Classical Natural Deduction from Truth Tables

Herman Geuvers
Radboud University, Nijmegen, The Netherlands
Technical University Eindhoven, The Netherlands

Tonny Hurkens
Unaffiliated Researcher, Haps, The Netherlands

Abstract

In earlier articles we have introduced *truth table natural deduction* which allows one to extract natural deduction rules for a propositional logic connective from its truth table definition. This works for both intuitionistic logic and classical logic. We have studied the proof theory of the intuitionistic rules in detail, giving rise to a general Kripke semantics and general proof term calculus with reduction rules that are strongly normalizing. In the present paper we study the classical rules and give a term interpretation to classical deductions with reduction rules. As a variation we define a multi-conclusion variant of the natural deduction rules as it simplifies the study of proof term reduction. We show that the reduction is normalizing and gives rise to the sub-formula property. We also compare the logical strength of the classical rules with the intuitionistic ones and we show that if one non-monotone connective is classical, then all connectives become classical.

2012 ACM Subject Classification Theory of computation → Proof theory; Theory of computation → Type theory; Theory of computation → Constructive mathematics; Theory of computation → Functional constructs

Keywords and phrases Natural deduction, classical proposition logic, multiple conclusion natural deduction, proof terms, formulas-as-types, proof normalization, subformula property, Curry-Howard isomorphism

Digital Object Identifier 10.4230/LIPIcs.TYPES.2022.2

Acknowledgements We want to thank the reviewers for their valuable comments.

1 Introduction

Classically, the meaning of a propositional connective is fixed by its *truth table*. This immediately implies consistency, a decision procedure, completeness (with respect to Boolean algebras) for classical logic. Constructively, following the *Brouwer-Heyting-Kolmogorov* interpretation [17], the meaning of a connective is fixed by explaining what a *proof* is that involves the connective. Basically, this explains the *introduction rule(s)* for each connective, from which the elimination rules follow. This was first phrased like this by Prawitz in [14], who studied *natural deduction* in detail, including the reduction of proofs (deductions). By analyzing constructive proofs we then also get consistency (from proof normalization), a decision procedure (from the sub-formula property) and completeness (with respect to Heyting algebras and Kripke models).

In previous papers [6, 7], we have defined a general method to derive natural deduction rules for a connective from its truth table definition, which we have coined TT-ND, Truth Table Natural Deduction. This also works for constructive logic, which we have shown in detail by relating the method to Kripke semantics and by studying proof normalization. For classical logic, a similar method has been described by Milne [10]. The advantage is that the derived rules give natural deduction rules for a connective "in isolation", so without the need to explain a connective in terms of another (e.g. explaining the classical properties of implication using the double negation law). Also, this gives constructive rules for connectives that haven't been studied so far, like if-then-else and nand. These constructive connectives

are described and studied in detail in [7]. Finally, it allows to study various properties for a whole set of connectives all at once, like proof normalization and a generic (sound and complete) Kripke semantics. Proof normalization has been defined and studied in [7], where a weak normalization result is proven. Strong normalization has been proven in [8] and [1]. These proofs proceed by defining a proof-term calculus for TT-ND, following the Curry-Howard *proofs-as-terms* (and formulas-as-types) interpretation, and by defining a reduction relation on these proof terms.

These results all apply to the constructive case. In the present paper we study the classical case. We first prove some results only in terms of the logic. We show that for monotonic connectives (like $\vee$, $\wedge$), the classical and constructive rules are equivalent. This has also been shown in [18, 9], but we give a new (arguably simpler) proof. This shows that non-monotonic connectives "make classical logic classical". In our systems, the difference between constructive logic and classical logic for a connective c lies only in the introduction rules for c. To substantiate that non-monotonicity is crucial, we prove that if we allow one classic introduction rule for one non-monotonic connective, all connectives become classical. This implies, for example, that the classical rules for $\rightarrow$ imply the (seemingly stronger) classical rules for $\neg$ in presence of the constructive rules for $\neg$.

We also study proof reduction for classical logic derived from truth tables. To do this, we define a proof term calculus which now also has *conclusion variables*, similar in style with $\lambda\mu$ of Parigot [13] or variants of that studied by Ariola and Herbelin [2] and Curien and Herbelin [4]. We define various variants of this, depending on whether one has a single conclusion or multiple conclusions. We also define these as logics and we show – as was to be expected – that multiple conclusion intuitionistic TT-ND is logically equivalent to classical TT-ND. On the proof terms (that include conclusion variables and binding of them), we define a reduction relation that conforms with the reduction of deductions arising from detour elimination and with the goal to obtain a deduction that satisfies the sub-formula property. We describe this in detail for classical multi-conclusion logic, classical single-conclusion logic and intuitionistic single-conclusion logic. We define the reduction on proof-terms, show that it satisfies the subject reduction property and show that proof-terms in normal form satisfy the sub-formula property.

For the study of normalization, we introduce the unified framework of *Truth Table Logic* that arises quite naturally as a system unifying classical/intuitionistic multi-conclusion logic and classical/intuitionistic single conclusion logic. It works with *elimination patterns* and *introduction patterns* which can be combined to form proof terms. For this system we prove strong normalization, and from that, strong normalization for the original intuitionistic logic follows immediately. For the original versions of classical logic, the reduction is actually too "fine-grained" to derive strong normalization directly. But we can conclude that, if there is a proof term in classical logic (multi-conclusion or single conclusion), there is a proof-term in normal form of that same formula. From this we conclude that all logics satisfy the subformula property.

2 Natural Deduction from Truth Tables

We recap our earlier work on Truth Table Natural Deduction. To be able to reason generically about natural deduction rules, all our rules have a "standard form" that looks like this

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

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

1. $\Gamma \vdash A$: instead of proving D from Γ, we now need to prove A from Γ. We call A a Lemma.
2. $\Gamma, B \vdash D$: we are given extra data B to prove D from Γ. We call B a Casus.

Given this standard form of the rules, we don't have to give the Γ explicitly, as it can be retrieved, so we write

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

Various well-known deduction rules follow this format:

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

But there are others that do not follow this format, for example implication introduction:

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

In our set-up, implication introduction will break down in two rules, one introducing the implication, and one discharging the hypothesis. these together are equivalent to standard implication introduction.

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

▶ **Definition 1** (Natural Deduction rules from truth tables). *Let c be an n-ary connective c with truth table t_c.*

Each row of t_c gives rise to an elimination rule or an introduction rule for c. (We write $\Phi = c(A_1, \dots, A_n)$.)

$$\frac{\begin{array}{ccc|c} A_1 & \dots & A_n & \Phi \\ \hline p_1 & \dots & p_n & 0 \end{array}}{} \quad \mapsto \quad \frac{\vdash \Phi \dots \vdash A_i \ (\text{if } p_i = 1) \dots A_j \vdash D \ (\text{if } p_j = 0) \dots}{\vdash D}\ el$$

constructive *intro*

$$\frac{\begin{array}{ccc|c} A_1 & \dots & A_n & \Phi \\ \hline q_1 & \dots & q_n & 1 \end{array}}{} \quad \mapsto \quad \frac{\dots \vdash A_i \ (\text{if } q_i = 1) \dots A_j \vdash \Phi \ (\text{if } q_j = 0) \dots}{\vdash \Phi}\ in^i$$

classical *intro*

$$\frac{\begin{array}{ccc|c} A_1 & \dots & A_n & \Phi \\ \hline r_1 & \dots & r_n & 1 \end{array}}{} \quad \mapsto \quad \frac{\Phi \vdash D \dots \vdash A_i \ (\text{if } r_i = 1) \dots A_j \vdash D \ (\text{if } r_j = 0) \dots}{\vdash D}\ in^c$$

We call $\vdash \Phi$ (resp. $\Phi \vdash D$) the major premise *and the other hypotheses of the rule we call the* minor premises*. The minor premises are either a* Lemma*, A_i (if $p_i = 1$ or $q_i = 1$ or $r_i = 1$ in t_c), or a* Casus*, A_j (if $p_j = 0$ or $q_j = 0$ $r_j = 0$) in t_c.*

▶ **Definition 2** (Definition of the logics). *Given a set of connectives $\mathcal{C} := \{c_1, \dots, c_n\}$, we define the* intuitionistic *and* classical *natural deduction systems for $\mathcal{C}$, $\mathsf{IPC}_\mathcal{C}$ and $\mathsf{CPC}_\mathcal{C}$ as follows.*

- *Both $\mathsf{IPC}_\mathcal{C}$ and $\mathsf{CPC}_\mathcal{C}$ have an* axiom rule

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

- *Both $\mathsf{IPC}_\mathcal{C}$ and $\mathsf{CPC}_\mathcal{C}$ have the* elimination rules *for the connectives in $\mathcal{C}$.*

- $\mathsf{IPC}_{\mathcal{C}}$ *has the intuitionistic introduction rules for the connectives in* $\mathcal{C}$*.*
- $\mathsf{CPC}_{\mathcal{C}}$ *has the classical introduction rules for the connectives in* $\mathcal{C}$*.*

In [6], we have given a sound and complete a Kripke semantics for $\mathsf{IPC}_{\mathcal{C}}$. Briefly, a Kripke model is defined as usual and for w a world in the Kripke model, we define $[\![\varphi]\!]_w \in \{0,1\}$ with the meaning that $[\![\varphi]\!]_w = 1$ if and only if formula φ is true in world w. For $\varphi = c(\varphi_1, \ldots, \varphi_n)$, we define $[\![\varphi]\!]_w := 1$ if $t_c([\![\varphi_1]\!]_{w'}, \ldots, [\![\varphi_n]\!]_{w'}) = 1$ for each $w' \geq w$, where t_c is the truth table of c, and otherwise $[\![\varphi]\!]_w := 0$. Similarly, a sound and complete valuation semantics can be given for $\mathsf{CPC}_{\mathcal{C}}$, where a valuation is a map from the proposition letters to $\{0,1\}$ and the interpretation of composite formulas follows the truth table.

▶ **Example 3.** Constructive rules for $\wedge$ (3 elimination rules and one intro rule):

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

$$\frac{\vdash A \wedge B \quad A \vdash D \quad B \vdash D}{\vdash D}\ \wedge\text{-el}_{00} \qquad \frac{\vdash A \wedge B \quad A \vdash D \quad \vdash B}{\vdash D}\ \wedge\text{-el}_{01}$$

$$\frac{\vdash A \wedge B \quad \vdash A \quad B \vdash D}{\vdash D}\ \wedge\text{-el}_{10} \qquad \frac{\vdash A \quad \vdash B}{\vdash A \wedge B}\ \wedge\text{-in}_{11}$$

These rules can be shown to be equivalent to the well-known constructive rules. These rules can be optimized to the three rules we are familiar with.

▶ **Example 4.** Rules for $\neg$: 1 elimination rule and 1 introduction rule.

A	$\neg A$
0	1
1	0

Constructive:

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

Classical:

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

Using the classical rules for $\neg$, we show that $\neg\neg A \vdash A$ is derivable:

$$\dfrac{\dfrac{\neg\neg A, \neg A \vdash \neg\neg A \quad \neg\neg A, \neg A \vdash \neg A}{\neg\neg A, \neg A \vdash A}\ \neg\text{-el} \qquad \neg\neg A, A \vdash A}{\neg\neg A \vdash A}\ \neg\text{-in}^{c}$$

Simplifying the set of rules

There are various ways to optimize the rules, for example by taking two rules together in one that is equivalent. These have already been described and proven in [6], so we only recap the Lemmas here. To describe these, a first important operation is substituting one derivation on top of another, a kind of "cut operation".

