\,\vee\text{-in}_b \qquad \frac{\vdash A \quad \vdash B}{\vdash A \vee B}\,\vee\text{-in}_c$$

These rules are all intuitionistically correct, as one can observe by inspection. We will show that these are equivalent to the well-known intuitionistic rules. We will also show how these rules can be optimized and be reduced to 1 elimination rule and 2 introduction rules, which are the well-known ones.

2. From the truth table for $\wedge$ we derive the following intuitionistic rules for $\wedge$, 3 elimination rules and one introduction rule.

$$\frac{\vdash A \wedge B \quad A \vdash D \quad B \vdash D}{\vdash D}\,\wedge\text{-el}_a \qquad \frac{\vdash A \wedge B \quad A \vdash D \quad \vdash B}{\vdash D}\,\wedge\text{-el}_b$$

$$\frac{\vdash A \wedge B \quad \vdash A \quad B \vdash D}{\vdash D}\,\wedge\text{-el}_c \qquad \frac{\vdash A \quad \vdash B}{\vdash A \wedge B}\,\wedge\text{-in}$$

These rules are all intuitionistically correct, as one can observe by inspection. We will show that these are equivalent to the well-known intuitionistic rules. We will also show how these rules can be optimized and be reduced to 2 elimination rules and 1 introduction rule, which are the well-known ones. The elimination rules for $\wedge$ have a bit the flavor of the so called "generalized elimination rules" of Schroeder-Heister [17] and Von Plato [22], in the sense that we don't derive A, respectively B, from $A \wedge B$, but an auxiliary conclusion D is derived. This rule, also called the *parallel elimination rule* by Tennant [20], is as follows.

$$\frac{\vdash A \wedge B \qquad A, B \vdash D}{\vdash D}\ \wedge\text{-el}^{\text{par}}$$

We will show that this rule can be derived from ours.

3. From the truth table for $\neg$ we also derive the following rules for $\neg$, one elimination rule and one introduction rule.

$$\frac{\vdash \neg A \qquad \vdash A}{\vdash D}\ \neg\text{-el} \qquad \frac{A \vdash \neg A}{\vdash \neg A}\ \neg\text{-in}$$

The elimination rule is familiar. For the introduction rule: to prove $\neg A$, one "only" has to prove $\neg A$ from A, which may seem limited. The traditional $\neg$-in rule is the following.

$$\frac{A \vdash \neg B \qquad A \vdash \neg B}{\vdash \neg A}\ \neg\text{-in}^{\text{t}}$$

The two *neg*-introduction rules are equivalent, which we will show in detail (using proof terms) in Lemma 31). To derive $\neg$-in$^{\text{t}}$ from $\neg$-in one also needs $\neg$-el, so we view $\neg$-in as more primitive then the traditional rule $\neg$-in$^{\text{t}}$.
As an example of the intuitionistic derivation rules for $\neg$ we show that $A \vdash \neg\neg A$ is derivable:

$$\frac{\dfrac{A, \neg A \vdash \neg A \qquad A, \neg A \vdash A}{A, \neg A \vdash \neg\neg A}\ \neg\text{-el}}{A \vdash \neg\neg A}\ \neg\text{-in}$$

4. From the truth table for $\to$ we derive the following intuitionistic rules for $\to$.

$$\frac{A \vdash A \to B \qquad B \vdash A \to B}{\vdash A \to B}\ \to\text{-in}_a \qquad \frac{A \vdash A \to B \qquad \vdash B}{\vdash A \to B}\ \to\text{-in}_b$$

$$\frac{\vdash A \to B \qquad \vdash A \qquad B \vdash D}{\vdash D}\ \to\text{-el} \qquad \frac{\vdash A \qquad \vdash B}{\vdash A \to B}\ \to\text{-in}_c$$

These rules are all intuitionistically correct, as one can verify by inspection. For example, for $\to$-in$_a$, observe that if $A \vdash A \to B$, then $\vdash A \to B$, so the second hypothesis is superfluous. Similarly for $\to$-in$_b$, the first hypothesis is superfluous. We will show that these rules are equivalent to the well-known intuitionistic rules. We will also show how these rules can be optimized and be reduced to 1 elimination rule and 2 introduction rules. These are not the well-known ones, because the well-known $\to$-in-rule does not fit into our scheme:

$$\frac{A \vdash B}{\vdash A \to B}\ \to\text{-in}$$

In this rule, both the conclusion is changed *and* an assumption (casus) is added. In our system, each rule has the property that a hypothesis either adds an assumption or changes the conclusion (while retaining the same set of assumptions), and this "or" is exclusive.

1.2 Contribution of the paper and related work

Natural deduction has been studied extensively, since the original work by Gentzen [6], both for classical and intuitionistic logic. Overviews can be found in [21] and [12]. Also the generalization of natural deduction to include other connectives or allow different derivation rules has been studied by various researchers. Notably, there is the work of Schroeder-Heister [17], Read [16], Tennant [20], Von Plato [22, 12], Milne [11], Francez and Dyckhoff [4, 3] that is related to ours. Schroeder-Heister studies general formats of natural deduction where also rules may be discharged (as opposed to the normal situation where only formulas may be discharged). He also studies a general rule format for intuitionistic logic and shows that the connectives $\wedge, \vee, \rightarrow, \bot$ are complete for it. Von Plato, Milne, Francez and Dyckhoff, Read and Tennant study "generalized elimination rules", where the idea is that elimination rules arise naturally from the introduction rules, following Prawitz's [15] inversion principle: "the conclusion obtained by an elimination does not state anything more than what must have already been obtained if the major premise of the elimination was inferred by an introduction". The elimination rules obtained have the same flavor as the elimination rules we derive from truth tables: the conclusion of elimination Φ is not a sub-formula of Φ, but a general formula D, where there are additional hypothesis that connect Φ and D. For the standard intuitionistic connectives the general elimination rules are quite close to ours, but $\wedge$-elimination is slightly different. Von Plato [22], Lopez-Escobar [10] and Tennant [20] study the standard intuitionistic connectives with generalized rules.

A difference is that we focus not so much on the rules but on the fact that we can define different and new connectives constructively. In our work, we do not take the introduction rules as primary, with the elimination rules defined from them, but we derive elimination and introduction rules directly from the truth table. Then we optimize them, which can be done in various ways, where we adhere to a fixed format for the rules. Many of the generalized elimination rules, for example for $\wedge$, appear naturally as a consequence of our approach. of deriving the rules from the truth table.

The idea of deriving deduction rules from the truth table also occurs in the work of Milne [11], but in a slightly different way: from the introduction rules, a truth table is derived and then the classical elimination rules are derived from the truth table. For the if-then-else connective, this amounts to rules equivalent to ours, but not optimized. We start from the truth table and derive rules for intuitionistic logic.

The basic results for our logic can be found in [7]. This includes a general Kripke semantics for the intuitionistic rules. The contribution of the present paper is a definition of direct and indirect cut for our rules, which applies generally to the intuitionistic rules for any set of connectives defined from the truth tables. Furthermore, we analyze the proof by giving a term interpretation for derivation, thus extending the Curry-Howard isomorphism to our generalized connectives. The term interpretation makes it easier to study cut-elimination as term-reduction. Using these proof-terms, we prove that eliminating indirect cuts is strongly normalizing, that is: any process of eliminating indirect cuts leads to a derivation without indirect cuts. Similarly we prove that eliminating direct cuts is strongly normalizing, by applying the well-known Tait method of saturated sets. Finally we prove that the combination of the two cut-eliminations is weakly normalizing: we give a procedure for eliminating all cuts from a derivation.

2 Simple properties and examples

Cut-elimination in natural deduction involves the placing of one derivation on top of another, replacing a leaf A of a derivation tree (an assumption) by a derivation of A. In our case, this amounts to plugging a derivation of $\Gamma \vdash A$ in place of an axiom $\Gamma, A \vdash A$. We first define precisely how the "plugging one derivation in another" works.

▶ **Lemma 4.** *If $\Gamma \vdash \varphi$ and $\Delta, \varphi \vdash \psi$, then $\Gamma, \Delta \vdash \psi$*

Proof. By induction on the derivation of $\Delta, \varphi \vdash \psi$, using the fact that, in general (for all Γ, Γ' and φ): If $\Gamma \vdash \varphi$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash \varphi$. ◀

Let's be a bit more precise about what happens with the derivations in the proof of Lemma 4: If Π is the derivation of $\Delta, \varphi \vdash \psi$, then, due to the format of our rules, the only place in Π where the hypothesis φ can be used is at a leaf of Π, in an instance of the (axiom) rule. Again, due to the format of our rules,these leaves are of the form $\Delta', \varphi \vdash \varphi$ for some $\Delta' \supseteq \Delta$. We replace this leaf with Σ, the derivation of $\Gamma \vdash \varphi$. This Σ is also a derivation of $\Gamma, \Delta' \vdash \varphi$, so Π with the leaves of the form $\Delta', \varphi \vdash \varphi$ replaced by Σ yields a correct derivation of $\Gamma, \Delta \vdash \psi$.

▶ Notation 5. If Σ is a derivation of $\Gamma \vdash \varphi$ and Π is a derivation of $\Delta, \varphi \vdash \psi$, then we have a derivation of $\Gamma, \Delta \vdash \psi$ that looks like this:

$$\begin{array}{ccc} \vdots\,\Sigma & & \vdots\,\Sigma \\ \Gamma \vdash \varphi & \ldots & \Gamma \vdash \varphi \\ & \vdots\,\Pi & \\ & \Delta \vdash \psi & \end{array}$$

So in Π, every application of an (axiom) rule at a leaf, deriving $\Delta' \vdash \varphi$ for some $\Delta' \supseteq \Delta$ is replaced by a copy of a derivation Σ, which is also a derivation of $\Delta', \Gamma \vdash \varphi$.

In Definitions 1 and 2, we have given the precise rules for our logic, in intuitionistic and classical format. We can freely reuse formulas and weaken the context, so the structural rules of contraction and weakening are wired into the system. In examples, to simplify derivations we will often use the following format for an elimination rule (and equivalently for an introduction rule).

$$\frac{\Gamma_0 \vdash \Phi \qquad \ldots \Gamma_j \vdash A_j \text{ (if } a_j = 1) \ldots \qquad \ldots \Gamma_i, A_i \vdash D \text{ (if } a_i = 0) \ldots}{\cup_{k=0}^{n} \Gamma_k \vdash D}\ \text{el}$$

To reduce the number of rules, we can take a number of rules together and drop one or more hypotheses. We show this by again looking at the example of the rules for $\wedge$ (Example 3).

▶ **Example 6.** From the truth table we have derived the following 3 intuitionistic elimination rules for $\wedge$.

$$\frac{\vdash A \wedge B \qquad A \vdash D \qquad B \vdash D}{\vdash D}\ \wedge\text{-el}_a \qquad \frac{\vdash A \wedge B \qquad A \vdash D \qquad \vdash B}{\vdash D}\ \wedge\text{-el}_b$$

$$\frac{\vdash A \wedge B \qquad \vdash A \qquad B \vdash D}{\vdash D}\ \wedge\text{-el}_c$$

These rules can be reduced to the following 2 equivalent elimination rules.

$$\frac{\vdash A \wedge B \qquad A \vdash D}{\vdash D}\wedge\text{-el}'_1 \qquad \frac{\vdash A \wedge B \qquad B \vdash D}{\vdash D}\wedge\text{-el}'_2$$

It can be shown that these sets of rules are equivalent Here we only show the derivability of the $\wedge$-el$'_1$ rule from the rules $\wedge$-el$_a$ and $\wedge$-el$_b$. Suppose we have derivations of $\Gamma \vdash A \wedge B$ and of $\Gamma, A \vdash D$. Then we have the following derivation.

$$\frac{\Gamma \vdash A \wedge B \qquad \Gamma, A \vdash D \qquad \dfrac{\Gamma, B \vdash A \wedge B \qquad \Gamma, B, A \vdash D \qquad \Gamma, B \vdash B}{\Gamma, B \vdash D}}{\Gamma \vdash D}$$

The general method here is that we can replace two rules that only differ in one hypothesis, which in one rule occurs as a lemma and in the other as a casus, by one rule where the hypothesis is removed. It will be clear that the Γ's above are not relevant for the argument, so we will not write these.