▶ **Lemma 5** (Substituting a deduction in another [6]). *If* $\Gamma \vdash A$ *and* $\Delta, A \vdash B$*, then* $\Gamma, \Delta \vdash B$.

Proof. If Σ is a deduction of $\Gamma \vdash A$ and Π is a deduction of $\Delta, A \vdash B$, then we have the following deduction of $\Gamma, \Delta \vdash B$.

$$\begin{array}{c} \vdots\,\Sigma \\ \Gamma \vdash A \end{array} \quad \ldots \quad \begin{array}{c} \vdots\,\Sigma \\ \Gamma \vdash A \end{array}$$
$$\vdots\,\Pi$$
$$\Gamma, \Delta \vdash B$$

The idea here is that in Π, the leaves that derive $\Delta' \vdash A$ (for some Δ') as an (axiom) is replaced by a copy of the deduction Σ, which is also a deduction of $\Delta', \Gamma \vdash A$ (due to weakening). As contexts only grow if one goes upwards in the tree, we have $\Delta' \supseteq \Delta$, so this new derivation is well-formed. ◀

▶ **Lemma 6** ([6]). *A system with two deduction rules of the following form*

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

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

is equivalent to the system with these two rules replaced by

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

We don't repeat the proof, which is in [6], but we give an example.

▶ **Example 7.** The two rules for $\wedge$ from Example 3, ($\wedge$-el$_{00}$) and ($\wedge$-el$_{10}$) can be replaced by one rule:

$$\frac{\vdash A \wedge B \quad B \vdash D}{\vdash D}\ \wedge\text{-el}\_0$$

The intuition is that, as A can occur as a Lemma or as a Casus in an elimination rule where everything else is the same, we can just omit it.

As we can also leave rule ($\wedge$-el$_{00}$) in (as it is derivable), we can do the replacement again, and replace rules ($\wedge$-el$_{00}$) and ($\wedge$-el$_{01}$) by

$$\frac{\vdash A \wedge B \quad A \vdash D}{\vdash D}\ \wedge\text{-el}_0\_$$

▶ **Lemma 8.** *[6] A system with a deduction rule of the form to the left is equivalent to the system with this rule replaced by the rule on the right.*

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

Again, we don't repeat the proof, as it is in [6], but we give an example.

▶ **Example 9.** Having the optimized rules for $\wedge$ from Example 7, ($\wedge$-el$_0$) and ($\wedge$-el$_0_$), we can replaced them by the following rules, which are the well-known elimination rules for $\wedge$.

$$\frac{\vdash A \wedge B}{\vdash B}\ \wedge\text{-el}'\_0 \qquad \frac{\vdash A \wedge B}{\vdash A}\ \wedge\text{-el}'_0\_$$

The constructive connectives

We have already seen the $\wedge, \neg$ rules. The optimized rules for $\vee, \to, \top$ and $\bot$ we obtain are:

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

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

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

The rules for the classical $\to$ connective

The classical rules for implication are as follows. The elimination is rule is the same as the constructive one and we also have the first introduction rule $\to$-in$_1$. In addition we have the rule on the right. It is classical in the sense that one can derive Peirce's law from it (without using negation).

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

▶ **Example 10.** We give a classical derivation of Peirce's law, using only the classical rules for $\to$.

$$\dfrac{\dfrac{A \vdash A}{A \vdash ((A \to B) \to A) \to A} \qquad \dfrac{\dfrac{\dfrac{(A \to B) \to A \vdash (A \to B) \to A \quad A \to B \vdash A \to B}{A \to B, (A \to B) \to A \vdash A}}{A \to B, (A \to B) \to A \vdash ((A \to B) \to A) \to A}}{A \to B \vdash ((A \to B) \to A) \to A}\ \to\text{-in}_2^c}{\vdash ((A \to B) \to A) \to A}$$

3 Monotone and non-monotone connectives

▶ **Definition 11.** *A connective c is monotone if its truth table t_c is a monotone function from $\{0,1\}^n$ to $\{0,1\}$ with respect to the ordering $0 \leq 1$.*

Of the standard connectives, $\vee$ and $\wedge$ are monotone, while $\neg$, $\to$ and $\leftrightarrow$ are non-monotone. For monotone connectives, the constructive and classical rules are equivalent. (So this holds for $\wedge$, $\vee$.) For the non-monotone connectives like $\to$ and $\neg$, this is not the case, which is well-known, as Example 10 shows. There is an even stronger result: the classical intro rule for the one ($\to$ or $\neg$) implies the classical intro rule for the other ($\neg$ or $\to$). This hold in general: if we add one classical introduction rule for one non-monotone connective, then all non-monotone connectives are classical.

▶ **Proposition 12.** *For c monotone, the classical and constructive derivation rules are equivalent.*

Proof. Let c be a monotone connective, say with an introduction rule derived from the truth table row $r = (r_1, \ldots, r_i, \ldots, r_n|1)$. If $r_i = 0$, then we also have the truth table row $r' = (r_1, \ldots, 1, \ldots, r_n|1)$. That is, we have rows $(r_1, \ldots, 0, \ldots, r_n|1)$ and $(r_1, \ldots, 1, \ldots, r_n|1)$. Now, the first lemma for simplifying the rules (Lemma 6) says that the i-th minor premise is immaterial for the rule. This reasoning applies to every 0-entry in row r, so we can eliminate all 0-s from r and we obtain an equivalent introduction rule without any Casus, which therefore looks like this:

$$\frac{\Phi \vdash D \quad \vdash A_1 \ \ldots \ \vdash A_m}{\vdash D}$$

Now, by the second lemma for simplifying rules, Lemma 8, this rule is equivalent to

$$\frac{\vdash A_1 \ \ldots \ \vdash A_m}{\vdash \Phi}$$

which is the constructive introduction rule for c. So for c monotone, a classical introduction rule is equivalent to a constructive one. ◀

3.1 For non-monotone connectives, one classical introduction suffices

We now show that, if we have a set of connectives $\mathcal{C}$ and $c \in \mathcal{C}$ is non-monotone, then adding one classical introduction rule for c makes the whole logic classical. So, if we add this one classical introduction rule, we can derive the other classical introduction rules for c and also the classical introduction rules for all other connectives d. In case d is monotone, this doesn't add anything, because of Proposition 12. But, for example in the case of $\neg$ and $\to$, which are both non-monotone, this shows that the classical rule for $\to$ can be derived from the classical rule for $\neg$, and vice versa, the classical rule for $\neg$ can be derived from the classical rule for $\to$. The first might be expected, but the second maybe not, as one might have the impression that the "double negation law" is really stronger then the classical rules for $\to$. Theorem 13 below is even more general, as it says that all non-monotonic classical connectives are equally strong.

It should be noted that, when we talk about a non-monotonic connective c, say of arity n, we should actually speak about a "non-monotonic pattern in t_c", which is an index i ($1 \leq i \leq n$) and a sequence $a_1, \ldots, a_{i-1}, 1, a_{i+1}, \ldots, a_n$ such that $t_c(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n) = 1$ and $t_c(a_1, \ldots, a_{i-1}, 1, a_{i+1}, \ldots, a_n) = 0$. When we say we "add a classical introduction rule for this non-monotone c", we mean to add a classical introduction rule for a line $(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n)$ that is part of such a pattern. (There may be other lines $\overrightarrow{b}$ in the truth table of c, with $t_c(\overrightarrow{b}) = 0$, that are not part of such a pattern; making the introduction rule for such a $\overrightarrow{b}$ classical leaves the whole system intuitionistic.) We will not use the terminology of "non-monotonic pattern in t_c", to keep the text simple, and for $\neg$, $\to$ etcetera it isn't relevant. But in the proof we will start from such a situation.

▶ **Theorem 13.** *Let $\mathcal{C}$ be a set of connectives and let $c \in \mathcal{C}$ be non-monotone. If we add one classical introduction rule for c, we can derive the classical rules for all connectives.*

Proof. To simplify the presentation we do not consider the situation where we have optimized rules, but the proof goes through basically unaltered. Let c be an n-ary connective which is not monotone. Then there are rows $r = (r_1 \ldots r_n|1)$ and $s = (s_1 \ldots s_n|0)$ which only differ at position f and such that $r_f = 0$ and $s_f = 1$. So the corresponding classical introduction and elimination rules for c have the same minor premises, except for position f where the introduction rule (based on r) has a Casus while the elimination rule (based on s) has a Lemma.

We want to show that *any* other classical introduction rule, say the one for m-ary connective d based on a row $t = (t_1 \dots t_m|1)$, can be derived from this particular classical introduction rule for c based on row $r = (r_1 \dots r_n|1)$, by just using the constructive introduction rules and the elimination rules of the connectives c and d. Note that c and d need not be different connectives and the language need not contain any other connective.

The problem can be stated more precisely as follows. We have the following rows in the truth tables.

$r = (r_1 \dots r_n|1)$ for n-ary c r_k ranges over the 1-entries, r_ℓ ranges over the 0-entries,
$s = (s_1 \dots s_n|0)$ for n-ary c s_g ranges over the 1-entries, s_h ranges over the 0-entries,
$t = (t_1 \dots t_m|1)$ for m-ary d t_i ranges over the 1-entries, t_j ranges over the 0-entries.

Based on these rows, we have the following rules, where D and the As are arbitrary.

$$\frac{c(A_1, \dots, A_n) \vdash D \qquad \vdash A_k \quad (\text{for } r_k = 1) \qquad A_\ell \vdash D \quad (\text{for } r_\ell = 0)}{\vdash D}\ c\text{-intro based on } r$$

$$\frac{\vdash c(A_1, \dots, A_n) \qquad \vdash A_g \quad (\text{for } s_g = 1) \qquad A_h \vdash D \quad (\text{for } s_h = 0)}{\vdash D}\ c\text{-elim based on } s$$

$$\frac{\vdash A_i \quad (\text{for } t_i = 1) \qquad A_j \vdash d(A_1, \dots, A_m) \quad (\text{for } t_j = 0)}{\vdash d(A_1, \dots, A_m)}\ d\text{-intro based on } t, \text{intuitionistic}$$

Fix the formulas $\Phi = d(C_1, \dots, C_m)$ and D (where $C_1, \dots, C_m, D$ are arbitrary). We need to show that the following is derivable (without using classical introduction for d):

$$\frac{\Phi \vdash^1 D \qquad \vdash^2 C_i \quad (\text{for } t_i = 1) \qquad C_j \vdash^3 D \quad (\text{for } t_j = 0)}{\vdash D}$$

We have marked, using $\vdash^1$, $\vdash^2$ and $\vdash^3$, the hypotheses that we will need to derive $\vdash D$ so we can easily refer to them.

Solution from any extra assumption

We first show that we can derive our result D from an arbitrary additional assumption.

▷ Claim 14. For any formula X, we can derive $X \vdash D$ (under the assumptions laid out so far).

Proof of the Claim. Let X be any formula. Define $\Psi = c(B_1, \dots, B_n)$ where:

- $B_u = X$ if $r_u = s_u = 1$,
- $B_u = D$ if $r_u = 0$ and $s_u = 1$ (so this is the case where $u = f$, the only place where rows r and s differ),
- $B_u = \Phi$ if $r_u = s_u = 0$.

We now have the following derivation of $X \vdash D$ (where we still have to create a derivation of the major premise, $X, \Psi \vdash^* D$).

$$\frac{X, \Psi \vdash^* D \qquad X \vdash B_k \quad (r_k = 1) \qquad B_f \vdash D \qquad B_\ell \vdash D \quad (r_\ell = 0)}{X \vdash D}\ c\text{-intro based on } r$$

Note that the minor premises of this rule are all derivable:

- For $r_k = 1$, we have $B_k = X$ and we have $X \vdash X$,
- For $r_\ell = 0$ with $\ell = f$, we have $B_f = D$ and we have $D \vdash D$,
- For $r_\ell = 0$ with $\ell \neq f$, we have $B_\ell = \Phi$ and we have $\Phi \vdash^1 D$, one of our hypotheses.

We now show the derivability of the major premise, $X, \Psi \vdash^* D$, by giving a derivation of $X, \Psi \vdash \Phi$, which, combined with $\Phi \vdash^1 D$, gives $X, \Psi \vdash D$. Here is the derivation:

$$\dfrac{\vdash^2 C_i \text{ (for } t_i = 1) \qquad \dfrac{\Psi \vdash \Psi \quad X, \Psi \vdash B_g \text{ (for } s_g = 1) \quad C_j \vdash B_f \quad B_h \vdash \Phi \text{ (for } s_h = 0)}{X, \Psi, C_j \vdash \Phi \text{ (for } t_j = 0)}\, c\text{-el}}{\dfrac{X, \Psi \vdash \Phi}{X, \Psi \vdash D} \text{ by } \Phi \vdash^1 D}\, d\text{-in}$$

Note that the minor premises of this rule are all derivable:

- For $s_g = 1$ and $r_g = 1$, we have $B_g = X$ and we have $X \vdash X$,
- For $s_g = 1$ with $g = f$, we have $B_f = D$ and we have $C_j \vdash^3 D$, one of our hypotheses,
- For $s_h = 0$, we have $B_h = \Phi$ and we have $\Phi \vdash \Phi$.

So we have shown that, given $\Phi \vdash^1 D$, $\vdash^2 C_i$ (for $t_i = 1$) and $C_j \vdash^3 D$ for $t_j = 0$, we can derive $X \vdash D$ for any formula X. This proves our Claim 14. ◁

Full Solution. If we can find a formula X such that $\vdash X$ then in combination with Claim 14, $X \vdash D$ we get $\vdash D$ so we are done. If we have some standard connectives in our language, then we can take any theorem for X, like $\top$, $\neg\bot$ or $A \to A$. If we happen to be inside a non-empty Γ, we can take X to be any formula in Γ. But in general, we don't have a Γ and we cannot assume any other connectives than c and d (where c and d may even be the same).

If the introduction rule for d that is based on row t has at least one Lemma (C_i), then we can take for X any such C_i. So if in $t = (t_1 \dots t_m|1)$ we have $t_i = 1$ somewhere, we are done.

Now suppose that $t = (0 \dots 0|1)$, so each minor premise is a Casus in the introduction rule for d based on t that we are considering. Let, for A a formula, $\delta(A)$ denote the formula $d(A, \dots, A)$. Then constructive d-intro gives

$$\dfrac{A \vdash \delta(A)}{\vdash \delta(A)}\, d\text{-intro based on } t, \text{intuitionistic}$$

Let's denote by v the row in the truth table for d with just 1s as arguments: either $v = (1 \dots 1|1)$ or $v = (1 \dots 1|0)$.

case $v = (1 \dots 1|1)$.

Then the constructive introduction rule derived from v directly gives $A \vdash \delta(A)$ so we are done: take $X := \delta(A)$ (for arbitrary A), then $\vdash X$ is shown by:

$$\dfrac{\dfrac{A \vdash A}{A \vdash d(A, \dots, A)}\, d\text{-intro based on } v, \text{intuitionistic}}{\vdash d(A, \dots, A)}\, d\text{-intro based on } t, \text{intuitionistic}$$

case $v = (1 \dots 1|0)$.

Now take $X := \delta(\delta(D))$, where D is the formula that we want to prove $\vdash D$ for.

From row v we have an elimination rule for d, in particular we have, for any E:

$$\dfrac{\vdash \delta(D) \qquad \vdash D}{\vdash E}\, d\text{-elim based on } v$$

We take $\delta(\delta(D))$ for E and we have the following derivation of $\delta(\delta(D))$ (and we are done). Note that $\delta(D) \vdash D$ holds by our earlier Claim 14, because $X \vdash D$ for any X.

$$\dfrac{\dfrac{\delta(D) \vdash \delta(D) \qquad \delta(D) \vdash D}{\delta(D) \vdash \delta(\delta(D))}\, d\text{-elim based on } v}{\vdash \delta(\delta(D))}\, d\text{-intro based on } t, \text{intuitionistic}$$

This completes the proof of Theorem 13. ◀

4 Variants of Classical Natural Deduction

There are various ways to move from constructive logic to classical logic. In sequent calculus, this is done by considering multi-conclusion judgments of the form $\Gamma \vdash \Delta$, where Δ is a sequence (or finite set) of formulas, with the intuitive meaning that the conjunction of the formulas in Γ implies the disjunction of the formulas in Δ. We can also make our truth table natural deduction "multi-conclusion", which helps to clarify the connection between the various systems, but more importantly, it is helpful in describing the proof reduction of classical logic. We first introduce these logics and prove their connection.

▶ **Definition 15.** *Let $\mathcal{C}$ be a set of connectives with their associated truth tables and let Γ and Δ be finite sets of formulas over $\mathcal{C}$. We define intuitionistic and classical multi-conclusion logic by considering the following derivation rules.*

$$\frac{}{\Gamma, A \vdash A, \Delta}\ (axiom)$$

$$\frac{\Gamma \vdash \Phi, \Delta_0 \ldots \Gamma \vdash A_k, \Delta_k \ldots \quad \ldots \Gamma, A_\ell \vdash \Delta_\ell}{\Gamma \vdash \Delta}\ (el)\ \text{if } \Delta \supseteq \Delta_0, \Delta_k, \Delta_\ell$$

$$\frac{\ldots \Gamma \vdash A_i, \Delta_i \ldots \quad \ldots \Gamma, A_j \vdash \Phi, \Delta_j \ldots}{\Gamma \vdash \Phi, \Delta}\ (in\text{-}int)\ \text{if } \Delta \supseteq \Delta_i, \Delta_j$$

$$\frac{\Gamma, \Phi \vdash \Delta_0 \ldots \Gamma \vdash A_i, \Delta_i \ldots \quad \ldots \Gamma, A_j \vdash \Delta_j \ldots}{\Gamma \vdash \Delta}\ (in\text{-}class)\ \text{if } \Delta \supseteq \Delta_0, \Delta_i, \Delta_j$$

In each rule, $\Phi = c(A_1, \ldots, A_n)$ for some connective c. The A_i and A_j range over the entries with $r_i = 1$ and $r_j = 0$ of some 1-row r for c. The A_k and A_ℓ range over the entries with $r'_k = 1$ and $r'_\ell = 0$ of some 0-row r' for c.

We define the following logics by having the rules (axiom) and (el) and one introduction rule:

1. Int-mc, *intuitionistic multi-conclusion logic, has introduction rule (in-int).*
2. Class-mc, *classical multi-conclusion logic, has introduction rule (in-class).*

For the multi-conclusion logics, we can also let the Δ be a constant in the rules (just like the Γ), as the (axiom) rule gives weakening anyway. We get the original truth table natural deduction rules by restricting these derivation rules to judgments with a single conclusion:

- *(axiom-sc) is rule (axiom) with Δ empty*
- *(el-sc) is rule (el) with Δ a singleton, Δ_0 and the Δ_k empty, and the Δ_ℓ equal to Δ*
- *(in-int-sc) is rule (in-int) with Δ empty*
- *(in-class-sc) is rule (in-class) with Δ a singleton, Δ_0 and the Δ_j equal to Δ, and the Δ_i empty*

We define the original truth table natural deduction logics in this format by having the rules (axiom-sc) and (el-sc) and one introduction rule:

1. Int, *intuitionistic single-conclusion logic, has introduction rule (in-int-sc).*
2. Class, *classical single-conclusion logic, has introduction rule (in-class-sc).*

▶ **Proposition 16.**

1. Int-mc, Class *and* Class-mc *are equivalent in the sense that*
   $$\Gamma \vdash \Phi \text{ in Class} \quad \Leftrightarrow \quad \Gamma \vdash \Phi \text{ in Class-mc}$$
   $$\Gamma \vdash \Delta \text{ in Int-mc} \quad \Leftrightarrow \quad \Gamma \vdash \Delta \text{ in Class-mc}$$
2. Int *is really weaker than the other 3 systems if $\mathcal{C}$ contains a non-monotone connective.*

Proof.

1. The proof of $\Gamma \vdash \Phi$ in Class $\Leftrightarrow$ $\Gamma \vdash \Phi$ in Class-mc is by showing completeness for both Class and Class-mc in the standard way (using maximally consistent extensions of sets of formulas) with respect to a valuation semantics. For Class-mc, one proves $\Gamma \vdash \Delta \Leftrightarrow \Gamma \vDash \Delta$, where $\Gamma \vDash \Delta$ is defined as: for all $v : \mathsf{At} \to \{0,1\}$, if $\forall \varphi \in \Gamma(v(\varphi) = 1)$, then $\exists \psi \in \Delta(v(\psi) = 1)$. One proves the same result for Class, from which the first equivalence in the Proposition follows.
   For the proof of $\Gamma \vdash \Delta$ in Int-mc $\Leftrightarrow$ $\Gamma \vdash \Delta$ in Class-mc, the $\Rightarrow$ is immediate, while for the $\Leftarrow$ we show that the classical introduction rule (in-class) is derivable in Int-mc: suppose we have $\Gamma, \Phi \vdash \Delta_0$, $\Gamma \vdash A_i, \Delta_i$ (for all appropriate i, according to row r in truth table t_c) and $\Gamma, A_j \vdash \Delta_j$ (for all appropriate j) in Int-mc. We also have $\Gamma, A_j \vdash \Phi, \Delta_j$, so we can apply the intuitionistic introduction rule (in-int) to conclude $\Gamma \vdash \Phi, \Delta$ for $\Delta \supseteq \Delta_i, \Delta_j$. We also have $\Gamma, \Phi \vdash \Delta_0$, so by Substitution Lemma (Lemma 5 extends directly to the case for Int-mc and Class-mc), we conclude $\Gamma \vdash \Delta, \Delta_0$ (for $\Delta \supseteq \Delta_i, \Delta_j$) and we are done.
2. For well-known non-monotone connectives like $\neg$ and $\to$, it is known that Int is really weaker than Class, e.g. one can prove Peirce's law and $\neg\neg A \vdash A$ in Class, which cannot be proven in Int, as one can show by constructing a Kripke counter-model. This also implies that for other non-monotone connective, say c of arity n, the classical rules are really stronger: with the classical rules for c and the intuitionistic rules for $\neg$ one can prove $\neg\neg A \vdash A$. (This follows directly from Theorem 13.) With the intuitionistic rules for c and $\neg$ one cannot prove $\neg\neg A \vdash A$, as the Kripke semantics for intuitionistic truth table natural deduction shows, see [6]. ◀

5 Proof terms for natural deduction

In earlier articles, we gave proof terms for natural deductions. This simplified the study of proof normalization. We followed the Curry-Howard formulas-as-types (and proofs-as-terms) paradigm. We defined systems with judgments $\Gamma \vdash t : B$, where B is a formula, Γ is a set of declarations $\{x_1 : A_1, \ldots, x_n : A_n\}$, 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. In these systems, the type of a proof term t is the conclusion B of the proof and the types of the free variables x_i of t are the assumptions A_i of the proof. A Lemma D (a sub-proof of formula D as part of the proof of B) is represented by a sub-term of t of type D. A Casus C (a part of the proof that uses C as extra assumption) is represented by a λ-abstraction over a variable of type C.