▶ **Lemma 7.** *A system with two derivation rules of the form*

$$\frac{\vdash A_1 \ \ldots \ \vdash A_n \qquad B_1 \vdash D \ \ldots \ B_m \vdash D \qquad A \vdash D}{\vdash D} \qquad \frac{\vdash A_1 \ \ldots \ \vdash A_n \qquad \vdash A \qquad B_1 \vdash D \ \ldots \ B_m \vdash D}{\vdash D}$$

is equivalent to the system with these two rules replaced by

$$\frac{\vdash A_1 \ \ldots \ \vdash A_n \qquad B_1 \vdash D \ \ldots \ B_m \vdash D}{\vdash D}$$

Proof. The implication from bottom to top is immediate. From top to bottom, suppose we have the two given rules. We now derive the bottom one. Assume we have derivations of $\vdash A_1, \ldots, \vdash A_n, B_1 \vdash D, \ldots, B_m \vdash D$. We now have the following derivation of $\vdash D$.

$$\frac{\vdash A_1 \ \ldots \ \vdash A_n \qquad B_1 \vdash D \ \ldots \ B_m \vdash D \qquad \dfrac{A \vdash A_1 \ \ldots \ A \vdash A_n \qquad A \vdash A \qquad A, B_1 \vdash D \ \ldots \ A, B_m \vdash D}{A \vdash D}}{\vdash D}$$

◀

Lemma 7 can be applied to elimination and introduction rules. An application to elimination rules is given in Example 6. We now give two applications to introduction rules.

▶ **Example 8.** From the truth table we have derived the following 3 intuitionistic introduction rules for $\vee$.

$$\frac{A \vdash A \vee B \qquad \vdash B}{\vdash A \vee B}\vee\text{-in}_a \qquad \frac{\vdash A \qquad B \vdash A \vee B}{\vdash A \vee B}\vee\text{-in}_b \qquad \frac{\vdash A \qquad \vdash B}{\vdash A \vee B}\vee\text{-in}_c$$

Using Lemma 7, these rules can be reduced to the following 2 equivalent introduction rules.

$$\frac{\vdash A}{\vdash A \vee B}\vee\text{-in}_1 \qquad \frac{\vdash B}{\vdash A \vee B}\vee\text{-in}_2$$

▶ **Example 9.** Similar to $\vee$, we can optimize the introduction rules for $\to$. From the truth table we have derived the following 3 intuitionistic introduction rules for $\to$.

$$\frac{A \vdash A \to B \qquad B \vdash A \to B}{\vdash A \to B} \to\text{-in}_a \qquad \frac{A \vdash A \to B \qquad \vdash B}{\vdash A \to B} \to\text{-in}_b \qquad \frac{\vdash A \qquad \vdash B}{\vdash A \to B} \to\text{-in}_c$$

Using Lemma 7, these rules can be reduced to the following 2 equivalent introduction rules.

$$\frac{A \vdash A \to B}{\vdash A \to B} \to\text{-in}_1 \qquad \frac{\vdash B}{\vdash A \to B} \to\text{-in}_2$$

It can easily be shown that, in presence rules $\to$-in$_1$ and $\to$-in$_2$ together are equivalent with the well-known $\to$-in:

$$\frac{A \vdash B}{\vdash A \to B} \to\text{-in}$$

NB. To derive $\to$-in$_1$ from $\to$-in, on also needs $\to$-el.

As $\to$-in does not conform with our format for rules, we will be using $\to$-in$_1$ and $\to$-in$_2$ as our basic rules and treat $\to$-in as a defined rule, the composition of first $\to$-in$_2$ and then $\to$-in$_1$.

Another optimization we can perform is to replace a rule which has only one casus by a rule where the casus is the conclusion. To illustrate this: the simplified elimination rules for $\wedge$, $\wedge$-el$'_1$ and $\wedge$-el$'_2$ have only one casus. The rule $\wedge$-el$'_1$ (left) can thus be replaced by the rule $\wedge$-el$_1$ (right), which is the usual projection rule.

$$\frac{\vdash A \wedge B \qquad A \vdash D}{\vdash D} \wedge\text{-el}'_1 \qquad \frac{\vdash A \wedge B}{\vdash A} \wedge\text{-el}_1$$

There is a general Lemma stating this simplification is correct.

▶ **Lemma 10.** *A system with a derivation rule of the form to the left is equivalent to the system with this rule replaced by the rule on the right.*

$$\frac{\vdash A_1 \ \ldots \ \vdash A_n \qquad B \vdash D}{\vdash D} \qquad\qquad \frac{\vdash A_1 \ \ldots \ \vdash A_n}{\vdash B}$$

Proof. The implication from left to right is immediate. From right to left, assume we have derivations of $\vdash A_1, \ldots, \vdash A_n$. Then, by the rule to the right, we have $\Gamma \vdash B$. Now assume we also have a derivation of $B \vdash D$. By Lemma 4, we also have a derivation of $\Gamma \vdash D$. ◀

▶ **Definition 11.** The *standard derivation rules* for the intuitionistic propositional connectives $\wedge, \vee, \to, \neg, \perp$ and $\top$ are given below. These rules are derived from the truth tables and optimized following Lemmas 7 and 10. We have seen most of the rules in previous Examples, except for the rules for $\top$ and $\perp$, which are derived immediately from Definition 1.

$$\frac{\vdash A \qquad \vdash B}{\vdash A \wedge B}\ \wedge\text{-in} \qquad \frac{\vdash A \wedge B}{\vdash A}\ \wedge\text{-el}_1 \qquad \frac{\vdash A \wedge B}{\vdash B}\ \wedge\text{-el}_2$$

$$\frac{\vdash A}{\vdash A \vee B}\ \vee\text{-in}_1 \qquad \frac{\vdash B}{\vdash A \vee B}\ \vee\text{-in}_2 \qquad \frac{\vdash A \vee B \quad A \vdash D \quad B \vdash D}{\vdash D}\ \vee\text{-el}$$

$$\frac{\vdash B}{\vdash A \to B}\ \to\text{-in}_1 \qquad \frac{A \vdash A \to B}{\vdash A \to B}\ \to\text{-in}_2 \qquad \frac{\vdash A \to B \quad \vdash A}{\vdash B}\ \to\text{-el}$$

$$\frac{A \vdash \neg A}{\vdash \neg A}\ \neg\text{-in} \qquad \frac{\vdash \neg A \qquad \vdash A}{\vdash D}\ \neg\text{-el} \qquad \frac{}{\vdash \top}\ \top\text{-in} \qquad \frac{\vdash \bot}{\vdash D}\ \bot\text{-el}$$

2.1 Two larger examples

We now look at if-then-else, the ternary "if-then-else" connective, and most, the ternary connective that is true if at least 2 of the arguments are true. The truth tables of most and if-then-else are as follows, where we denote if A then B else C by $A{\to}B/C$.

A	B	C	most(A, B, C)	$A{\to}B/C$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

From the lines in the truth table of $A{\to}B/C$ with a 0 we get the following four elimination rules.

$$\frac{\vdash A{\to}B/C \quad A \vdash D \quad B \vdash D \quad C \vdash D}{\vdash D} \qquad \frac{\vdash A{\to}B/C \quad A \vdash D \quad \vdash B \quad C \vdash D}{\vdash D}$$

$$\frac{\vdash A{\to}B/C \quad \vdash A \quad B \vdash D \quad C \vdash D}{\vdash D} \qquad \frac{\vdash A{\to}B/C \quad \vdash A \quad B \vdash D \quad \vdash C}{\vdash D}$$

Using Lemmas 7 and 10, these can be reduced to the following two. (The two rules on the first line reduce to else-el, the two rules on the second line reduce to then-el.)

$$\frac{\vdash A{\to}B/C \quad A \vdash D \quad C \vdash D}{\vdash D}\ \text{else-el} \qquad \frac{\vdash A{\to}B/C \quad \vdash A}{\vdash B}\ \text{then-el}$$

These are not the only possible optimizations: the two rules on the left can also be combined into an "if-el" rule:

$$\frac{\vdash A{\to}B/C \quad B \vdash D \quad C \vdash D}{\vdash D}\ \text{if-el}$$

From the lines in the truth table of $A{\to}B/C$ with a 1 we get the following four introduction rules:

$$\frac{A \vdash A{\to}B/C \quad B \vdash A{\to}B/C \quad \vdash C}{\vdash A{\to}B/C} \qquad \frac{A \vdash A{\to}B/C \quad \vdash B \quad \vdash C}{\vdash A{\to}B/C}$$

$$\frac{\vdash A \quad \vdash B \quad C \vdash A{\to}B/C}{\vdash A{\to}B/C} \qquad \frac{\vdash A \quad \vdash B \quad \vdash C}{\vdash A{\to}B/C}$$

Using Lemmas 7 and 10 can be reduced to the following two. (The two rules on the first line reduce to else-in, the two rules on the second line reduce to then-in.)

$$\frac{A \vdash A{\to}B/C \quad \vdash C}{\vdash A{\to}B/C}\ \text{else-in} \qquad \frac{\vdash A \quad \vdash B}{\vdash A{\to}B/C}\ \text{then-in}$$

Again, these are not the only possible optimizations: the two rules on the right can also be combined into an "if-in" rule:

$$\frac{\vdash B \quad \vdash C}{\vdash A{\to}B/C}\ \text{if-in}$$

In [7], we have studied the if-then-else connective in more detail, and we show that if-then-else, together with $\top$ and $\bot$ is *functionally complete*: all other constructive connectives can be defined in terms of it. We also prove a type of "disjunction property": if $\vdash A{\to}B/C$, then $\vdash A$ or $\vdash C$, and similarly, if $\vdash A{\to}B/C$, then $\vdash B$ or $\vdash C$.

From the lines in the truth table of $\mathsf{most}(A,B,C)$ with a 0 we get the following four elimination rules:

$$\frac{\vdash \mathsf{most}(A,B,C) \quad A \vdash D \quad B \vdash D \quad C \vdash D}{\vdash D} \qquad \frac{\vdash \mathsf{most}(A,B,C) \quad A \vdash D \quad B \vdash D \quad \vdash C}{\vdash D}$$

$$\frac{\vdash \mathsf{most}(A,B,C) \quad A \vdash D \quad \vdash B \quad C \vdash D}{\vdash D} \qquad \frac{\vdash \mathsf{most}(A,B,C) \quad \vdash A \quad B \vdash D \quad C \vdash D}{\vdash D}$$

Using Lemmas 7 and 10, these can be reduced to the following three.

$$\frac{\vdash \mathsf{most}(A,B,C) \quad A \vdash D \quad B \vdash D}{\vdash D}\ \mathsf{most}\text{-el}_1 \qquad \frac{\vdash \mathsf{most}(A,B,C) \quad A \vdash D \quad C \vdash D}{\vdash B}\ \mathsf{most}\text{-el}_2$$

$$\frac{\vdash \mathsf{most}(A,B,C) \quad B \vdash D \quad C \vdash D}{\vdash B}\ \mathsf{most}\text{-el}_3$$

From the lines in the truth table of $\mathsf{most}(A,B,C)$ with a 1 we get the following four introduction rules:

$$\frac{A \vdash \mathsf{most}(A,B,C) \quad \vdash B \quad \vdash C}{\vdash \mathsf{most}(A,B,C)} \qquad \frac{\vdash A \quad B \vdash \mathsf{most}(A,B,C) \quad \vdash C}{\vdash \mathsf{most}(A,B,C)}$$

$$\frac{\vdash A \quad \vdash B \quad C \vdash \mathsf{most}(A,B,C)}{\vdash \mathsf{most}(A,B,C)} \qquad \frac{\vdash A \quad \vdash B \quad \vdash C}{\vdash \mathsf{most}(A,B,C)}$$

Using Lemmas 7 and 10 can be reduced to the following three.

$$\frac{\vdash A \quad \vdash B}{\vdash \mathsf{most}(A,B,C)}\ \mathsf{most}\text{-}\mathrm{in}_1 \qquad \frac{\vdash A \quad \vdash C}{\vdash \mathsf{most}(A,B,C)}\ \mathsf{most}\text{-}\mathrm{in}_2 \qquad \frac{\vdash B \quad \vdash C}{\vdash \mathsf{most}(A,B,C)}\ \mathsf{most}\text{-}\mathrm{in}_3$$

In [7], it is shown that the connective most also satisfies a type of "disjunction property": if $\vdash \mathsf{most}(A,B,C)$, then $\vdash A$ or $\vdash B$ (and similarly, $\vdash A$ or $\vdash C$, and similarly $\vdash B$ or $\vdash C$).

3 Cuts and cut-elimination

The notion of *direct cut* has already been described in [7]: an introduction of Φ immediately followed by an elimination of Φ. In such case there is (referring back to the truth table, see Definition 1) at least one k for which $a_k \neq b_k$. In case $a_k = 0, b_k = 1$, we have a sub-derivation Σ of $\vdash A_k$ and a sub-derivation Θ of $A_k \vdash D$ and we can plug Σ on top of Θ to obtain a derivation of $\vdash D$. In case $a_k = 1, b_k = 0$, we have a sub-derivation Σ of $A_k \vdash \Phi$ and a sub-derivation Θ of $\vdash A_k$ and we can plug Θ on top of Σ to obtain a derivation of $\vdash \Phi$. This is then used as a hypothesis for the elimination rule (that remains in this case) instead of the original one that was a consequence of the introduction rule (that now disappears).

In general there are more k for which $a_k \neq b_k$, so the general cut-elimination procedure is non-deterministic. We view this non-determinism as a natural feature in natural deduction; the fact that for some connectives (or combination of connectives), cut-elimination is deterministic is an "emerging" property. We will show two examples of the non-determinism of cut-elimination later.

The introduction of a formula Φ immediately followed by an elimination of Φ we will call a *direct intuitionistic cut*. In general in between the intro rule for Φ and the elim rule for Φ, there may be other auxiliary rules, so occasionally we may have to first commute the elim rule with these auxiliary rules to obtain a direct cut that can be contracted. So, we will also define the notion of *indirect intuitionistic cut* or *commuting intuitionistic cut*.

▶ **Definition 12.** Let c be a connective of arity n, with an elim rule and an intuitionistic intro rule derived from the truth table, as in Definition 1. So suppose we have the following rules in the truth table t_c.

p_1	$\ldots$	p_n	$c(p_1,\ldots,p_n)$
a_1	$\ldots$	a_n	0
b_1	$\ldots$	b_n	1

An *intuitionistic direct cut* in a derivation is a pattern of the following form, where $\Phi = c(A_1,\ldots,A_n)$.

$$\dfrac{\dfrac{\ldots\quad \begin{matrix}\boxed{\Sigma_j}\\ \Gamma \vdash A_j\end{matrix} \quad\ldots\qquad \ldots\quad \begin{matrix}\boxed{\Sigma_i}\\ \Gamma, A_i \vdash \Phi\end{matrix}\quad \ldots}{\Gamma \vdash \Phi}\ \mathrm{in} \qquad \ldots\quad \begin{matrix}\boxed{\Pi_k}\\ \Gamma \vdash A_k\end{matrix}\quad \ldots \qquad \ldots\quad \begin{matrix}\boxed{\Pi_\ell}\\ \Gamma, A_\ell \vdash D\end{matrix}\quad \ldots}{\Gamma \vdash D}\ \mathrm{el}$$

- Here, in is an arbitrary introduction rule. In this rule, A_j ranges over all propositions where $b_j = 1$; A_i ranges over all propositions where $b_i = 0$,
- Here, el is an arbitrary elimination rule. In this rule, A_k ranges over all propositions where $a_k = 1$; A_ℓ over all propositions where $a_\ell = 0$,

The *elimination of a direct cut* is defined by replacing the derivation pattern above by

1. If $\ell = j$ for some ℓ, j (that is: $A_\ell = A_j$):

$$\begin{array}{c}
\begin{array}{ccc}
\vdots\ \boxed{\Sigma_j} & & \vdots\ \boxed{\Sigma_j} \\
\Gamma \vdash A_j & \dots & \Gamma \vdash A_j
\end{array} \\
\vdots\ \boxed{\Pi_\ell} \\
\Gamma \vdash D
\end{array}$$

2. If $k = i$ for some k, i (that is: $A_k = A_i$):

$$\dfrac{\begin{array}{c}
\begin{array}{ccc}
\vdots\ \boxed{\Pi_k} & & \vdots\ \boxed{\Pi_k} \\
\Gamma \vdash A_i & \dots & \Gamma \vdash A_i
\end{array} \\
\vdots\ \boxed{\Sigma_i} \\
\Gamma \vdash \Phi
\end{array}
\qquad \dots \quad \begin{array}{c}\vdots\ \boxed{\Pi_k} \\ \Gamma \vdash A_k\end{array} \quad \dots \quad \dots \quad \begin{array}{c}\vdots\ \boxed{\Pi_\ell} \\ \Gamma, A_\ell \vdash D\end{array} \quad \dots}{\Gamma \vdash D}\,\mathrm{el}$$

There may be several choices for the i and j in the previous definition, so cut-elimination is non-deterministic in general. We give an example of most to illustrate this. For simplicity, we use the optimized rules.

▶ **Example 13.** Consider the following direct cut for most.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\ \boxed{\Sigma_1} \\ \Gamma \vdash A\end{array} \qquad \begin{array}{c}\vdots\ \boxed{\Sigma_2} \\ \Gamma \vdash B\end{array}}{\Gamma \vdash \mathsf{most}(A, B, C)}\,\mathsf{most\text{-}in}_1 \qquad \begin{array}{c}\vdots\ \boxed{\Pi_1} \\ \Gamma, A \vdash D\end{array} \qquad \begin{array}{c}\vdots\ \boxed{\Pi_2} \\ \Gamma, B \vdash D\end{array}}{\Gamma \vdash D}\,\mathsf{most\text{-}el}_1$$

Here we can reduce to either one of the following derivations of $\Gamma \vdash D$, which shows the non-confluence of the cut-elimination process. (Of course, one could fix a choice, e.g. always take the first possible cut from the left, but that would be completely arbitrary.)

$$\begin{array}{c}
\begin{array}{ccc}
\vdots\ \boxed{\Sigma_1} & & \vdots\ \boxed{\Sigma_1} \\
\Gamma \vdash A & \dots & \Gamma \vdash A
\end{array} \\
\vdots\ \boxed{\Pi_1} \\
\Gamma \vdash D
\end{array}
\qquad\qquad
\begin{array}{c}
\begin{array}{ccc}
\vdots\ \boxed{\Sigma_2} & & \vdots\ \boxed{\Sigma_2} \\
\Gamma \vdash B & \dots & \Gamma \vdash B
\end{array} \\
\vdots\ \boxed{\Pi_2} \\
\Gamma \vdash D
\end{array}$$

A more concrete example is the following.

$$\dfrac{\dfrac{\dfrac{A \wedge B \vdash A \wedge B}{A \wedge B \vdash A}\,\wedge\text{-el}_1 \qquad \dfrac{A \wedge B \vdash A \wedge B}{A \wedge B \vdash B}\,\wedge\text{-el}_2}{A \wedge B \vdash \mathsf{most}(A, B, C)}\,\mathsf{most\text{-}in}_1 \qquad \dfrac{A \vdash A}{A \vdash A \vee B}\,\vee\text{-in}_1 \qquad \dfrac{B \vdash B}{B \vdash A \vee B}\,\vee\text{-in}_2}{A \wedge B \vdash A \vee B}\,\mathsf{most\text{-}el}_1$$

This derivation can either be reduced to a derivation of $A \wedge B \vdash A \vee B$ via $A \wedge B \vdash A$ or via $A \wedge B \vdash B$.

It can happen that the introduction of a formula $\Phi = c(A_1, \dots, A_n)$ is not followed directly by an elimination for c, but first by other elimination rules, where Φ acts as a minor premise. In that way, a direct cut can be "blocked" by other elimination rules. So, apart from the cut-elimination arising from an introduction rule immediately followed by an elimination, we have a notion of "hidden" or *indirect* cut, where we want to permute one elimination rule over another.

▶ **Example 14.**

$$\dfrac{\dfrac{\Gamma \vdash A\vee B \qquad \dfrac{\Gamma, A, C \vdash C\to D}{\Gamma, A \vdash C \to D}\to\text{-in}_2 \qquad \Gamma, B\vdash C\to D}{\Gamma \vdash C\to D}\vee\text{-el} \qquad \Gamma\vdash C}{\Gamma\vdash D}\to\text{-el}$$

In this derivation, the cut arising from $\to$-in$_2$ followed by $\to$-el is blocked by the $\vee$-el rule where the major premise of the $\to$-el rule is a minor premise. This is an *indirect cut*, which can be contracted by permuting the $\to$-el rule over the $\vee$-el rule.

▶ **Definition 15.** Let c and c' be a connectives of arity n and n', with elimination rules r and r' respectively, both derived from the truth table. An *intuitionistic indirect cut* in a derivation is a pattern of the following form, where $\Phi = c(B_1,\ldots,B_n)$, $\Psi = c'(A_1,\ldots,A_{n'})$.

$$\dfrac{\dfrac{\Gamma\vdash\Psi\ldots\ \begin{array}{c}\vdots\,\boxed{\Sigma_j}\\ \Gamma\vdash A_j\end{array}\ \ldots\quad\ldots\ \begin{array}{c}\vdots\,\boxed{\Sigma_i}\\ \Gamma, A_i\vdash\Phi\end{array}\ \ldots}{\Gamma\vdash\Phi}\text{el}_{r'} \qquad \ldots\ \begin{array}{c}\vdots\,\boxed{\Pi_k}\\ \Gamma\vdash B_k\end{array}\ \ldots\quad\ldots\ \begin{array}{c}\vdots\,\boxed{\Pi_\ell}\\ \Gamma, B_\ell\vdash D\end{array}\ \ldots}{\Gamma\vdash D}\text{el}_r$$

- A_j ranges over all propositions that have a 1 in the truth table of c'; A_i ranges over all propositions that have a 0,
- B_k ranges over all propositions that have a 1 in the truth table of c; B_ℓ ranges over all propositions that have a 0.

The *elimination of an indirect cut* is defined by replacing the derivation pattern above by

$$\dfrac{\Gamma\vdash\Psi\ldots\ \begin{array}{c}\vdots\,\boxed{\Sigma_j}\\ \Gamma\vdash A_j\end{array}\ \ldots\quad\ldots\ \dfrac{\begin{array}{c}\vdots\,\boxed{\Sigma_i}\\ \Gamma,A_i\vdash\Phi\end{array}\ \ldots\ \begin{array}{c}\vdots\,\boxed{\Pi_k}\\ \Gamma,A_i\vdash B_k\end{array}\ \ldots\quad\ldots\ \begin{array}{c}\vdots\,\boxed{\Pi_\ell}\\ \Gamma,A_i,B_\ell\vdash D\end{array}\ \ldots}{\Gamma,A_i\vdash D}\text{el}_r\ \ldots}{\Gamma\vdash D}\text{el}_{r'}$$

This gives rise to copying of sub-derivations: for every A_i we copy the sub-derivations $\Pi_1,\ldots,\Pi_n$.

NB. Due to weakening, $\boxed{\Pi_k}$ is also a derivation of $\Gamma, A_i \vdash B_k$ and $\boxed{\Pi_\ell}$ is also a derivation of $\Gamma, A_i, B_\ell \vdash D$.

▶ **Example 16.** If we contract the indirect cut in Example 14, we obtain the following derivation.

$$\dfrac{\Gamma\vdash A\vee B \qquad \dfrac{\dfrac{\Gamma,A,C\vdash C\to D}{\Gamma,A\vdash C\to D}\to\text{-in}_2 \qquad \Gamma,A\vdash C}{\Gamma,A\vdash D}\to\text{-el} \qquad \dfrac{\Gamma,B\vdash C\to D\qquad \Gamma,B\vdash C}{\Gamma,B\vdash D}\to\text{-el}}{\Gamma\vdash D}\vee\text{-el}$$

4 The Curry-Howard isomorphism

We now define types terms for derivations, which enables the study of “proofs as terms” and emphasizes the computational interpretation of proofs. Here, we only define terms for derivations in the inituitionistic logic. We first define terms associated with connectives in general. Then, to show the usefulness of our approach to logical connectives, we will focus on the if-then-else connective.