In this section, we first define proof term calculi for the logics we have already discussed, and also for some logics we have studied in earlier work, but now with both variables for assumptions (hypotheses) and for conclusions (goals).

By using (abstractions over) conclusion variables, we can distinguish between terms like x (representing an assumption A), $\gamma \cdot x$ (representing a proof of $A \vdash A$) and $\mu\gamma : A.\gamma \cdot x$ (representing a Lemma A). It is also very useful to represent multi-conclusion deductions.

5.1 Proof terms for truth table natural deduction with conclusion variables

We define proof term calculi for TT-ND with conclusion variables for various logics:

1. Int, intuitionistic logic,
2. Class, classical logic,
3. Int-mc, intuitionistic multi-conclusion logic,
4. Class-mc, classical multi-conclusion logic,

The judgments of these calculi will be of the form

$$t : (\Gamma \vdash \Delta)$$

where t denotes the proof term, Γ contains the labelled assumptions, typically $\Gamma = x_1, : A_1, \ldots, x_n : A_n$, and Δ contains the labelled conclusions, typically $\Delta = \alpha_1 : B_1, \ldots, \alpha_m : B_m$.

Such a judgment expresses that the sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ is derivable in the corresponding logic. As in sequent calculus, this can informally be read as "t is a proof of the disjunction of the B_j from the conjunction of the A_i". In case we have single conclusion logic Δ is a singleton, but still with a labelled conclusion, so $\Delta = \alpha : B$. We use λ to abstract over assumption variables and μ to abstract over conclusion variables. Each free variable of t is declared in Γ or Δ. Each proof term consists of two parts (separated by $\cdot$). Conclusion variables only occur as left part of a (nested) proof term and assumption variables only as right part.

▶ **Definition 17.** *Let $\mathcal{C}$ be a set of connectives with their associated truth tables and let Γ be an assumption context and Δ a conclusion context for formulas over $\mathcal{C}$. We consider the following derivation rules for terms, where the syntactic class of terms is as follows.*

$$\begin{aligned} t \quad ::= \quad & \alpha \cdot x \mid (\mu\alpha : A.t) \cdot [\overline{\mu\beta : B.t'} \; ; \; \overline{\lambda z : C.t''}]_r \\ & \mid (\lambda x : A.t) \cdot \{\overline{\mu\beta : B.t'}; \overline{\lambda z : C.t''}\}_r \mid \gamma \cdot \{\overline{\mu\beta : B.t'}; \overline{\lambda z : C.t''}\}_r \end{aligned}$$

$$\frac{}{\alpha \cdot x : (\Gamma, x : A \vdash \alpha : A, \Delta)} \; (axiom)$$

$$\frac{t : (\Gamma \vdash \beta : \Phi, \Delta_0) \ldots p_k : (\Gamma \vdash \alpha_k : A_k, \Delta_k) \ldots \quad \ldots q_\ell : (\Gamma, y_\ell : A_\ell \vdash \Delta_\ell)}{(\mu\beta : \Phi.t) \cdot [\overline{\mu\alpha_k : A_k.p_k} \; ; \; \overline{\lambda y_\ell : A_\ell.q_\ell}]_{r'} : (\Gamma \vdash \Delta)} \; (el)^*$$

$$\frac{\ldots p_i : (\Gamma \vdash \alpha_i : A_i, \Delta_i) \ldots \quad \ldots q_j : (\Gamma, y_j : A_j \vdash \gamma : \Phi, \Delta_j) \ldots}{\gamma \cdot \{\overline{\mu\alpha_i : A_i.p_i}; \overline{\lambda y_j : A_j.q_j}\}_r : (\Gamma \vdash \gamma : \Phi, \Delta)} \; (in\text{-}int)^{**}$$

$$\frac{t : (\Gamma, x : \Phi \vdash \Delta_0) \ldots p_i : (\Gamma \vdash \alpha_i : A_i, \Delta_i) \ldots \quad \ldots q_j : (\Gamma, y_j : A_j \vdash \Delta_j) \ldots}{(\lambda x : \Phi.t) \cdot \{\overline{\mu\alpha_i : A_i.p_i}; \overline{\lambda y_j : A_j.q_j}\}_r : (\Gamma \vdash \Delta)} \; (in\text{-}class)^{***}$$

In each rule, $\Phi = c(A_1, \ldots, A_n)$ for some connective c. Each r is a 1-row for connective c, where A_i ranges over the entries of r with $r_i = 1$ and A_j ranges over the entries with $r_j = 0$. Each r' is a 0-row for c and A_k ranges over the entries of r' with $r'_i = 1$ and A_ℓ ranges over the entries with $r'_\ell = 0$.

Just as in the definition of the logics (Definition 15), each term rule has the same side condition on the Δ in the conclusion: it should be a superset of the Δ's in the hypotheses. In particular, this means that * represents $\Delta \supseteq \Delta_0, \Delta_k, \Delta_\ell$ where k and ℓ range over a number of Δ's, ** represents $\Delta \supseteq \Delta_i, \Delta_j$ where i and j range over a number of Δ's, *** represents $\Delta \supseteq \Delta_0, \Delta_i, \Delta_j$ where i and j range over a number of Δ's. We define the multi-conclusion term calculi by having the rules (axiom) and (el) and one introduction rule:

1. Int-mc, *intuitionistic multi-conclusion calculus, has introduction rule (in-int).*
2. Class-mc, *classical multi-conclusion calculus, has introduction rule (in-class).*

Each rule for terms can be restricted to judgments $t : (\Gamma \vdash \Delta)$ in which Δ is a singleton, just like the corresponding derivation rule for sequents in Definition 15. We define the single-conclusion term calculi by having the rules (axiom-sc) and (el-sc) and one introduction rule:

1. Int, *intuitionistic single-conclusion calculus, has introduction rule (in-int-sc).*
2. Class, *classical single-conclusion calculus, has introduction rule (in-class-sc).*

We want to single out specific sub-term patterns that occur in the elimination rule and the introduction rule.

▶ **Definition 18.** *An* intro pattern *is a sub-term of the form* $\{\overline{\mu\alpha_i : A_i.p_i}; \overline{\lambda y_j : A_j.q_j}\}_r$. *An* elim pattern *is a sub-term of the form* $[\overline{\mu\alpha_k : A_k.p_k}\ ;\ \overline{\lambda y_\ell : A_\ell.q_\ell}]_{r'}$.

We say that the intro pattern $\{\overline{\mu\alpha_i : A_i.p_i}; \overline{\lambda y_j : A_j.q_j}\}_r$ *is* well-typed *in* $\Gamma \vdash \Delta$, *if for all* i, $p_i : (\Gamma \vdash \alpha_i : A_i, \Delta_i)$ *for some* Δ_i *and for all* j, $q_j : (\Gamma, y_j : A_j \vdash \gamma : \Phi, \Delta_j)$ *for some* Δ_j *and all* $\Delta_i, \Delta_j \subseteq \Delta$. *(And all this is relative to line* r *in the truth table for* c, *where* $\Phi = c(A_1, \ldots, A_n)$.*)*

We say that the elim pattern $[\overline{\mu\alpha_k : A_k.p_k}\ ;\ \overline{\lambda y_\ell : A_\ell.q_\ell}]_{r'}$ *is* well-typed *in* $\Gamma \vdash \Delta$, *if for all* k, $p_k : (\Gamma \vdash \alpha_k : A_k, \Delta_k)$ *for some* Δ_k *and for all* ℓ, $q_\ell : (\Gamma, y_\ell : A_\ell \vdash \Delta_\ell)$ *for some* Δ_k *and all* $\Delta_k, \Delta_\ell \subseteq \Delta$. *(And all this is relative to line* r' *in the truth table for some* c.*)*

▶ **Example 19.** The derivation rules for negation are as follows in the classical multi-conclusion logic. We omit the Γ and Δ.

$$\frac{t : (x : \neg B \vdash \alpha : D) \qquad q : (y : B \vdash \alpha : D)}{(\lambda x : \neg B.t) \cdot \{; \lambda y : B.q\}_r : (\vdash \alpha : D)}\ \text{in}\neg \qquad \frac{t : (\vdash \alpha : \neg B) \qquad q : (\vdash \beta : B)}{(\mu\alpha : \neg B.t) \cdot [\mu\beta : B.q\ ;\ ]_{r'} : (\vdash)}\ \text{el}\neg$$

As logics, the relation between these calculi has been given in Proposition 16. We can also give the relation between the term calculi.

▶ **Lemma 20.** *The calculus* Int *is a subsystem of* Int-mc *and the calculi* Class *and* Int-mc *are subsystems of* Class-mc*:*

$$\begin{array}{rcccl} \mathsf{Int} & \subseteq & \mathsf{Int\text{-}mc} & \subseteq & \mathsf{Class\text{-}mc} \\ & & \mathsf{Class} & \subseteq & \mathsf{Class\text{-}mc} \end{array}$$

where Int $\subseteq$ Int-mc *and* Class $\subseteq$ Class-mc *are by the identity on terms and* Int-mc $\subseteq$ Class-mc *is by interpreting*

$$[\![\gamma \cdot \{\overline{\mu\alpha_i : A_i.p_i}; \overline{\lambda y_j : A_j.q_j}\}_r]\!] := (\lambda x : \Phi.\gamma \cdot x) \cdot \{\overline{\mu\alpha_i : A_i.[\![p_i]\!]}; \overline{\lambda y_j : A_j.[\![q_j]\!]}\}_r$$

Proof. The proof is by a straightforward induction on the derivation, using the fact that all systems have weakening: if $t : (\Gamma \vdash \Delta)$ and $\Gamma \subseteq \Gamma'$, $\Delta \subseteq \Delta'$, then $t : (\Gamma' \vdash \Delta')$. ◀

▶ **Example 21.** We consider a proof of the double negation law in classical multi-conclusion calculus, given by the proof term $t : (z : \neg\neg A \vdash \alpha : A)$, where

$$t \quad := \quad (\mu\gamma : \neg\neg A.\gamma \cdot z) \cdot [\mu\beta : \neg A.(\lambda y : \neg A.\beta \cdot y) \cdot \{; \lambda x : A.\alpha \cdot x\}_r\ ;\ ]_{r'}$$

Observe that t contains a sub-term $(\lambda y : \neg A.\beta \cdot y) \cdot \{; \lambda x : A.\alpha \cdot x\}_r$, of type $(\vdash \beta : \neg A, \alpha : A)$, with two different free conclusion variables. Such proof terms, having multiple free conclusion variables, can only occur in multiple-conclusion calculus.