▶ **Definition 17.** Given a logic with intuitionistic derivation rules, as derived from truth tables for a set of connectives $\mathcal{C}$, as in Definition 1, we now define the typed λ-calculus $\lambda^{\mathcal{C}}$. The system $\lambda^{\mathcal{C}}$ has judgments $\Gamma \vdash t : A$, where A is a formula, Γ is a set of declarations $\{x_1 : A_1, \ldots, x_m : A_m\}$, where the A_i are formulas and the x_i are term-variables such that every x_i occurs at most once in Γ, and t is a *proof-term*.

Let $c \in \mathcal{C}$ be a connective of arity n, which has 2^n rules (introduction plus elimination rules). For each rule r we have a term: an *introduction term*, $\{\overline{p} \,;\, \overline{Q}\}_r$, if r is an introduction rule, or an *elimination term*, $t \cdot_r [\overline{p} \,;\, \overline{Q}]$, if r is an elimination rule. Here, t is again a term, $\overline{p}$ is a finite sequence of terms and $\overline{Q}$ is a finite sequence of *abstracted terms* $\lambda x.q$, where x is a term-variable and q is a term. So the abstract syntax for proof-terms is is as follows.

$$t ::= x \mid \{\overline{t} \,;\, \overline{\lambda x.t}\}_r \mid t \cdot_r [\overline{t} \,;\, \overline{\lambda x.t}]$$

where x ranges over variables and r ranges over the rules of all the connectives.

The terms are *typed* using the following derivation rules.

$$\frac{}{\Gamma \vdash x_i : A_i} \text{ if } x_i : A_i \in \Gamma$$

$$\frac{\ldots \Gamma \vdash p_j : A_j \ldots \qquad \ldots \Gamma, y_i : A_i \vdash q_i : \Phi \ldots}{\Gamma \vdash \{\overline{p} \,;\, \overline{\lambda y.q}\}_r : \Phi} \text{ in}$$

$$\frac{\Gamma \vdash t : \Phi \qquad \ldots \Gamma \vdash p_k : A_k \ldots \qquad \ldots \Gamma, y_\ell : A_\ell \vdash q_\ell : D}{\Gamma \vdash t \cdot_r [\overline{p} \,;\, \overline{\lambda y.q}] : D} \text{ el}$$

Here, $\overline{p}$ is the sequence of terms $p_1, \ldots, p_{m'}$ for all the 1-entries in the truth table, and $\overline{\lambda y.q}$ is the sequence of terms $\lambda y_1.q_1, \ldots, \lambda y_m.q_m$ for all the 0-entries in the truth table.

One may think of the λ-abstracted variables as being *typed* so then one could write $\overline{\lambda y : A.q}$ and $\lambda y_1 : A_1.q_1, \ldots, \lambda y_m : A_r.q_m$. However, this clutters up the syntax considerably, so we will leave these types implicit. Moreover, decidability of typing, or a typing algorithm for (untyped) terms of the calculus $\lambda^{\mathcal{C}}$ is not our concern here.

We will sometimes leave the rule r that the elimination or introduction term corresponds to implicit, or we will just number the terms or introduce special names for them without explicit reference to the rule. It should be clear that every line in the truth table for the connective gives rise to one rule, which again gives rise to one term-constructor, which is either an elimination or an introduction term-constructor.

There are term reduction rules that correspond to the elimination of direct cuts.

▶ **Definition 18.** Given a direct cut as defined in Definition 12, we add reduction rules for the associated terms as follows.

- For the $\ell = j$ case, that is, $y_\ell : A_\ell$ and $p_j : A_j$ with $A_\ell = A_j$:

$$\{\overline{p, p_j} \,;\, \overline{\lambda x.q}\} \cdot [\overline{s} \,;\, \overline{\lambda y.r, \lambda y_\ell.r_\ell}] \longrightarrow_a r_\ell[y_\ell := p_j]$$

- For the $k = i$ case, that is, $s_k : A_k$ and $x_i : A_i$ with $A_k = A_i$:

$$\{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_a q_i[x_i := s_k] \cdot [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$$

For simplicity of presentation we write the "matching cases" in Definition 12 as last term of the sequence. So when writing $\overline{p, p_j}$, this should be understood as a sequence of terms

$p_1, \ldots, p_j, \ldots p_{m'}$, where we have singled out the p_j that matches the r_ℓ in $\overline{\lambda y.r, \lambda y_\ell.r_\ell}$. Similarly for $\overline{s, s_k}$ and $\overline{\lambda x.q, \lambda x_i.q_i}$.

The reduction is extended in the straightforward way to sub-terms, by defining it as a congruence with respect to the term constructions.

This Definition gives a reduction rule, and possibly more than one, for every combination of an elimination and an introduction. For an n-ary connective, there are 2^n rules in the truth table, and therefore 2^n term-constructors (introduction plus elimination constructors). Often, we will want to just look at the optimized rules, following Lemmas 7 and 10. For these optimized rules, there is also a straightforward definition of proof-terms and of the reduction relation associated with cut-elimination. The Lemmas 7 and 10 can be extended to terms and reductions: the proof-terms for the optimized rules can be defined in terms of our original calculus $\lambda^{\mathcal{C}}$, and the reduction rules for the optimized proof terms are an instance of reductions in the original calculus (often multi-step).

▶ **Definition 19.** Given an indirect cut as defined in Definition 15, we add reduction rules for the associated terms as follows.

$$(t \cdot [\overline{p} \,;\, \overline{\lambda x.q}]) \cdot [\overline{s} \,;\, \overline{\lambda y.r}] \quad \longrightarrow_b \quad t \cdot [\overline{p} \,;\, \overline{\lambda x.(q \cdot [\overline{s} \,;\, \overline{\lambda y.r}])}]$$

Here, the notation $\overline{\lambda x.(q \cdot [\overline{s} \,;\, \overline{\lambda y.r}])}$ should be understood as a sequence $\lambda x_1.q_1, \ldots, \lambda x_m.q_m$ where each q_j is replaced by $q_j \cdot [\overline{s} \,;\, \overline{\lambda y.r}]$.

The reduction is extended in the straightforward way to sub-terms, by defining it as a congruence with respect to the term constructions.

▶ Notation 20. We omit brackets by letting the application operator $- \cdot -$ associate to the left, so $t \cdot [\overline{p} \,;\, \overline{\lambda x.q}] \cdot [\overline{s} \,;\, \overline{\lambda y.r}]$ denotes $(t \cdot [\overline{p} \,;\, \overline{\lambda x.q}]) \cdot [\overline{s} \,;\, \overline{\lambda y.r}]$. We will also omit the brackets in $\overline{\lambda x.(q \cdot [\overline{s} \,;\, \overline{\lambda y.r}])}$, because no ambiguity can arise here.

▶ Remark. In [22], yet another notion of cut is defined, called *simplification cut*. This is a situation where the assumption is unused in an introduction or elimination rule and the rule can be removed all together. Adding these rules is not necessary for the sub-formula property, so we don't introduce it here. On the term level, an elimination of simplification cuts would amount to the following reduction rules.

$$\begin{array}{rcll} t \cdot [\overline{p} \,;\, \overline{\lambda x.q}] & \longrightarrow_b & q_i & \text{if } x_i \notin \mathrm{FV}(q_i) \\ \{\overline{p} \,;\, \overline{\lambda x.q}\} & \longrightarrow_b & q_i & \text{if } x_i \notin \mathrm{FV}(q_i) \end{array}$$

4.1 Extending the Curry-Howard isomorphism to definable rules

The optimizations for the logical rules, as given in Lemmas 7 and 10 can be extended to the proof terms and also to cut-elimination. This gives us the possibility to capture questions related to normalization by looking at normalization for terms in the original calculus $\lambda^{\mathcal{C}}$. We will now describe the terms for the optimized rules in detail.

▶ **Definition 21.** For each optimization step in Lemmas 7 and 10 we give the canonical term for the optimized rule and its translation in terms of $\lambda^{\mathcal{C}}$ of Definition 17.

We first treat the two optimizations arising from Lemma 7, and then the optimization arising from Lemma 10.

- Given two rules

$$\frac{\vdash p_1 : A_1 \ \ldots \ \vdash p_n : A_n \qquad x_1 : B_1 \vdash q_1 : \Phi \ \ldots \ x_m : B_m \vdash q_m : \Phi \qquad z : A \vdash s : \Phi}{\vdash \{\overline{p} \,;\, \overline{\lambda x.q, \lambda z.s}\}_r : \Phi}\ \mathrm{in}_r$$

$$\frac{\vdash p_1 : A_1 \ \dots \ \vdash p_n : A_n \qquad \vdash a : A \qquad x_1 : B_1 \vdash q_1 : \Phi \ \dots \ x_m : B_m \vdash q_m : \Phi}{\vdash \{\overline{p, a} \ ; \ \overline{\lambda x.q}\}_{r'} : \Phi} \text{in}_{r'}$$

we have the following term for the optimized introduction rule

$$\frac{\vdash p_1 : A_1 \ \dots \ \vdash p_n : A_n \qquad x_1 : B_1 \vdash q_1 : \Phi \ \dots \ x_m : B_m \vdash q_m : \Phi}{\vdash \{\overline{p} \ ; \ \overline{\lambda x.q, \lambda z.\{\overline{p, z} \ ; \ \overline{\lambda x.q}\}_{r'}}\}_r : \Phi} \text{in}^{\text{opt}}_{r,r'}$$

We will abbreviate the latter term to $\{\overline{p} \ ; \ \overline{\lambda x.q}\}^{\circ}_{r,r'}$

- Given two rules

$$\frac{\vdash t : \Phi \qquad \vdash p_1 : A_1 \ \dots \ \vdash p_n : A_n \qquad x_1 : B_1 \vdash q_1 : D \ \dots \ x_m : B_m \vdash q_m : D \qquad z : A \vdash s : D}{\vdash t \cdot_r [\overline{p} \ ; \ \overline{\lambda x.q, \lambda z.s}] : D} \text{el}_r$$

$$\frac{\vdash t : \Phi \qquad \vdash p_1 : A_1 \ \dots \ \vdash p_n : A_n \qquad \vdash a : A \qquad x_1 : B_1 \vdash q_1 : D \ \dots \ x_m : B_m \vdash q_m : D}{\vdash t \cdot_{r'} [\overline{p, a} \ ; \ \overline{\lambda x.q}] : D} \text{el}_{r'}$$

we have the following term for the optimized elimination rule

$$\frac{\vdash t : \Phi \qquad \vdash p_1 : A_1 \ \dots \ \vdash p_n : A_n \qquad x_1 : B_1 \vdash q_1 : D \ \dots \ x_m : B_m \vdash q_m : D}{\vdash t \cdot_r [\overline{p} \ ; \ \overline{\lambda x.q, \lambda z.t \cdot_{r'} [\overline{p, z} \ ; \ \overline{\lambda x.q}]}] : D} \text{el}^{\text{opt}}_{r,r'}$$

We will abbreviate the latter term to $t \odot_{r,r'} [\overline{p} \ ; \ \overline{\lambda x.q}]$

- Given the rule

$$\frac{\vdash t : \Phi \qquad \vdash p_1 : A_1 \ \dots \ \vdash p_n : A_n \qquad z : A \vdash s : D}{\vdash t \cdot_r [\overline{p} \ ; \ \lambda z.s] : D} \text{el}_r$$

we have the following term for the optimized elimination rule

$$\frac{\vdash t : \Phi \qquad \vdash p_1 : A_1 \ \dots \ \vdash p_n : A_n}{\vdash t \cdot_r [\overline{p} \ ; \ \lambda z.z] : A} \text{el}^{\text{opt}}_{r}$$

We will abbreviate the latter term to $t \odot_r [\overline{p}]$

There is a canonical way in which the notions of direct cut and cut-elimination extend to the optimized rules: the same rules as in Definition 18 apply. In case of a term of the form $\{\dots \ ; \ \dots\} \cdot [\dots \ ; \ \dots]$, a reduction is always possible, also in the case of optimized rules. For the indirect cuts, the situation is similar: the same rules as in Definition 19 apply.

▶ Remark. To clarify, we want to note explicitly that $t \odot_r [\overline{p}] \cdot_{r'} [\overline{q} \ ; \ \overline{\lambda x.s}]$ does not reduce to $t \odot_r [\overline{p}]$. In case we only have the optimized rules, it does not reduce at all. If we consider $t \odot_r [\overline{p}]$ as a definition in the original calculus $\lambda^{\mathcal{C}}$, we do have a reduction,

$$t \odot_r [\overline{p}] \cdot_{r'} [\overline{q} \ ; \ \overline{\lambda x.s}] \longrightarrow_b t \cdot_r [\overline{p} \ ; \ \lambda z.z \cdot_{r'} [\overline{q} \ ; \ \overline{\lambda x.s}]]$$

but this uses a non-optimized elimination.

▶ **Lemma 22.** *The translation of an $\longrightarrow_a$ step in the optimized calculus translates to a (possibly multistep) $\longrightarrow_a$ step in the original calculus $\lambda^{\mathcal{C}}$.*

Proof. We show two cases:

1. If $\{\overline{t}\ ;\ \overline{\lambda y.v}\}^{\circ}_{r_3,r_4} \odot_{r_1,r_2} [\overline{p}\ ;\ \overline{\lambda x.q}] \longrightarrow_a R$ (using the optimized rules) and $\{\overline{t}\ ;\ \overline{\lambda y.r}\}^{\circ}_{r_3,r_4} \odot_{r_1,r_2} [\overline{p}\ ;\ \overline{\lambda x.q}]$ translates to T in the original calculus $\lambda^{\mathcal{C}}$, then there is a term T' such that $T \longrightarrow_a^+ T'$ and R translates to T' in $\lambda^{\mathcal{C}}$. Here $\longrightarrow_a^+$ denotes a non-zero sequence of reductions.
   Then the translation T is as follows. $T = M \cdot [\overline{p}\ ;\ \overline{\lambda x.q, \lambda z.\overline{M \cdot [\overline{p,z}\ ;\ \overline{\lambda x.q}]}}]$, where we abbreviate $M := \{\overline{t}\ ;\ \overline{\lambda y.v, \lambda z.\{\overline{t,z}\ ;\ \overline{\lambda y.v}\}}\}$. There are two possible cases for the reduction.
   - Case $\{\overline{t}\ ;\ \overline{\lambda y.v}\}^{\circ}_{r_3,r_4} \odot_{r_1,r_2} [\overline{p}\ ;\ \overline{\lambda x.q}] \longrightarrow_a q_\ell[x_\ell := t_j]$. Then $T \longrightarrow_a q_\ell[x_\ell := t_j]$ and we are done.
   - Case $\{\overline{t}\ ;\ \overline{\lambda y.v}\}^{\circ}_{r_3,r_4} \odot_{r_1,r_2} [\overline{p}\ ;\ \overline{\lambda x.q}] \longrightarrow_a v_i[y_i := p_k] \odot_{r_1,r_2} [\overline{p}\ ;\ \overline{\lambda x.q}]$. Then
   $$\begin{aligned} T &\longrightarrow_a v_i[y_i := p_k] \cdot [\overline{p}\ ;\ \overline{\lambda x.q, \lambda z.M \cdot [\overline{p,z}\ ;\ \overline{\lambda x.q}]}] \\ &\longrightarrow_a v_i[y_i := p_k] \cdot [\overline{p}\ ;\ \overline{\lambda x.q, \lambda z.v_i[y_i := p_k] \cdot [\overline{p,z}\ ;\ \overline{\lambda x.q}]}] \end{aligned}$$
   and we are done.
2. If $\{\overline{t}\ ;\ \overline{\lambda y.v}\}^{\circ}_{r_2,r_3} \odot_{r_1} [\overline{p}] \longrightarrow_a R$ and $\{\overline{t}\ ;\ \overline{\lambda y.r}\}^{\circ}_{r_2,r_3} \odot_{r_1} [\overline{p}]$ translates to T in the original calculus $\lambda^{\mathcal{C}}$, then there is a term T' such that $T \longrightarrow_a^+ T'$ and R translates to T' in $\lambda^{\mathcal{C}}$.
   Now the translation T is as follows. $T = \{\overline{t}\ ;\ \overline{\lambda y.v, \lambda z.\{\overline{t,z}\ ;\ \overline{\lambda y.v}\}}\} \cdot [\overline{p}\ ;\ \lambda z.z]$. There is one possibility for the reduction.
   - Case $\{\overline{t}\ ;\ \overline{\lambda y.v}\}^{\circ}_{r_2,r_3} \odot_{r_1} [\overline{p}] \longrightarrow_a v_i[y_i := p_k] \odot_{r_1} [\overline{p}]$. Then
   $$T \longrightarrow_a v_i[y_i := p_k] \cdot [\overline{p}\ ;\ \lambda z.z]$$
   and we are done.

◀

As mentioned, Schroeder-Heister[17] has proposed another elimination rule for $\wedge$ which is slightly different from ours. Von Plato [22] calls this *general elimination* while Tennant [20] calls it *parallel elimination*. We call it parallel $\wedge$-elimination and give it in typed λ-calculus format.

▶ **Definition 23.** We define the *parallel $\wedge$-elimination rule* as follows

$$\frac{\Gamma \vdash t : A \wedge B \qquad \Gamma, x : A, y : B \vdash q : D}{\vdash t \cdot^{\mathrm{par}} [\lambda x, y.q] : D} \wedge\text{-el}$$

The cut-elimination rule associated with this rule is as follows.

$$\{a, b\ ;\ \} \cdot^{\mathrm{par}} [\lambda x, y.q] \longrightarrow_{\mathrm{par}} q[x := a, y := b].$$

We show that this elimination rule can be translated in terms of ours and that cut-elimination reduction is preserved.

▶ **Definition 24.** We translate the parallel $\wedge$-elimination rule of Definition 23 by defining it in terms of the optimized elimination rules for $\wedge$, rules $\wedge$-el$'_1$ and $\wedge$-el$'_2$ of Example 6 as follows.

Given terms for these rules:

$$\frac{\Gamma, \vdash t : A \wedge B \qquad \Gamma, x : A \vdash q : D}{\Gamma \vdash t \cdot_1 [\lambda x.q] : D} \wedge\text{-el}'_1 \qquad \frac{\Gamma, \vdash t : A \wedge B \qquad \Gamma, y : B \vdash q : D}{\Gamma \vdash t \cdot_2 [\lambda y.q] : D} \wedge\text{-el}'_2$$

define

$$t \cdot^{\mathrm{par}} [\lambda x, y.q] := t \cdot_1 [\lambda x.t \cdot_2 [\lambda y.q]].$$

▶ **Lemma 25.** *The defined term $t \cdot^{\mathrm{par}} [\lambda x, y.q]$ is of the right type and the translation of an $\longrightarrow_{\mathrm{par}}$ step in the calculus with the parallel $\wedge$-elimination rule translates to multistep $\longrightarrow_a$ in the original calculus $\lambda^{\mathcal{C}}$.*

Proof. Given $\Gamma \vdash t : A \wedge B$ and $\Gamma, x : A, y : B \vdash q : D$, we have

$$\dfrac{\Gamma \vdash t : A \wedge B \qquad \dfrac{\Gamma, x : A \vdash t : A \wedge B \qquad \Gamma, x : A, y : B \vdash q : D}{\Gamma, x : A \vdash t \cdot_2 [\lambda y.q] : D}\ \wedge\text{-el}_2'}{\Gamma \vdash t \cdot_1 [\lambda x.t \cdot_2 [\lambda y.q]] : D}\ \wedge\text{-el}_1'$$

The reduction can easily be verified:

$$\{a, b \,;\, \} \cdot_1 [\lambda x.\{a, b \,;\, \} \cdot_2 [\lambda y.q]] \longrightarrow_a \{a, b \,;\, \} \cdot_2 [\lambda y.q[x := a]] \longrightarrow_a q[x := a, y := b].$$

◀

We define the standard rule for $\to$-introduction and show that this introduction rule can be translated in terms of ours and that cut-elimination reduction is preserved.

▶ **Definition 26.** We define the *standard rule for $\to$-introduction* as follows, where we describe it using terms.

$$\dfrac{\Gamma, x : A \vdash q : B}{\Gamma \vdash \{\lambda x.q\}^{\to} : A \to B}\ \to\text{-in}$$

The cut-elimination rule associated with this rule is as follows.

$$\{\lambda x.q\}^{\to} \cdot [a \,;\,] \longrightarrow_s q[x := a].$$

▶ **Definition 27.** We define the standard $\to$-introduction rule in terms of optimized $\to$-rules (Example 9) which are – with terms – as follows.

$$\dfrac{\Gamma, x : A \vdash q : A \to B}{\Gamma \vdash \{\,;\, \lambda x.q\}_1 : A \to B}\ \to\text{-in}_1 \qquad \dfrac{\Gamma \vdash t : B}{\Gamma \vdash \{t \,;\, \}_2 : A \to B}\ \to\text{-in}_2$$

Now, given $\Gamma, x : A \vdash q : B$ we define

$$\{\lambda x.q\}^{\to} := \{\,;\, \lambda x.\{q \,;\, \}_2\}_1.$$

▶ **Lemma 28.** *The translation of $\{\lambda x.q\}^{\to}$ is well-typed and the translation of an $\longrightarrow_s$ step in the calculus with the standard rule for $\to$ translates to multistep $\longrightarrow_a$ in the original calculus $\lambda^{\mathcal{C}}$.*

Proof. The well-typedness is easily verified. For the reduction:

$$\{\,;\, \lambda x.\{q \,;\, \}_2\}_1 \cdot [a \,;\,] \longrightarrow_a \{q[x := a] \,;\, \}_2 \cdot [a \,;\,] \longrightarrow_a q[x := a].$$

◀

We define the traditional rule for $\neg$-introduction and show that it can be translated in terms of ours and that cut-elimination reduction is preserved.

▶ **Definition 29.** We define the *traditional rule for $\neg$-introduction* as follows, where we describe it using terms.

$$\dfrac{\Gamma, x : A \vdash t : \neg B \qquad \Gamma, y : A \vdash q : B}{\Gamma \vdash \{\lambda x.t, \lambda y.q\}^{\mathrm{t}} : A}\ \neg\text{-in}^{\mathrm{t}}$$

The cut-elimination rule associated with this rule is as follows.

$$\{\lambda x.t, \lambda y.q\}^{\mathrm{t}} \cdot [a \,;\,] \longrightarrow_{\neg} t[x := a] \cdot [q[y := a] \,;\,].$$

▶ **Definition 30.** We define the traditional ¬-introduction rule in terms of the ¬-rules of Example 9 which are – with terms – as follows.

$$\frac{\Gamma \vdash t : \neg A \qquad \Gamma \vdash a : A}{\Gamma \vdash t \cdot [a \,;\,] : D}\ \neg\text{-el} \qquad \frac{\Gamma, x : A \vdash q : \neg A}{\Gamma \vdash \{\ ;\ \lambda x.q\} : \neg A}\ \neg\text{-in}$$

Now, given $\Gamma, x : A \vdash t : \neg B$ and $\Gamma, y : A \vdash q : B$ we define

$$\{\lambda x.t, \lambda y.q\}^{\mathrm{t}} := \{\ ;\ \lambda x.t \cdot [q[y := x]\ ;\ ]\}$$

▶ **Lemma 31.** *The translation of $\{\lambda x.t, \lambda y.q\}^{\mathrm{t}}$ is well-typed and the translation of an $\longrightarrow_{\neg}$ step in the calculus with the traditional rule for ¬ translates to multistep $\longrightarrow_a$ in the original calculus $\lambda^{\mathcal{C}}$.*

Proof. For the well-typedness:

$$\cfrac{\cfrac{\Gamma, x : A \vdash t : \neg B \qquad \Gamma, y : A \vdash q : B}{\Gamma, x : A \vdash t \cdot [q[y := x]\ ;\ ] : \neg A}\ \neg\text{-el}}{\Gamma \vdash \{\ ;\ \lambda x.t \cdot [q[y := x]\ ;\ ]\} : \neg A}\ \neg\text{-in}$$

For the reduction:

$$\{\ ;\ \lambda x.t \cdot [q[y := x]\ ;\ ]\} \cdot [a\ ;\ ] \longrightarrow_a t[x := a] \cdot [q[x := a]\ ;\ .]$$

◀

5 Normalization

In this section we prove that $\longrightarrow_a$ and $\longrightarrow_b$ are both strongly normalizing. We also give a proof of weak normalization of the combination of $\longrightarrow_a$ and $\longrightarrow_b$.