We give the derivation of t, using the rules given in Example 19. Except for the last line, we omit the declarations $z : \neg\neg A$ and $\alpha : A$ from the context.

$$\frac{\gamma \cdot z : (\vdash \gamma : \neg\neg A) \qquad \dfrac{\beta \cdot y : (y : \neg A \vdash \beta : \neg A) \quad \alpha \cdot x : (x : A \vdash \beta : \neg A)}{(\lambda y : \neg A.\beta \cdot y) \cdot \{; \lambda x : A.\alpha \cdot x\}_r : (\vdash \beta : \neg A)}\ \text{in}\neg}{(\mu\gamma : \neg\neg A.\gamma \cdot z) \cdot [\mu\beta : \neg A.(\lambda y : \neg A.\beta \cdot y) \cdot \{; \lambda x : A.\alpha \cdot x\}_r\ ;\ ]_{r'} : (z : \neg\neg A \vdash \alpha : A)}\ \text{el}\neg$$

In the right branch of this derivation, we have as conclusion context $\alpha : A, \beta : \neg A$.

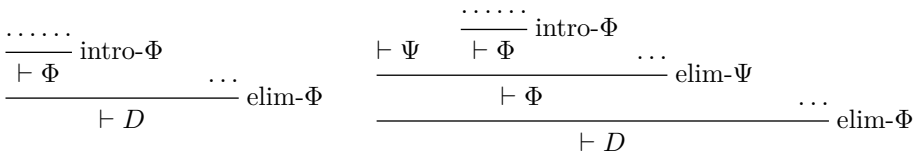

Figure 1 Detour and Permutation.

5.2 Proof terms and reductions: the intuitionistic case

We now describe the term reduction rules. For Int these should correspond to what we have defined and studied in [7] and [8]. There we have defined a proof term calculus for intuitionistic TT-ND (but without conclusion variables), and we have *detour elimination* and *permutation* rules to normalize proofs and to obtain a proof in normal form that satisfies the sub-formula property.

A *detour* in intuitionistic logic is a pattern of an introduction of a formula Φ immediately followed by an elimination of Φ. Such a step can be eliminated by not using Φ at all. This is depicted on the left in Figure 1. A *permutation* is necessary when a detour for Φ is blocked by the elimination of another formula Ψ. Then we first have to permute the two elimination rules, for Φ and Ψ, to make the detour of Φ explicit. See Figure 1 on the right.

It turns out that in our new setting, the conclusion variables nicely take care of the permutation rules. We use the following shorthand notation to improve readability:

- $\overline{p}$ represents $\overline{\mu\alpha_k : A_k.p_k}$,
- $\overline{q}$ represents $\overline{\lambda y_\ell : A_\ell.q_\ell}$.
- $\overline{r}$ represents $\overline{\mu\alpha_i : A_i.r_i}$,
- $\overline{s}$ represents $\overline{\lambda y_j : A_j.s_j}$.

▶ **Definition 22.** *The reduction rule for* Int *is as follows, for* $t \neq \beta \cdot x$.

$$(\mu\beta : \Phi.t) \cdot [\overline{p}\,;\,\overline{q}]_{r'} \quad \longrightarrow \quad t[\beta := [\overline{p}\,;\,\overline{q}]_{r'}]$$

Here the substitution $t[\beta := [\overline{p}\,;\,\overline{q}]_{r'}]$ *is defined by, inside* t,

replacing	$\beta \cdot x$	*by*	$(\mu\alpha : \Phi.\alpha \cdot x) \cdot [\overline{p}\,;\,\overline{q}]_{r'}$	*for a fresh* α
replacing	$\beta \cdot \{\overline{r}; \overline{s}\}_r$	*by*	$q_\ell[y_\ell := \mu\alpha_i : A_i.r_i]$	*if* $i = \ell$
		or by	$s_j[y_j := \mu\alpha_k : A_k.p_k]$	*if* $j = k$.

Here $q_\ell[y_\ell := \mu\alpha_i : A_i.r_i]$ *is defined by replacing* $\gamma \cdot y_\ell$ *by* $r_i[\alpha_i := \gamma]$.

Note that all free occurrences of β in t should be replaced, including those in s_j. We will see an example in Example 27. Since Int is a single-conclusion calculus, the (only) conclusion variable β of t does not occur in $\mu\alpha_i : A_i.r_i$.

▶ **Lemma 23** (Subject Reduction). *If* $t : (\Gamma \vdash \gamma : D)$ *and* $t \longrightarrow t'$, *then* $t' : (\Gamma \vdash \gamma : D)$.

Proof. By induction on the derivation of $t : (\Gamma \vdash \gamma : D)$, which treats the cases where the reduction takes place deeper inside t. The only interesting case is when t is itself a redex. For that we have to prove an auxiliary Substitution Lemma 24, which we give below and from which the case of t itself being a redex follows immediately. ◀

▶ **Lemma 24** (Substitution Property). *Let* $\Phi = c(A_1, \ldots, A_n)$ *and* $t : (\Gamma \vdash \beta : \Phi)$. *If the elim pattern* $[\overline{\mu\alpha_k : A_k.p_k}\,;\,\overline{\lambda y_\ell : A_\ell.q_\ell}]_{r'}$ *is well-typed in* $\Gamma \vdash \gamma : D$, *where* r' *is a line in the truth table for* c *that gives an elimination rule, then* $t[\beta := [\overline{p}\,;\,\overline{q}]_{r'}] : (\Gamma \vdash \gamma : D)$.

Proof. The proof is by induction on t, using the definition of the substitution given in Definition 22. ◀

The role of the side condition $t \neq \beta \cdot x$ in Definition 22 should be clear: without it we create an infinite reduction sequence right away. As a matter of fact, $(\mu\beta : \Phi.\beta \cdot x) \cdot [\overline{\mu\alpha_k : A_k.p_k}\ ;\ \overline{\lambda y_\ell : A_\ell.q_\ell}]_{r'}$ is not a redex as it is an elimination of an assumption, which is exactly what we want from a proof in normal form in Int: that all occurrences of elimination rules eliminate a hypothesis. It is convenient to syntactically characterize the normal forms of Int, $\mathsf{NF}_{\mathsf{Int}}$.

▶ **Definition 25.** *We define* $\mathsf{NF}_{\mathsf{Int}}$*, the normal forms of* Int*, as follows.*

$$\mathsf{NF}_{\mathsf{Int}} ::= \alpha \cdot x \mid (\mu\beta : \Phi.\beta \cdot x) \cdot [\overline{\mu\alpha_k : A_k.p_k}\ ;\ \overline{\lambda y_\ell : A_\ell.q_\ell}]_{r'} \mid \gamma \cdot \{\overline{\mu\alpha_k : A_k.p_k}; \overline{\lambda y_\ell : A_\ell.q_\ell}\}_{r'}$$

where all sub-terms are again in $\mathsf{NF}_{\mathsf{Int}}$*.*

We give the main properties of normal forms and reduction.

▶ **Lemma 26.**
- $\mathsf{NF}_{\mathsf{Int}}$ *captures precisely the normal forms of* Int*.*
- *The normal forms of* Int *satisfy the Sub-formula Property: If* $t : (\Gamma \vdash \gamma : D)$*,* $t \in \mathsf{NF}_{\mathsf{Int}}$*, then for each bound variable,* $\alpha : A$ *or* $x : A$*, occurring in* t*,* A *is a sub-type of a type in* Γ *or* D*.*
- *The reduction is strongly normalizing.*

Proof. The proof of the first is immediate: terms in $\mathsf{NF}_{\mathsf{Int}}$ cannot reduce and all terms that cannot reduce are in $\mathsf{NF}_{\mathsf{Int}}$. The second follows by induction on the derivation of $t : (\Gamma \vdash \gamma : D)$, where we analyze t based on the cases that arise from $t \in \mathsf{NF}_{\mathsf{Int}}$. The strong normalization property follows from the strong normalization proofs for intuitionistic TT-ND that have been given in [8] and [1]. It also follows from strong normalization of Truth Table Logic (using (2) of Lemma 41) which we prove in Theorem 46. ◀

The term reduction rules of Definition 22 indeed correspond to the ones of [7] and [8]. The interpretation is straightforward, but we do not give it here, because we would have to introduce the definitions of [7] first. To be more precise: every reduction step of the Definition corresponds to a combination of multiple detour elimination/permutation steps from[7]. So the reduction of Definition 22 covers both the detour elimination and the permutation rules. We illustrate this in an example.

▶ **Example 27.** The following deduction has a hidden detour which can be made explicit using a permutation elimination step. (The $\cdots$ are auxiliary parts of the deduction.) The $\to$ -el and $\to$ -in are separated by an $\vee$-el, so the detour arising from an $\to$ -in followed by an $\to$ -el is blocked. The detour can be made explicit using a permutation step. (Note that in TT-ND, the "normal" $\to$-introduction proceeds in two steps, first introducing the $\to$-formula $C \to D$, then abstracting over the hypothesis C.)

$$\dfrac{\dfrac{\dfrac{\cdots}{\vdash A \vee B} \quad \dfrac{\dfrac{\dfrac{\cdots}{A, C \vdash D}}{A, C \vdash C \to D} \to\text{-in}}{C \to D} \to\text{-in} \quad \dfrac{\cdots}{B \vdash C \to D}}{C \to D} \vee\text{-el} \quad \dfrac{\cdots}{\vdash C}}{D} \quad D \vdash D \to\text{-el}$$

Using the proof terms of [6], this derivation looks like this.

$$\dfrac{\dfrac{\vdash t{:}A \vee B \quad \dfrac{\dfrac{\dfrac{x{:}A, z{:}C \vdash q{:}D}{x{:}A, z{:}C \vdash \{q\ ;\ -\}{:}C \to D}\to\text{-in}}{x{:}A \vdash \{-\ ;\ \lambda z{:}C.\{q\ ;\ -\}\}{:}C \to D}\to\text{-in} \qquad y{:}B \vdash r{:}C \to D}{\vdash t \cdot [-\ ;\ \lambda x{:}A.\{-\ ;\ \lambda z{:}C.\{q\ ;\ -\}\}, \lambda y{:}B.r]{:}C \to D}\vee\text{-el} \qquad \vdash p{:}C \quad u{:}D \vdash u{:}D}{\vdash t \cdot [-\ ;\ \lambda x{:}A.\{-\ ;\ \lambda z{:}C.\{q\ ;\ -\}\}, \lambda y{:}B.r] \cdot [p\ ;\ \lambda u{:}D.u]{:}D}\to\text{-el}$$

The permutation step for this proof term

$$t \cdot [-\ ;\ \lambda x{:}A.\{-\ ;\ \lambda z{:}C.\{q\ ;\ -\}\}, \lambda y{:}B.r] \cdot [p\ ;\ \lambda u{:}D.u]$$

allows one elimination to move into another, in this case to permute the $\to$ -el into the $\vee$-el, leading to

$$t \cdot [-\ ;\ \lambda x{:}A.\{-\ ;\ \lambda z{:}C.\{q\ ;\ -\}\} \cdot [p\ ;\ \lambda u{:}D.u], \lambda y{:}B.r \cdot [p\ ;\ \lambda u{:}D.u]]$$