▶ **Theorem 32.** *The reduction $\longrightarrow_b$ is strongly normalizing.*

Proof. We define a measure $|-|$ from terms to natural numbers that decreases with every reduction step. For notational convenience we suppress the reference to the derivation rule r.

$$\begin{array}{rcl} |x| & := & 1 \\ |\{\overline{p}\ ;\ \overline{\lambda y.q}\}| & := & \Sigma|p_i| + \Sigma|q_j| \\ |t \cdot [\overline{s}\ ;\ \overline{\lambda y.u}]| & := & |t|(2 + \Sigma|s_k| + \Sigma|u_\ell|) \end{array}$$

It can easy be verified that, if $t_0 \longrightarrow_b t_1$, then $|t_0| > |t_1|$, so $\longrightarrow_b$ is strongly normalizing. ◀

▶ **Corollary 33.** *The reduction $\longrightarrow_b$ for the optimized rules of Definition 21, the standard rule for →-elimination of Definition 26, the parallel ∧-elimination rule of Definition 23 and the traditional rule for ¬-elimination of Definition 29 are strongly normalizing.*

Proof. The same metrics as in the proof of Theorem 32 applies. For the parallel reduction, define $|t \cdot^{\mathrm{par}} [\lambda x, y.q]| := |t|(2 + |q|)$.

◀

5.1 Strong Normalization of the elimination of direct cuts

We now prove SN for $\longrightarrow_a$ by adapting the well-known *saturated sets method* of Tait [19] and Girard [8] to our calculus. We write SN for the set of strongly normalizing (untyped) terms and we write Term for the set of all untyped terms and Var for the set of variables.

▶ **Definition 34.** **1.** The set Neut of *neutral terms* is defined by

  a. $\mathsf{Var} \subseteq \mathsf{Neut}$,

  b. $t \cdot [\overline{p}\ ;\ \overline{\lambda y.q}] \in \mathsf{Neut}$ for all $t \in \mathsf{Neut}$ and $\overline{p}, \overline{\lambda y.q} \in \mathsf{SN}$.

2. The term t does a *key reduction* to t', notation $t \longrightarrow_a^k t'$, in case

  a. t is a redex itself (according to Definition 17) and t' is its reduct,

  b. $t = t_0 \cdot [\overline{p}\ ;\ \overline{\lambda y.q}]$, $t' = t_1 \cdot [\overline{p}\ ;\ \overline{\lambda y.q}]$ and $t_0 \longrightarrow_a^k t_1$.

3. A set $X \subseteq \mathsf{Term}$ is *saturated* if it satisfies the following properties

  a. $X \subseteq \mathsf{SN}$,

  b. $\mathsf{Neut} \subseteq X$

  c. X is closed under *key-redex expansion*: if $t \in \mathsf{SN}$ and $\forall q(t \longrightarrow_a^k q \Rightarrow q \in X)$, then $t \in X$.

4. For a connective c of arity n and $X_1, \ldots, X_n \in \mathsf{SAT}$ we define the set $c(X_1, \ldots, X_n)$ as follows. Assume that $r_1, \ldots, r_m$ are the elimination rules for c.

$$\begin{aligned} c(X_1, \ldots, X_n) := \{t \mid \forall r_i \in \{r_1, \ldots, r_m\} \\ \forall D \in \mathsf{SAT}, \forall \overline{p}, \overline{q} \in \mathsf{Term} \\ \forall k(p_k \in X_k) \wedge (\forall \ell\, \forall u_\ell \in X_\ell\, (q_\ell[y_\ell := u_\ell] \in D)) \quad \Longrightarrow \quad t \cdot [\overline{p}\ ;\ \overline{\lambda y.q}]_{r_i} \in D\ \}\end{aligned}$$

In the definition of $c(X_1, \ldots, X_n)$ it should be clear that we quantify over all elimination rules for the connective c.

▶ **Lemma 35.** *If* $X_1, \ldots, X_n \in \mathsf{SAT}$*, then* $c(X_1, \ldots, X_n) \in \mathsf{SAT}$.

Proof. We check the 3 conditions for $c(X_1, \ldots, X_n)$. Suppose $X_1, \ldots, X_n \in \mathsf{SAT}$.

a. That $c(X_1, \ldots, X_n) \subseteq \mathsf{SN}$ follows directly from the fact that if $t \in c(X_1, \ldots, X_n)$, then $t \cdot [\overline{p}\ ;\ \overline{\lambda x.q}] \in D$ and $D \subseteq \mathsf{SN}$, so $t \cdot [\overline{p}\ ;\ \overline{\lambda x.q}] \in \mathsf{Term}$, so $t \in \mathsf{SN}$.

b. For $t \in \mathsf{Neut}$ and $D \in \mathsf{SAT}$ and $\overline{p}, \overline{q} \in \mathsf{SN}$ with $\forall k(p_k \in X_k)$ and $\forall \ell\, \forall u_\ell \in X_\ell\, (q_\ell[y_\ell := u_\ell] \in D)$, we have $t \cdot [\overline{p}\ ;\ \overline{\lambda y.q}]_{r_i} \in \mathsf{Neut} \subset D$, so we can conclude that $t \in c(X_1, \ldots, X_n)$.

c. Suppose $t \in \mathsf{SN}$ and $\forall t'(t \longrightarrow_a^k t' \Rightarrow t' \in c(X_1, \ldots, X_n))$ (*). Let r_i be a rule for c and let $D \in \mathsf{SAT}$, $\overline{p}, \overline{q} \in \mathsf{Term}$ with $\forall k(p_k \in X_k)$ and $\forall \ell\, \forall u_\ell \in X_\ell\, (q_\ell[y_\ell := u_\ell] \in D)$. For all t' with $t \longrightarrow_a^k t'$ we have $t \cdot [\overline{p}\ ;\ \overline{\lambda y.q}]_{r_i} \longrightarrow_a^k t' \cdot [\overline{p}\ ;\ \overline{\lambda y.q}]_{r_i}$ and $t' \cdot [\overline{p}\ ;\ \overline{\lambda y.q}]_{r_i} \in D$ by (*). So, $t \cdot [\overline{p}\ ;\ \overline{\lambda y.q}]_{r_i} \in D$ and so $t \in c(X_1, \ldots, X_n)$.

◀

We use the saturated sets as a semantics for types: if A is a type, $\langle A \rangle$ will be a saturated set. The simplest way to do this is to interpret all type variables (proposition letters) as the set SN, which is indeed a saturated set.

▶ **Definition 36.** For A a type, we define $\langle A \rangle$ by induction on A as follows.

- $\langle A \rangle := \mathsf{SN}$ if A is a proposition letter.
- $c(A_1, \ldots, A_n) := c(\langle A_1 \rangle, \ldots, \langle A \rangle_n)$, where the right hand side is the interpretation of the connective c on saturated sets, as given in Definition 34, case (4).

We will often confuse A and $\langle A \rangle$, to avoid notational overhead, and just identify the proposition A with its interpretation as a saturated set $\langle A \rangle$.

▶ **Definition 37.** Given a context Γ, a map (valuation) $\rho : \mathsf{Var} \to \mathsf{Term}$ satisfies Γ, notation $\rho \models \Gamma$, in case $\rho(x) \in \langle A \rangle$ for all $x : A \in \Gamma$.

If $t \in \mathsf{Term}$ and $\rho : \mathsf{Var} \to \mathsf{Term}$, we write $\langle t \rangle_\rho$ for t where ρ has been carried out as a substitution on t.

A valuation $\rho : \mathsf{Var} \to \mathsf{Term}$ is only relevant for a finite number of variables: those that are declared in the context Γ under consideration. So we will always assume that $\rho(x) \neq x$ only for a finite number of $x \in \mathsf{Var}$. Those x we call the *support of* ρ. When applying ρ as a substitution to a term t we may need to "go under a λ", e.g. when applying ρ to $\{\overline{p}\ ; \overline{\lambda x.q}\}$ In this case we always assume that the bound variable is not in the support of ρ. (We can always rename it.)

▶ **Lemma 38.** *If* $\Gamma \vdash t : A$*, and* $\rho \models \Gamma$*, then* $\langle t \rangle_\rho \in \langle A \rangle$*.*

Proof. By induction on the derivation of $\Gamma \vdash t : A$. Suppose $\rho \models \Gamma$. For the (axiom) case, it is trivial. We ignore ρ for the rest of the proof, as it gives a lot of notational overhead, so we just write t for $\langle t \rangle_\rho$.

- Suppose $\Phi = c(A_1, \ldots, A_n)$ and

$$\frac{\ldots \Gamma \vdash s_j : A_j \ldots \qquad \ldots \Gamma, x_i : A_i \vdash t_i : \Phi \ldots}{\Gamma \vdash \{\overline{s}\ ; \overline{\lambda x.t}\}_r : \Phi}\ \text{in}$$

  Let r' be a rule for c, $D \in \mathsf{SAT}$, $\overline{p}, \overline{q} \in \mathsf{Term}$ with $\forall k (p_k \in A_k)$ and $\forall \ell \forall u_\ell \in A_\ell\, (q_\ell[y_\ell := u_\ell] \in D)$. For $\{\overline{s}\ ; \overline{\lambda x.t}\}_r \cdot_{r'} [\overline{p}\ ; \overline{\lambda y.q}]$ there are the following possible key-reductions:

$$\{\overline{s}\ ; \overline{\lambda x.t}\}_r \cdot_{r'} [\overline{p}\ ; \overline{\lambda y.q}] \quad \longrightarrow_a^k \quad q_l[y_l := s_j] \tag{1}$$

$$\{\overline{s}\ ; \overline{\lambda x.t}\}_r \cdot_{r'} [\overline{p}\ ; \overline{\lambda y.q}] \quad \longrightarrow_a^k \quad t_i[x_i := p_k] \cdot_{r'} [\overline{p}\ ; \overline{\lambda y.q}] \tag{2}$$

  In case (1), $q_l[y_l := s_j] \in D$ by the assumption and the induction hypothesis. In case (2), $t_i[x_i := p_k] \in \Phi$ by the induction hypothesis and so $t_i[x_i := p_k] \cdot_{r'} [\overline{p}\ ; \overline{\lambda y.q}] \in D$ by the definition of $\Phi = c(A_1, \ldots, A_n)$ as a saturated set. So, $\{\overline{s}\ ; \overline{\lambda x.t}\}_r \cdot_{r'} [\overline{p}\ ; \overline{\lambda y.q}] \in \mathsf{SN}$ and all its key reductions are in D, so the term is in D. Therefore, $\{\overline{s}\ ; \overline{\lambda x.t}\}_r \in \Phi$.
- Suppose $\Phi = c(A_1, \ldots, A_n)$ and

$$\frac{\Gamma \vdash t : \Phi \qquad \ldots \Gamma \vdash p_k : A_k \ldots \qquad \ldots \Gamma, y_\ell : A_\ell \vdash q_\ell : D}{\Gamma \vdash t \cdot_r [\overline{p}\ ; \overline{\lambda y.q}] : D}\ \text{el}$$

  Then $t \cdot_r [\overline{p}\ ; \overline{\lambda y.q}] = t \cdot_r [\overline{p}\ ; \overline{\lambda y.q}] \in D$ by $t \in \Phi = c(A_1, \ldots, A_n)$ and the definition of $c(A_1, \ldots, A_n)$ as a saturated set and the induction hypothesis.

◀

The following is now an immediate corollary by taking $\rho(x) := x$ for all $x \in \mathsf{Var}$. Because $\mathsf{Var} \subseteq \mathsf{Neut} \subseteq \langle A \rangle$, we know that $\rho \models \Gamma$. So, if $\Gamma \vdash t : A$, then $\langle t \rangle_\rho = t \in \langle A \rangle \subseteq \mathsf{SN}$.