Here is the same proof in Int of Definition 17.

$$\dfrac{t{:}(\vdash \alpha{:}A \vee B) \quad \dfrac{\dfrac{q{:}(x{:}A, z{:}C \vdash \beta{:}D)}{\gamma \cdot \{\mu\beta{:}D.q\ ;\ -\}{:}(x{:}A, z{:}C \vdash \gamma{:}C \to D)}}{\gamma \cdot \{-\ ;\ \lambda z{:}C.\gamma \cdot \{\mu\beta{:}D.q\ ;\ -\}\}{:}(x{:}A \vdash \gamma{:}C \to D)} \quad r{:}(y{:}B \vdash \gamma{:}C \to D)}{(\mu\alpha{:}A \vee B.t) \cdot [-\ ;\ \lambda x{:}A.\gamma \cdot \{-\ ;\ \lambda z{:}C.\gamma \cdot \{\mu\beta{:}D.q\ ;\ -\}\}, \lambda y{:}B.r]{:}(\vdash \gamma{:}C \to D)}$$

Abbreviating $M := (\mu\alpha{:}A \vee B.t) \cdot [-\ ;\ \lambda x{:}A.\gamma \cdot \{-\ ;\ \lambda z{:}C.\gamma \cdot \{\mu\beta{:}D.q\ ;\ -\}\}, \lambda y{:}B.r]$ we then have

$$\dfrac{M{:}(\vdash \gamma{:}C \to D) \quad p{:}(\vdash \eta{:}C) \quad \delta \cdot u{:}(u{:}D \vdash \delta{:}D)}{(\mu\gamma{:}C \to D.M) \cdot [\mu\eta{:}C.p\ ;\ \lambda u{:}D.\delta \cdot u]{:}(\vdash \delta : D)}$$

So the final proof-term is

$$(\mu\gamma{:}C \to D.(\mu\alpha{:}A \vee B.t) \cdot [-\ ;\ \lambda x{:}A.\gamma \cdot \{-\ ;\ \lambda z{:}C.\gamma \cdot \{\mu\beta{:}D.q\ ;\ -\}\}, \lambda y{:}B.r]) \cdot [\mu\eta{:}C.p\ ;\ \lambda u{:}D.\delta \cdot u]$$

and the "hidden detour" is directly accessible via the conclusion variable γ. We have underlined the places where a substitution for γ can take place (following Definition 22). It is noteworthy that also the first $\to$ -in is contracted with the $\to$ -el. If we reduce the term according to Definition 22, we obtain, writing $N := [\mu\eta{:}C.p\ ;\ \lambda u{:}D.\delta \cdot u]$

$$(\mu\alpha{:}A \vee B.t) \cdot [-\ ;\ \lambda x{:}A.q[\beta := \delta, z := \mu\eta{:}C.p], \lambda y{:}B.r[\gamma := N]]$$

5.3 Proof terms and reductions: the classical multi-conclusion case

We now define reduction for Class-mc. We again use the following shorthand notation to improve readability:

- $\overline{p}$ represents $\overline{\mu\alpha_k : A_k.p_k}$,
- $\overline{q}$ represents $\overline{\lambda y_\ell : A_\ell.q_\ell}$.
- $\overline{r}$ represents $\overline{\mu\alpha_i : A_i.r_i}$,
- $\overline{s}$ represents $\overline{\lambda y_j : A_j.s_j}$.

▶ **Definition 28.** *The reduction rules for* Class-mc *are as follows.*

$$\begin{array}{rcll}(\lambda x:\Phi.t)\cdot\{\overline{r};\overline{s}\}_r & \longrightarrow & t & \text{if } x\notin t\\ (\mu\beta:\Phi.t)\cdot[\overline{p}\,;\,\overline{q}]_{r'} & \longrightarrow & t & \text{if } \beta\notin t\end{array}$$

$$\begin{array}{l}(\lambda x:\Phi.\ldots(\mu\beta:\Phi.\ldots\beta\cdot x\ldots)\cdot[\overline{p}\,;\,\overline{q}]_{r'}\ldots)\cdot\{\overline{r};\overline{s}\}_r \longrightarrow\\ \quad(\lambda x:\Phi.\ldots(\mu\beta:\Phi.\ldots M\ldots)\cdot[\overline{p}\,;\,\overline{q}]_{r'}\ldots)\cdot\{\overline{r};\overline{s}\}_r\end{array}$$

$$\begin{array}{l}(\mu\beta:\Phi.\ldots(\lambda x:\Phi.\ldots\beta\cdot x\ldots)\cdot\{\overline{r};\overline{s}\}_r\ldots)\cdot[\overline{p}\,;\,\overline{q}]_{r'} \longrightarrow\\ \quad(\mu\beta:\Phi.\ldots(\lambda x:\Phi.\ldots M\ldots)\cdot\{\overline{r};\overline{s}\}_r\ldots)\cdot[\overline{p}\,;\,\overline{q}]_{r'}\end{array}$$

where for M we can choose

$$\begin{array}{rcl}M & = & q_\ell[y_\ell:=\mu\alpha_i:A_i.r_i] \text{ if } i=\ell\\ & = & s_j[y_j:=\mu\alpha_k:A_k.p_k] \text{ if } j=k.\end{array}$$

Here $q_\ell[y_\ell:=\mu\alpha_i:A_i.r_i]$ is defined by replacing $\gamma\cdot y_\ell$ by $r_i[\alpha_i:=\gamma]$.

Similar to the intuitionistic case (Lemma 23), we have Subject Reduction, which is (again) based on the type soundness of the substitution that is involved.

▶ **Lemma 29** (Subject Reduction)**.** *If $t:(\Gamma\vdash\Delta)$ and $t\longrightarrow t'$, then $t':(\Gamma\vdash\Delta)$.*

Proof. By induction on the derivation of $t:(\Gamma\vdash\Delta)$, which treats the cases where the reduction takes place deeper inside t. The only interesting case is when t is itself a redex. For that we have to verify that the substitution, of M for $\beta\cdot x$ (and the substitutions that are part of the definition of M) are type correct. That is the case, as we always apply goal variables to assumption variables of the right type, and we substitute expressions under a binder, which avoids the risk of variables being no longer in scope. (And the usual variable hygiene, with renaming of bound variables, avoids capture of free variables by a binder.) ◀

Just as for the single conclusion intuitionistic case, we give an explicit syntax for the normal forms, and give the basic properties for reduction and normal forms, as in Definitions 25 and Lemma 26 for the intuitionistic case.

▶ **Definition 30.** *We define* $\mathsf{NF}_{\text{Class-mc}}$*, the normal forms of* Class-mc*, as follows.*

$$\mathsf{NF}_{\text{Class-mc}} ::= \alpha\cdot x \mid (\mu\beta:\Phi.t)\cdot[\overline{p}\,;\,\overline{q}]_{r'} \mid (\lambda x:\Phi.q)\cdot\{\overline{p};\overline{q}\}_{r'}$$

where all sub-terms are again in $\mathsf{NF}_{\text{Class-mc}}$ *and*

- *x occurs in q only as $\alpha\cdot x$ with α free in q, and x occurs at least once.*
- *β occurs in t only as $\beta\cdot z$ with z free in t, and β occurs at least once.*

We give the main properties of normal forms and reduction.

▶ **Lemma 31.**
- $\mathsf{NF}_{\text{Class-mc}}$ *captures precisely the normal forms of* Class-mc*.*
- *The normal forms of* Class-mc *satisfy the Sub-formula Property: If $t:(\Gamma\vdash\Delta)$, $t\in\mathsf{NF}_{\text{Class-mc}}$, then for each bound variable, $\alpha:A$ or $x:A$, occurring in t, A is a sub-type of a type in Γ or Δ.*
- *The calculus is normalizing, in the sense that, if $t:(\Gamma\vdash\Delta)$, then there is a term in normal form q, such that $q:(\Gamma\vdash\Delta)$.*

Proof. The proof of the first is immediate: terms in $\mathsf{NF}_{\mathsf{Class\text{-}mc}}$ cannot reduce and all terms that cannot reduce are in $\mathsf{NF}_{\mathsf{Class\text{-}mc}}$. The second follows by induction on the derivation of $t : (\Gamma \vdash \Delta)$, where we analyze t based on the cases that arise from $t \in \mathsf{NF}_{\mathsf{Class\text{-}mc}}$. The normalization property follows from the strong normalization proof for Truth Table Logic (Theorem 46), which is a generalization of all these systems, and the fact that normal forms in Truth Table Logic can be reflected back to normal forms in Class-mc (Lemma 41 part (3)). ◀

5.4 Proof terms and reductions: the classical single-conclusion case

The reductions for proof-terms in the classical single-conclusion case are a bit less well-behaved than the ones for the multi-conclusion case, because in the classical single conclusion case, there should always be at most one free conclusion variable in a proof term. This means that if t is a proof-term in classical single conclusion logic, then

- a sub-expression $\lambda y : A.q$ of t should have at most one free conclusion variable,
- in a sub-expression $\mu\alpha : A.q$ of t, q should have only α as free conclusion variable.

If one starts from a proof-term t having these properties, and one performs the reductions of Definition 28, one easily ends up in a proof term t' that violates them. Then t' is still valid in multi-conclusion logic, but not in the single conclusion system. The way to prevent this is to first reduce a term to a *lemma-normal form*, which is a term where all Lemmas are assumptions (variables). Those can then be reduced safely in the "standard" way of Definition 28.

This means that reduction for proof terms in single conclusion classical logic proceeds as follows.

- First perform *permutation reductions* to obtain a *lemma-normal form*.
- Then perform *detour reductions*, where a *detour* is an elimination of Φ followed by an introduction of Φ.

This is similar to the constructive case, except for now a term is in "permutation normal form" if all lemmas are axioms.

▶ **Definition 32.** *This is the abstract syntax* $\mathsf{NF}_{\mathsf{lemma}}$ *for* lemma-normal forms*:*

$$\alpha \cdot x \mid (\lambda x{:}A.t) \cdot \{\overline{\mu\alpha_k{:}A_k.\alpha_k \cdot z_k}; \overline{\lambda y_\ell{:}A_\ell.q_\ell}\} \mid (\mu\beta{:}A.\beta \cdot y) \cdot [\overline{\mu\alpha_k{:}A_k.\alpha_k \cdot z_k}\ ;\ \overline{\lambda y_\ell{:}A_\ell.q_\ell}],$$

where x, y, z *range over variables and* t *and the* q *are again in* $\mathsf{NF}_{\mathsf{lemma}}$.

We can obtain a proof term in lemma-normal form by moving applications of an elimination or introduction rule that have a non-trivial Lemma upwards, until all Lemmas become trivial: the proof terms are variables. (Note that $\mu\beta{:}A.\beta \cdot y$ is basically the assumption $y : A$.) This only works in classical logic. If one tries this for intuitionistic proofs, which –from the point of view of classical logic– are proofs with a trivial main premise in the classical introduction rule, one immediately ends up with a proof that has a non-trivial main premise in the classical introduction rule.

We can now specialize Definition 28 to the single-conclusion case by considering only terms in lemma-normal form. We use a similar abbreviation style as before:

- $\overline{z}$ represents $\overline{\mu\alpha_k : A_k.\alpha_k \cdot z_k}$,
- $\overline{q}$ represents $\overline{\lambda y_\ell : A_\ell.q_\ell}$.
- $\overline{v}$ represents $\overline{\mu\alpha_i : A_i.\alpha_i \cdot v_i}$,
- $\overline{s}$ represents $\overline{\lambda y_j : A_j.s_j}$.

▶ **Definition 33.** *In* Class*, the notion of detour and the reduction rules are modified as follows.*

1. *A* detour *is a pattern of the following shape.*

$$(\lambda x : \Phi. \ldots (\mu\beta : \Phi.\beta \cdot x) \cdot [\overline{z} \,;\, \overline{q}] \ldots) \cdot \{\overline{v}; \overline{s}\}$$

2. *The reduction rules are*

$$(\lambda x : \Phi.t) \cdot \{\overline{r}; \overline{s}\}_r \quad \longrightarrow \quad t \text{ if } x \notin t$$
$$(\lambda x : \Phi. \ldots (\mu\beta : \Phi.\beta \cdot x) \cdot [\overline{z} \,;\, \overline{q}] \ldots) \cdot \{\overline{v}; \overline{s}\} \quad \longrightarrow \quad (\lambda x : \Phi. \ldots M \ldots) \cdot \{\overline{v}; \overline{s}\}$$

where for M we can choose

$$\begin{aligned} M &= q_\ell[y_\ell := v_i] \text{ if } i = \ell \\ &= s_j[y_j := z_k] \text{ if } j = k. \end{aligned}$$

Again we have Subject Reduction, which basically follows from the classical multi-conclusion case (Lemma 29). The only thing to verify is that we don't go "out of" the single conclusion fragment.

▶ **Lemma 34** (Subject Reduction)**.** *If $t : (\Gamma \vdash \gamma : D)$ in* Class*, $t \in \mathsf{NF}_{\mathsf{lemma}}$ and $t \longrightarrow t'$, then $t' : (\Gamma \vdash \gamma : D)$ in* Class *and $t' \in \mathsf{NF}_{\mathsf{lemma}}$.*

Proof. We don't have to verify the type of t, as its correctness follows from Lemma 29. To verify that t' is a single-conclusion proof term and $t' \in \mathsf{NF}_{\mathsf{lemma}}$ follows from the fact that we never substitute a term under a μ-binder and for assumption variables, we just substitute other assumption variables. ◀

Again, we give an explicit syntax for the normal forms, and the basic properties for reduction and normal forms, as in Definitions 30 and Lemma 31 for the multi-conclusion classical case.

▶ **Definition 35.** *We define $\mathsf{NF}_{\mathsf{Class\text{-}sc}}$, the normal forms of* Class*, as follows.*

$$\alpha \cdot x \mid (\lambda x{:}A.t) \cdot \{\overline{\mu\alpha_k{:}A_k.\alpha_k \cdot z_k}; \overline{\lambda y_\ell{:}A_\ell.q_\ell}\} \mid (\mu\beta{:}A.\beta \cdot y) \cdot [\overline{\mu\alpha_k{:}A_k.\alpha_k \cdot z_k} \,;\, \overline{\lambda y_\ell{:}A_\ell.q_\ell}],$$

where t and the q_ℓ are again in $\mathsf{NF}_{\mathsf{Class\text{-}sc}}$ and

- *x occurs in t only as $\alpha \cdot x$ with α free in t, and x occurs at least once.*

We give the main properties of normal forms and reduction.

▶ **Lemma 36.**

- *$\mathsf{NF}_{\mathsf{Class\text{-}sc}}$ captures precisely the normal forms of* Class-sc*.*
- *The normal forms of* Class-sc *satisfy the Sub-formula Property: If $t : (\Gamma \vdash \gamma : D)$, $t \in \mathsf{NF}_{\mathsf{Class\text{-}sc}}$, then for each bound variable, $\alpha : A$ or $x : A$, occurring in t, A is a sub-type of a type in Γ or D.*
- *The calculus is normalizing, in the sense that, if $t : (\Gamma \vdash D)$, then there is a term in normal form q, such that $q : (\Gamma \vdash D)$.*

Proof. The proof is the same as for Lemma 31. For the second second we analyze t based on the cases that arise from $t \in \mathsf{NF}_{\mathsf{Class\text{-}sc}}$. The normalization property follows from the normalization of Class-mc and the fact that, from a normal form in Class-mc we can construct a normal form in Class-sc by taking the lemma-normal form. ◀

6 Truth Table Logic

We now define Truth Table Logic as a unifying logic and proof term calculus for the various logics we have seen before. The idea of Truth Table Logic is that, for $\Phi = c(A_1, \ldots, A_n)$, we decompose an elimination rule for Φ in two parts:

1. the part to be eliminated, $\mu\beta : \Phi.t$, which is the proof of Φ,
2. the *elim pattern*, $[\overline{\mu\alpha_k : A_k.t_k}\ ;\ \overline{\lambda y_\ell : A_\ell.t_\ell}]_{r'}$, consisting of the Lemmas and Casuses that we eliminate Φ with, which are the proofs of A_k (in case A_k is a Lemma) or the proofs from assumption A_ℓ (in case A_ℓ is a Casus).

These elim patterns have already been introduced in Definition 18, but in Truth Table Logic they will get a "first class status", as expressions that have their own type. Similarly we decompose a classical introduction rule in two parts (and give intro patterns a first class status):

1. the part to be introduced, $\lambda y : \Phi.t$, which is the proof from assumption Φ,
2. the *intro pattern*, $\{\overline{\mu\alpha_i : A_i.t_i}; \overline{\lambda y_j : A_j.t_j}\}_r$, consisting of the Lemmas and Casuses that we intro Φ with, which are the proofs of A_i (in case A_i is a Lemma) or the proofs from assumption A_j (in case A_j is a Casus).

For conciseness, we again adopt the earlier abbreviations, where we let $\overline{p}$ and $\overline{r}$ represent series of μ-abstractions, while $\overline{q}$ and $\overline{s}$ represent series of λ-abstractions:

- $\overline{a}$ and $\overline{b}$ typically represent $\overline{\mu\alpha_k : A_k.t_k}$,
- $\overline{f}$ and $\overline{g}$ typicallty represent $\overline{\lambda y_\ell : A_\ell.t_\ell}$.

We will be able to combine intro patterns and elim patterns directly into a proof, without explicit "interference" of an elimination or introduction rule, so we will e.g. have the following as a proof term:

$$\{\overline{b}; \overline{g}\} \cdot [\overline{a}\ ;\ \overline{f}].$$

To make this work, we introduce a new type o, that can informally be read as the type of proofs. It is the only type in Truth Table Logic that is *not* related to a specific formula. In Truth Table Logic, each proof term t will be of the form $f \cdot a$, where the parts f and a have one of the following syntactical forms:

- a variable x or α
- an abstraction $\lambda x.t$ or $\mu\alpha.t$
- an elim pattern $[\overline{a}\ ;\ \overline{f}]$ or an intro pattern $\{\overline{b}; \overline{g}\}$.

These parts will be treated as typed terms on their own and proof terms are just applications $f \cdot a$ that result in type o. Which applications are allowed depends on the variant of Truth Table Logic: we have three variants, where one can choose for intuitionistic or classical logic and for classical logic the single-conclusion or multiple-conclusion variant.

When we consider the "application" $f \cdot a$, we treat f as function and a as argument, not the other way around. If a is of type T, then f should be of type $T \to o$. We abbreviate the function type $T \to o$ to $\sim T$. The type T can be a formula Φ, but it can also be $\sim\Phi$, and then we would have $a : \sim\Phi$ and $f : \sim\sim\Phi$ to make $f \cdot a : o$. So every type T is related to some formula: it can be Φ itself, or $\sim\Phi$, or $\sim\sim\Phi$ or $\sim\sim\sim\Phi$.

The typing of the μ-abstractions and λ-abstractions is determined by the typing of the variables: if $t : o$ and v is an assumption or conclusion variable of type T, then $\mu v{:}T.t : \sim T$ and $\lambda v{:}T.t : \sim T$.

There are three variants of Truth Table Logic, but we first give the generic definition.

▶ **Definition 37** (Truth Table Logic). *Given a set of connectives $\mathcal{C}$ with their truth tables and the propositional formulas generated from $\mathcal{C}$, we define the following classes of types and pre-terms:*

$$\begin{array}{rcl} T & ::= & \Phi \mid {\sim}T \\ t & ::= & f \cdot a \mid a \cdot f \\ f & ::= & \alpha \mid \lambda x : T.t \mid [\overline{a}\,;\,\overline{f}]_r \\ a & ::= & x \mid \mu\alpha : T.t \mid \{\overline{a};\overline{f}\}_r \end{array}$$

where Φ ranges over formulas, α ranges over conclusion variables, x ranges over assumption variables, and r ranges over the rules of the connectives.

Contexts are of the form $\Gamma;\Delta$, where Γ consists of declarations of assumption variables, typically $\Gamma = x_1 : T_1, \ldots, x_n : T_n$, and Δ consists of declarations of conclusion variables, typically $\Delta = \alpha_1 : T_1, \ldots, \alpha_m : T_m$.

A typing judgment is of the respective forms $\Gamma;\Delta \vdash t : o$, $\Gamma;\Delta \vdash f : T$, $\Gamma;\Delta \vdash a : T$, for some type T. We have the derivation rules below for deriving a typing judgment.

The allowed form of the type T in each of the rules depends on the variant of Truth Table Logic, and will be specified in Definition 38. In the rules (in-pat) and (el-pat), each T_i and T_k is the type of μ-abstractions *and each T_j and T_ℓ is the* type of λ-abstractions *in the specific variant of Truth Table Logic.*

$$\frac{}{\Gamma;\Delta \vdash x : T}\,(hyp),\ \text{if } x : T \in \Gamma \qquad \frac{}{\Gamma;\Delta \vdash \alpha : T}\,(conc),\ \text{if } \alpha : T \in \Delta$$

$$\frac{\Gamma, x : T;\Delta \vdash t : o}{\Gamma;\Delta \vdash \lambda x : T.t : {\sim}T}\,(\lambda) \qquad \frac{\Gamma;\Delta, \alpha : T \vdash t : o}{\Gamma;\Delta \vdash \mu\alpha : T.t : {\sim}T}\,(\mu)$$

$$\frac{\Gamma;\Delta \vdash f : {\sim}T \qquad \Gamma;\Delta \vdash a : T}{\Gamma;\Delta \vdash f \cdot a : o}\,(app_1) \qquad \frac{\Gamma;\Delta \vdash a : {\sim}T \qquad \Gamma;\Delta \vdash f : T}{\Gamma;\Delta \vdash a \cdot f : o}\,(app_2)$$

$$\frac{\ldots\Gamma;\Delta \vdash a_i : T_i \ldots \quad \ldots\Gamma;\Delta \vdash f_j : T_j \ldots}{\Gamma;\Delta \vdash \{\overline{a_i};\overline{f_j}\}_r : T}\,(in\text{-}pat)$$

$$\frac{\ldots\Gamma;\Delta \vdash a_k : T_k \ldots \quad \ldots\Gamma;\Delta \vdash f_\ell : T_\ell \ldots}{\Gamma;\Delta \vdash [\overline{a_k}\,;\,\overline{f_\ell}]_r : T}\,(el\text{-}pat)$$

Here, $\Phi = c(A_1, \ldots, A_n)$ and if r is a 1-row for connective c we have the rule (in-pat), where the T_i are related to the A_i for which $r_i = 1$ and the T_j are related to the A_j for which $r_j = 0$. If r is a 0-row for c we have the rule (el-pat), where the T_k are related to the A_k for which $r_k = 1$ and the T_ℓ are related to the A_ℓ for which $r_\ell = 0$.