▶ **Theorem 39.** *The reduction* $\longrightarrow_a$ *is strongly normalizing: all* $\longrightarrow_a$*-reductions on proof terms are finite.*

▶ **Corollary 40.** *The reduction* $\longrightarrow_a$ *for the optimized rules of Definition 21, the parallel* $\wedge$*-elimination rule of Definition 23, the standard* $\to$*-introduction of Definition 26 and the traditional rule for* $\neg$*-elimination of Definition 29 are strongly normalizing.*

Proof. By Theorem 39 and the fact that reduction is preserved by the translation: Lemmas 22, 25 and 28. ◀

5.2 Weak Normalization of cut-elimination

We now give a strategy for finding a normal form for the combined $\longrightarrow_{ab}$ reduction, the union of $\longrightarrow_a$ and $\longrightarrow_b$. This proves that $\longrightarrow_{ab}$ is weakly normalizing and it also gives a concrete procedure for finding a normal form. Due to the fact that, in general, reduction is not confluent, this normal form is not unique, but it does yield *decidability* via the *sub-formula property*. The weak normalization proof follows the well-known idea, originally due to Turing (see [5]) for simple type theory, to *contract the innermost redex of highest rank*.

▶ **Definition 41.** We define the *rank of a formula* A, $\mathsf{rk}(A)$ as follows.

- $\mathsf{rk}(A) := 1$ if A is a proposition letter.
- $\mathsf{rk}(c(A_1,\ldots,A_n) := 1 + \max\{\mathsf{rk}(A_1),\ldots,\mathsf{rk}(A_n)\}$ if c is a connective of arity n.

We define the *rank of a redex* as follows.

- The rank of $\{\overline{p} \,;\, \overline{\lambda x.q}\}_{r'} \cdot_r [\overline{s} \,;\, \overline{\lambda y.r}]$ is the rank of the type of $\{\overline{p} \,;\, \overline{\lambda x.q}\}_{r'}$.
- The rank of $(t \cdot_{r'} [\overline{p} \,;\, \overline{\lambda x.q}]) \cdot_r [\overline{s} \,;\, \overline{\lambda y.r}]$ is the rank of the type of $t \cdot_{r'} [\overline{p} \,;\, \overline{\lambda x.q}]$.

We will sometimes mark the redex with its type Φ such that $\mathsf{rk}(\Phi)$ is the rank of the redex. We do this by writing Φ as a subscript to the elimination constructor. To clarify, we summarize again the possible reduction steps of the form $\longrightarrow_a$ and $\longrightarrow_b$.

▶ Notation 42. From Definition 18, we have the reduction $\longrightarrow_a$ and form Definition 19 we have the reduction $\longrightarrow_b$. We introduce the following notation.

$$\begin{array}{rcl}
\{\overline{p, p_j} \,;\, \overline{\lambda x.q}\} \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r, \lambda y_\ell.r_\ell}] & \longrightarrow_{a1} & r_\ell[y_\ell := p_j] \\
\{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] & \longrightarrow_{a2} & q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \\
(t \cdot [\overline{p} \,;\, \overline{\lambda x.q}]) \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}] & \longrightarrow_b & t \cdot [\overline{p} \,;\, \overline{\lambda x.(q \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}])}]
\end{array}$$

Here, the proviso's of Definition 18 apply, so the first is the "$\ell = j$ case" which we will call $\longrightarrow_{a1}$, and the second is the "$k = i$ case" which we will call $\longrightarrow_{a2}$.

We give two Lemmas that show that the creation of new redexes is limited.

▶ **Lemma 43.** *1. If $t \longrightarrow_b t'$ by contracting a redex of $\mathsf{rk}(\Phi)$ then the newly created redexes are also of $\mathsf{rk}(\Phi)$.*

2. *Suppose $\{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$. If $q_i[x_i := s_k]$ is an introduction term (that is: $q_i[x_i := s_k]$ is of the form $\{\ldots \,;\, \ldots\}$), then q_i is an introduction term. Similarly, if $q_i[x_i := s_k]$ is an elimination term (that is: $q_i[x_i := s_k]$ is of the form $\ldots \cdot [\ldots \,;\, \ldots]$), then q_i is an elimination term.*

Proof. **1.** If $t \longrightarrow_b t'$ by contracting a redex of $\mathsf{rk}(\Phi)$, then t contains a sub-term $s \cdot [\overline{p} \,;\, \overline{\lambda x.q}] \cdot^{\Phi} [\overline{u} \,;\, \overline{\lambda y.r}]$ which is contracted to $s \cdot [\overline{p} \,;\, \overline{\lambda x.q \cdot^{\Phi} [\overline{u} \,;\, \overline{\lambda y.r}]}]$. The newly created redexes (if any) are all of $\mathsf{rk}(\Phi)$.

2. Suppose $\{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$. Then $q_i : \Phi$ and $s_k : A_k$ which is a sub-formula of Φ, as $\Phi = c(A_1,\ldots,A_n)$. If $q_i[x_i := s_k]$ is an introduction term, then either q_i is an introduction term itself or $q_i = x_i$ and s_k is an introduction term. The latter case can only occur if $s_k : \Phi$, but it is not, because its type is a sub-formula of Φ. So q_i is an introduction term. The case for $q_i[x_i := s_k]$ being an elimination term is similar. ◀

The Lemma states that both the newly created redexes due to $\longrightarrow_b$ and $\longrightarrow_{a2}$ are already "hidden" inside the term. We give a list of facts about redex creation and the ranks of redexes.

▶ Fact 44. **1.** A reduction step can produce more redexes either by (i) *copying existing redexes* or by (ii) *creating new redexes.* Copying occurs through substitution, in a reduction step $\longrightarrow_{a1}$ or $\longrightarrow_{a2}$.

2. Creating new redexes happens in either one of the following ways

a. When doing an $\longrightarrow_a$ step: in a sub-term $x \cdot [\overline{p} \,;\, \overline{\lambda y.q}]$, we substitute $\{\overline{s} \,;\, \overline{\lambda z.r}\}$ for x, creating a a-redex.

b. When doing an $\longrightarrow_a$ step: in a sub-term $x \cdot [\overline{p} \,;\, \overline{\lambda y.q}]$, we substitute $t \cdot [\overline{s} \,;\, \overline{\lambda z.r}]$ for x, creating a b-redex.

c. If $\{\overline{p, p_j} \,;\, \overline{\lambda x.q}\} \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r, \lambda y_\ell.r_\ell}] \longrightarrow_{a1} r_\ell[y_\ell := p_j]$ where this term occurs as a sub-term $- \cdot^{\Psi} [\ldots \,;\, \ldots]$ and $r_\ell[y_\ell := p_j] = \{\ldots \,;\, \ldots\}$, this creates a new a-redex.

d. If $\{\overline{p, p_j} \,;\, \overline{\lambda x.q}\} \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r, \lambda y_\ell.r_\ell}] \longrightarrow_{a1} r_\ell[y_\ell := p_j]$ where this term occurs as a sub-term $- \cdot^{\Psi} [\ldots \,;\, \ldots]$ and $r_\ell[y_\ell := p_j] = \ldots \cdot [\ldots \,;\, \ldots]$, this creates a new b-redex.

e. If $\{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$, this creates a new a-redex in case $q_i = \{\ldots \,;\, \ldots\}$.

f. If $\{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$, this creates a new b-redex in case $q_i = \ldots \cdot [\ldots \,;\, \ldots]$.

g. If $(t \cdot [\overline{p} \,;\, \overline{\lambda x.q}]) \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}] \longrightarrow_b t \cdot [\overline{p} \,;\, \overline{\lambda x.(q \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}])}]$, this creates a new a-redex (possibly more) in case $q_i = \{\ldots \,;\, \ldots\}$.

h. If $(t \cdot [\overline{p} \,;\, \overline{\lambda x.q}]) \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}] \longrightarrow_b t \cdot [\overline{p} \,;\, \overline{\lambda x.(q \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}])}]$, this creates a new b-redex (possibly more) in case $q_i = \ldots \cdot [\ldots \,;\, \ldots]$.

3. The first two cases of redex creation (**a** and **b**) create a new redex of *lower rank*, case **c** and **d** create a redex of unrelated rank, while the last four cases create a new redex of the same rank.

Note that in the cases **e** and **f** of Fact 44 we use the second part of Lemma 43.

The idea is to contract an innermost redex of highest rank of a term in b-normal form. The advantage of b-normal forms is that cases **c** and **d** of the Fact 44 do not occur. (Because in these cases, the term one starts with is not in b-normal form.)

▶ **Lemma 45.** *If f is a well-typed term in b-normal form that has one redex of maximum rank, say R, then f can be reduced to a term f' in b-normal form that has maximum rank below R.*

Proof. By induction on the size of f.

1. If $f = \{\overline{p} \,;\, \overline{\lambda x.q}\}$ or $f = x \cdot [\overline{p} \,;\, \overline{\lambda x.q}]$ or $f = \{\overline{p} \,;\, \overline{\lambda x.q}\} \cdot [\overline{s} \,;\, \overline{\lambda y.r}]$ and the redex of highest rank is inside $\overline{p}$, $\overline{q}$, $\overline{s}$ or $\overline{r}$, then we are done by the induction hypothesis.
2. Suppose $f = \{\overline{p} \,;\, \overline{\lambda x.q}\} \cdot^{\Phi} [\overline{s} \,;\, \overline{\lambda y.r}]$ is itself a redex of highest rank, $\mathsf{rk}(\Phi)$. We look at the possible ways in which a new redex may arise, following Fact 44. The cases **c**, **d**, **g** and **h** don't apply. The newly created redexes of cases **a** and **b** are of lower rank. In case **b**, the resulting term may not be in b-nf, but we can contract all the newly created b-redexes to obtain a b-normal form. According to Lemma 43, case (1), this does not create new redexes of higher rank, so we are done. This leaves cases **e** and **f**.
   - For case **e**: $f = \{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$ with $q_i = \{\ldots \,;\, \ldots\}$. By induction hypothesis, $q_i \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \twoheadrightarrow g$ for some g in b-normal form with all redexes of lower rank. (Note that $q_i \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$ is in b-normal form.) Then $q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \twoheadrightarrow g[x_i := s_k]$ and due to the fact that the type of s_k is a sub-formula of Φ, this only contains new redexes of lower rank, so we are done.
   - For case **f**: $f = \{\overline{p} \,;\, \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}] \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$ with $q_i = t \cdot [\overline{u} \,;\, \overline{\lambda z.v}]$. If we take g to be the b-normal form of $q_i \cdot^{\Phi} [\overline{s, s_k} \,;\, \overline{\lambda y.r}]$,

this term contains disjoint sub-terms of the shape $\lambda w.d \cdot^{\Phi} [\overline{s, s_k} ; \overline{\lambda y.r}]$ that all have one maximal redex of rank R and that have length smaller than the length of f. By induction hypothesis, these can all be reduced to terms with only redexes of lower rank. Having done this, we obtain g as a reduct of $q_i \cdot^{\Phi} [\overline{s, s_k} ; \overline{\lambda y.r}]$ that is in b-normal form and contains only redexes of rank lower than R. To obtain f', we notice that $f \longrightarrow_{a2} q_i[x_i := s_k] \cdot^{\Phi} [\overline{s, s_k} ; \overline{\lambda y.r}] \twoheadrightarrow g'[x_i := s_k]$, which only contains b-redexs of lower rank, so we can take f' to be the b-normal form of $g'[x_i := s_k]$. ◀

▶ **Theorem 46.** *For any set of connectives $\mathcal{C}$, the reduction $\longrightarrow_{ab}$ of the calculus $\lambda^{\mathcal{C}}$ is weakly normalizing and we have a procedure to compute a normal form for a well-typed term.*

Proof. We consider the following measure $\mathsf{m}(-)$ terms: $\mathsf{m}(t) := (R, m)$, where R is the maximal rank of a redex in t and m is the number of redexes of rank R in t. We consider this measure under the lexicographic ordering.

Given a term t, we first compute its b-normal form, t_1 and consider $\mathsf{m}(t_1) = (R, m)$. Then we pick p, an innermost redex of maximal rank inside t_1. Following Lemma 45, we reduce p to p', in which all redexes are of rank below R. We do this reduction on t_1, obtaining t_2. (So $t_1 \twoheadrightarrow t_2$.) Notice that $\mathsf{m}(t_1) > \mathsf{m}(t_2)$. We continue in this way, obtaining a normal form of t, because the lexicographic ordering is well-founded. ◀

▶ Fact 47. A useful fact about normal forms in $\lambda^{\mathcal{C}}$ is the following: If t is a term in normal form, then t is of either one of the following three forms

1. t is a variable,
2. $t = \{\overline{p} ; \overline{\lambda y.q}\}$, with all p_i and q_j in normal form,
3. $t = x \cdot [\overline{p} ; \overline{\lambda y.q}]$, with x a variable and all p_i and q_j in normal form.

5.3 Corollaries of normalization

▶ **Theorem 48.** *For any set of connectives $\mathcal{C}$, the calculus $\lambda^{\mathcal{C}}$ is consistent, that is: there are types A for which there is no closed term t with $\vdash t : A$.*

Proof. Take A to be a propositional variable and suppose $\vdash t : A$ with t in normal form. The three possible cases for t are given in Fact 47. The first and third case are impossible, because t cannot contain any free variable. The second case is impossible, because an introduction term is always of a composite type. ◀

The calculus (and logic) $\lambda^{\mathcal{C}}$ also satisfies the sub-formula property.

▶ **Theorem 49.** *Given a set of connectives $\mathcal{C}$, the calculus $\lambda^{\mathcal{C}}$ satisfies the* sub-formula property*, that is: if $\Gamma \vdash t : A$, then there is a term t' such that $\Gamma \vdash t' : A$ and all types of all sub-terms of t' are either sub-types of A or of some A_i for a declaration $x_i : A_i$ in Γ.*

Proof. If $\Gamma \vdash t : A$, then (by Theorem 46) there is a term t' in normal form with $\Gamma \vdash t' : A$. We use Fact 47 and prove by induction on t' that "all types of all sub-terms of t' are either sub-types of A or of some A_i for a declaration $x_i : A_i$ in Γ". For simplicity we abbreviate this property to "t' *satisfies the sub-type property for* $\Gamma; A$".

- $t' = x$, a variable. Then we are done.
- $t' = \{\overline{p} ; \overline{\lambda x.q}\}$, an introduction term. Then by induction hypothesis, all sub-terms of $\overline{p}$ satisfy the sub-type property for $\Gamma; A_i$ for some A_i which is a sub-type of A. For the $\lambda x_j.q_j$ in $\overline{\lambda x.q}$, we have $\Gamma, x_j : A_j \vdash q_j : A$ for some A_j which is a sub-type of A. By induction hypothesis, for all j, all sub-terms of q_j satisfy the sub-type property for

$\Gamma, x_j : A_j; A$. So all sub-terms of $\overline{\lambda x.q}$ satisfy the sub-type property for $\Gamma; A$ and we are done.

- $t' = x \cdot [\overline{p} \,;\, \overline{\lambda x.q}]$, an elimination term. Suppose $x : C$. Each p_i is of type B_i for some sub-type B_i of C, so the induction hypothesis yields that all sub-terms of $\overline{p}$ satisfy the sub-type property for $\Gamma; A$. For the $\lambda x_j.q_j$ in $\overline{\lambda x.q}$, we have $\Gamma, x_j : B_j \vdash q_j : A$ for some B_j which is a sub-type of C. By induction hypothesis, for all j, all sub-terms of q_j satisfy the sub-type property for $\Gamma, x_j : B_j; A$. So all sub-terms of $\overline{\lambda x.q}$ satisfy the sub-type property for $\Gamma; A$ and we are done.

◀

▶ **Theorem 50.** *In $\lambda^{\mathcal{C}}$, given a context Γ and a type D, the problem $\Gamma \vdash ? : D$ is decidable. That is, it is whether there is a term t for which $\Gamma \vdash t : D$.*

Proof. By Theorem 46 we can limit our search to a term in normal form. So we can restrict the elimination rules to the following restrictes case, where $\Phi = c(A_1, \ldots, A_n)$. (Compare with the original rules in Definition 17.)

$$\frac{x : \Phi \in \Gamma \quad \ldots \Gamma \vdash p_k : A_k \ldots \quad \ldots \Gamma, y_\ell : A_\ell \vdash q_\ell : D}{\Gamma \vdash x \cdot_r [\overline{p} \,;\, \overline{\lambda y.q}] : D}\,\text{el}$$

Now, given Γ and D, the following algorithm searches a term t in normal form with $\Gamma \vdash t : D$. (1) Check if $x : D \in \Gamma$ for some x and otherwise (2) try an introduction rule (in case D is composite) and (3) try an elimination rule for each $x : \Phi \in \Gamma$ with Φ a composite formula. In the recursive case, this gives finitely many possibilities to try and each try creates new goals of the form $\Gamma, y_j : A_j \vdash ? : D$ or of the form $\Gamma \vdash ? : A_i$ with A_j and A_i sub-formulas of Γ, D. This search terminates because the number of sub-formulas in the context increases (which is bound by the number of all sub-formulas of Γ, D), and otherwise the size of goal-formula decreases.

◀

As a corollary, we find that all the variants of the logical rules we have considered are decidable and consistent, simply because they are (with respect to derivability) equivalent to the set of rules for $\wedge, \vee, \to, \neg, \bot, \top$ that we extract from the truth tables, for which Theorems 48 and 50 apply. We can also say a bit more about the cut-elimination of these systems themselves: elimination of direct cuts is strongly normalizing, elimination of indirect cuts is strongly normalizing and we can also conclude weak normalization of the combined reduction.

▶ **Theorem 51.** *The reductions for the optimized rules of Definition 21, the parallel $\wedge$-elimination rule of Definition 23, the standard $\to$-introduction of Definition 26 and the traditional rule for $\neg$-elimination of Definition 29 are weakly normalizing.*

Proof. The proof follows the same argument as the proof of Theorem 46. The crucial Lemmas are Lemmas 45 and 43, which can be proved again with the reduction rules mentioned in the statement of Theorem 51 added. Furthermore, the reduction of indirect cuts, $\longrightarrow_b$ is strongly normalizing. (Corollary 33.) ◀

6 Conclusion and Further work

We have studied our general procedure for deriving intuitionistic natural deduction rules from truth tables, that we have published in [7]. We have compared our rules to more well-know sets of derivation rules and shown how other formats can be translated to ours while elimination of direct cuts is preserved. We have also made a proof-theoretic study of our general rules for an arbitrary set of connectives, establishing a general cut-elimination result.

The strength of our approach, as we see it, is that our deduction rules obey a specific format, making it easier to study. Also, our rules define every connective "in isolation": we have rules for each connective separately and don't translate one connective in terms of another.

The work described here raises new research questions. Most importantly: Is cut-elimination (the combination of eliminating direct and indirect cuts) strongly normalizing in general for an arbitrary set of connectives? We would believe so, but have not yet proved it. Techniques as in [9], where a this property is proved for intuitionistic logic, may be useful. Also the rules are not confluent in general, but one may wonder whether there is a certain condition that guarantees confluence. Another issue is to define cut-elimination for the classical case, and what is its connection is with known term calculi for classical logic, for example as studied in [13], [1] and [2].

Finally, we may wonder whether our research could contribute to the study of "harmony in logic", as first introduced by Prawitz [15] and further studied by various authors like [16, 22, 4, 3]. From our research, we would propose the following as a proper system for intuitionistic logic with "parallel elimination rules" that follow Prawitz [15] inversion principle. These rules are derived from the truth tables and optimized following Lemma 7, but not using Lemma 10. Compare with Definition 11.

▶ **Definition 52.** The *parallel elimination rules* for the intuitionistic propositional connectives $\wedge, \vee, \rightarrow, \neg, \bot$ and $\top$ are given below.

$$\frac{\vdash A \qquad \vdash B}{\vdash A \wedge B}\ \wedge\text{-in} \qquad \frac{\vdash A \wedge B \quad A \vdash D}{\vdash D}\ \wedge\text{-el}_1 \qquad \frac{\vdash A \wedge B \quad B \vdash D}{\vdash D}\ \wedge\text{-el}_2$$

$$\frac{\vdash A}{\vdash A \vee B}\ \vee\text{-in}_1 \qquad \frac{\vdash B}{\vdash A \vee B}\ \vee\text{-in}_2 \qquad \frac{\vdash A \vee B \quad A \vdash D \quad B \vdash D}{\vdash D}\ \vee\text{-el}$$

$$\frac{\vdash B}{\vdash A \to B}\ \to\text{-in}_1 \qquad \frac{A \vdash A \to B}{\vdash A \to B}\ \to\text{-in}_2 \qquad \frac{\vdash A \to B \quad \vdash A \quad B \vdash D}{\vdash D}\ \to\text{-el}$$

$$\frac{A \vdash \neg A}{\vdash \neg A}\ \neg\text{-in} \qquad \frac{\vdash \neg A \qquad \vdash A}{\vdash D}\ \neg\text{-el} \qquad \frac{}{\vdash \top}\ \top\text{-in} \qquad \frac{\vdash \bot}{\vdash D}\ \bot\text{-el}$$

7 References

References

1. Z. Ariola and H. Herbelin. Minimal classical logic and control operators. In *ICALP*, volume 2719 of *LNCS*, pages 871–885. Springer, 2003.
2. P.-L. Curien and H. Herbelin. The duality of computation. In *ICFP*, pages 233–243, 2000.
3. R. Dyckhoff. Some remarks on proof-theoretic semantics. In *Advances in Proof-Theoretic Semantics*, pages 79–93. Springer, 2016.
4. N. Francez and R. Dyckhoff. A note on harmony. *Journal of Philosophical Logic*, 41(3):613–628, 2012.

5 R.O. Gandy. An early proof of normalization by A.M. Turing. In J.P. Seldin and J.R. Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, page 453–455. Academic Press Limited, 1980.
6 G. Gentzen. Untersuchungen über das logische Schliessen. *Mathematische Zeitschrift*, pages 176–210, 405–431, 1935. English translation in [18].
7 H. Geuvers and T. Hurkens. Deriving natural deduction rules from truth tables. In *ICLA*, volume 10119 of *Lecture Notes in Computer Science*, pages 123–138. Springer, 2017.
8 J.-Y. Girard et al. *Proofs and types*, volume 7 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, Cambridge, 1989.
9 F. Joachimski and R. Matthes. Short proofs of normalization for the simply- typed lambda-calculus, permutative conversions and Gödel's T. *Arch. Math. Log.*, 42(1):59–87, 2003.
10 E.G.K López-Escobar. Standardizing the n systems of Gentzen. In *Models, Algebras and Proofs*, volume 203 of *Lecture Notes in Pure and Applied Mathematics*, page 411–434. Marcel Dekker Inc., New York, 1999.
11 P. Milne. Inversion principles and introduction rules. In *Dag Prawitz on Proofs and Meaning*, volume 7 of *Outstanding Contributions to Logic*, pages 189–224. Springer, 2015.
12 S. Negri and J. von Plato. *Structural Proof Theory*. Cambridge University Press, 2001.
13 M. Parigot. $\lambda\mu$-calculus: An algorithmic interpretation of classical natural deduction. In *LPAR*, volume 624 of *LNCS*, pages 190–201. Springer, 1992.
14 D. Prawitz. *Natural deduction: a proof-theoretical study*. Almqvist & Wiksell, 1965.
15 D. Prawitz. Ideas and results in proof theory. In J. Fenstad, editor, *2nd Scandinavian Logic Symposium*, pages 237–309. North-Holland, 1971.
16 S. Read. Harmony and autonomy in classical logic. *J. Philosophical Logic*, 29(2):123–154, 2000.
17 P. Schroeder-Heister. A natural extension of natural deduction. *J. Symb. Log.*, 49(4):1284–1300, 1984.
18 M.E. Szabo. *The Collected Papers of Gerhard Gentzen*. North-Holland, Amsterdam, 1969.
19 W.W. Tait. Intensional interpretations of functionals of finite type I. *J. Symb. Log.*, 32(2):198–212, 1967.
20 N. Tennant. Ultimate normal forms for parallelized natural deductions. *Logic Journal of the IGPL*, 10(3):299–337, 2002.
21 D. van Dalen. *Logic and structure (3. ed.)*. Universitext. Springer, 1994.
22 J. von Plato. Natural deduction with general elimination rules. *Arch. Math. Log.*, 40(7):541–567, 2001.