We now define how the various logics arise from the definition by specifying what is allowed for T in the various derivation rules. In each variant of Truth Table Logic, conclusion variables α for Φ and elim patterns $[\overline{a}\,;\,\overline{f}]_r$ for Φ are of type ${\sim}\Phi$. This implies that μ-abstractions for Φ, $\mu\alpha : {\sim}\Phi.t$, are of type ${\sim}{\sim}\Phi$ and substituting an elim pattern for a conclusion variable in a proof term is well-typed.

▶ **Definition 38.** *Classical multi-conclusion calculus* Class-mc *is specified by*

$\sim\sim\Phi$	$\sim\Phi$	Φ
$\mu\alpha.t$	$\lambda x.t$	
	α	x
	$[\overline{f}\ ;\ \overline{g}]_r$	$\{\overline{f};\overline{g}\}_r$

Intuitionistic single-conclusion calculus Int *is specified by*

$\sim\sim\sim\Phi$	$\sim\sim\Phi$	$\sim\Phi$	Φ
$\lambda x.t$	$\mu\alpha.t$		
	x	α	
		$[\overline{f}\ ;\ \overline{g}]_r$	$\{\overline{f};\overline{g}\}_r$

Classical single-conclusion calculus, Class-sc *is specified by*

$\sim\sim\Phi$	$\sim\Phi$	Φ
$\mu\alpha.t$	$\lambda x.t$	
	α	x
$\{\overline{f};\overline{g}\}_r$	$[\overline{f}\ ;\ \overline{g}]_r$	

In the Class-mc variant of Truth Table Logic, all proof-terms of the forms $(\mu\alpha.t)\cdot[\overline{f}\ ;\ \overline{g}]_r$ and $(\lambda x.t)\cdot\{\overline{f};\overline{g}\}_r$ are permitted. Note that we have reversed some of the applications, and some proof terms can be reduced further than in the original Class-mc.

▶ **Example 39.** We revisit Example 21, where we saw a classical multi-conclusion proof term of type $z:\neg\neg A\vdash\alpha:A$. We can recast that term in Class-mc in Truth Table Logic, and then we have the following term t of type o in the context $z:\neg\neg A;\alpha:\sim A$.

$$t \quad := \quad (\mu\gamma:\sim\neg\neg A.\gamma\cdot z)\cdot[\mu\beta:\sim\neg A.(\lambda y:\neg A.\beta\cdot y)\cdot\{;\lambda x:A.\alpha\cdot x\}_r\ ;\ ]_{r'}$$

But there is a simpler term, because we can normalize t further. (See below for the reduction rules.) Then we get $z:\neg\neg A;\alpha:\sim A\vdash t':o$ with

$$t' \quad := \quad [\mu\beta:\sim\neg A.\beta\cdot\{;\lambda x:A.\alpha\cdot x\}_r\ ;\ ]_{r'}\cdot z.$$

The derivation in Truth Table Logicis as follows, where we omit $z:\neg\neg A$ and $\alpha:\sim A$ from the context, except for the conclusion.

$$\dfrac{\dfrac{\beta:\sim\neg A\vdash\beta:\sim\neg A \qquad \dfrac{\dfrac{x:A.\vdash\alpha\cdot x:o}{\vdash\lambda x:A.\alpha\cdot x:\sim A}}{\vdash\{;\lambda x:A.\alpha\cdot x\}_r:\neg A}}{\beta:\sim\neg A\vdash\beta\cdot\{;\lambda x:A.\alpha\cdot x\}_r:o}}{\dfrac{\vdash[\mu\beta:\sim\neg A.\beta\cdot\{;\lambda x:A.\alpha\cdot x\}_r\ ;\ ]_{r'}:\sim\neg\neg A \qquad \vdash z:\neg\neg A}{z:\neg\neg A;\alpha:\sim A\vdash[\mu\beta:\sim\neg A.\beta\cdot\{;\lambda x:A.\alpha\cdot x\}_r\ ;\ ]_{r'}\cdot z:o}}$$

In Int, we have reversed the application $\alpha\cdot x$, but that is merely a syntactic reformulation. The main point of Int is that we do not have $(\lambda x.t)\cdot\{\overline{f};\overline{g}\}_r$, to avoid the classical introduction rule. In Int we only have $\gamma\cdot\{\overline{f};\overline{g}\}_r$, which is exactly the term we had for the intuitionistic introduction rule in Definition17.

The system Class-sc is a subsystem of Class-mc, but we avoid the redex $(\lambda x.t)\cdot\{\overline{f};\overline{g}\}_r$, by reversing the order of application to $\{\overline{f};\overline{g}\}_r\cdot(\lambda x.t)$. See below for the reduction rules, where we will enforce that only an abstraction on the left of an application gives a redex. This avoids the possibility of having multiple free conclusion variables in a proof term.

▶ **Definition 40** (Reduction of proof terms in Truth Table Logic). *In the (intuitionistic) variant in which patterns are applied in the order* $[\ldots] \cdot \{\ldots\}$ *and abstractions in the order* $(\lambda x.s) \cdot (\mu\alpha.t)$*, proof terms are reduced to proof terms as follows:*

$$\begin{array}{rcl}
(\lambda x : T.t) \cdot a & \longrightarrow & t[x := a] \\
(\mu\alpha : T.t) \cdot f & \longrightarrow & t[\alpha := f] \\
[\overline{b_k}\ ;\ \overline{g_\ell}]_r \cdot \{\overline{a_i}; \overline{f_j}\}_{r'} & \longrightarrow & g_\ell \cdot a_i \\
 & & \textit{if } i = \ell \textit{ as indexes in } 1, \ldots, n, \textit{ where } \Phi = c(A_1, \ldots, A_n) \\
 & & \textit{and } r, r' \textit{ are rules for } \Phi \\
[\overline{b_k}\ ;\ \overline{g_\ell}]_r \cdot \{\overline{a_i}; \overline{f_j}\}_{r'} & \longrightarrow & f_j \cdot b_k \\
 & & \textit{if } j = k \textit{ as indexes in } 1, \ldots, n, \textit{ where } \Phi = c(A_1, \ldots, A_n) \\
 & & \textit{and } r, r' \textit{ are rules for } \Phi
\end{array}$$

In the classical variants, the order of the patterns in the redex and/or the order of the parts in the reduct is reversed.

That Truth Table Logic is a unification of the logics (actually the calculi) that we have seen in the previous section can be stated and proven precisely.

▶ **Lemma 41.** *For each of the calculi of the previous section,* Int, Class *and* Class-mc*, the obvious interpretation* $[\![-]\!]$ *of proof terms of the logic as proof terms of Truth Table Logic has the following properties.*

1. *If* $t : (\Gamma \vdash \Delta)$ *in* Int, Class *or* Class-mc*, then* $\Gamma; \Delta \vdash [\![t]\!] : o$ *in Truth Table Logic.*
2. *If* $t \longrightarrow q$ *in* Int*, then* $[\![t]\!] \longrightarrow^+ [\![q]\!]$*, where* $\longrightarrow^+$ *is the transitive closure of* $\longrightarrow$*.*
3. *If* $\Gamma; \Delta \vdash q : o$ *in the classical multi-conclusion variant of Truth Table Logic, and* q *is in normal form, we can reconstruct a proof term* t *in* Class-mc *such that* $[\![t]\!] = q$ *and* $t : (\Gamma \vdash \Delta)$ *in* Class-mc

Proof. The interpretation is the obvious one, and the proof is a direct check of all the cases. ◀

A proof term $f \cdot a$ is a redex if f is an abstraction or both f and a are patterns. In most variants, for each proof term $f \cdot a$ in normal form, at least one of the parts is a variable. In such a calculus, each normal proof has the sub-formula property: except for the assumptions, conclusions and sub-formulas, no other formulas are used.

The normal forms of type o of the intuitionistic variant of our term calculus are of the following shape:

$$x \cdot \alpha \mid \alpha \cdot \{\overline{f}; \overline{g}\}_r \mid x \cdot [\overline{f}\ ;\ \overline{g}]_r$$

In the other variants, the order of the parts of some of these normal forms are reversed. There are no variants with proof terms $f \cdot a$ in which f is a variable and a an abstraction. But the single-conclusion classical variant has proof terms of the form $\{\ldots\} \cdot (\lambda x.t)$. In this calculus, more proof term reductions are needed to get the sub-formula property.

We now prove Strong normalization of reduction for each variant of Truth Table Logic, showing that every reduction sequence leads to a proof term in normal form. We will define, with induction on the type T of terms t the property "t is hereditarily strong normalizing". Then we prove, by induction on the structure of a term t, that each substitution of hereditarily strong normalizing terms for the free variables of t results in a (hereditarily) strongly normalizing term.

For the rest of this section, the variant of Truth Table Logic is fixed. So the types of the conclusion and assumption variables, λ- and μ-abstractions, elim and intro patterns for a formula Φ are specific choices from $\sim\sim\sim\Phi$, $\sim\sim\Phi$, $\sim\Phi$, and Φ.

To prove strong normalization we need the notion of "hereditarily strongly normalizing".

▶ **Definition 42.** *We say that a term t is* HSN (hereditarily strongly normalizing) *if*

1. *t is* SN*,*
2. *if t is an abstraction $\lambda x : T.e$ or $\mu\alpha : T.e$, then $t \cdot a$ is* SN *for each* HSN *$a : T$,*
3. *if t is a pattern $\{\overline{a}; \overline{f}\}_r$ or $[\overline{a}\ ;\ \overline{f}]_r$, then all $\overline{a}, \overline{f}$ are* HSN*.*

Note that this definition is by induction on the type of term t.

Also note that if t is HSN and $t \longrightarrow t'$, then t' is HSN. This follows from the fact that each redex and each reduct is a proof term:

- if t' is an abstraction $\lambda x.e'$ or $\mu\alpha.e'$, then t must be of the form $\lambda x.e$ or $\mu\alpha.e$ where $e \longrightarrow e'$,
- if t' is a pattern $\{\overline{a'}; \overline{f'}\}_r$ or $[\overline{a'}\ ;\ \overline{f'}]_r$, then t must be a pattern $\{\overline{a}; \overline{f}\}_r$ or $[\overline{a}\ ;\ \overline{f}]_r$ that is the same as t' except that for a single i or j, $a_i \longrightarrow a'_i$ or $f_j \longrightarrow f'_j$.

▶ **Definition 43.** *We consider* substitutions *σ that assign terms to variables in a well-typed way:*

- *for each assumption variable x of type T, $\sigma(x) : T$,*
- *for each conclusion variable α of type T, $\sigma(\alpha) : T$.*

Substitution for variables extends in a straightforward way to all terms, so we write $\sigma(t)$ for the result of substituting $\sigma(v)$ for each free occurrence of v in a term t (after having renamed bound variables in t if needed to avoid capturing of free variables of $\sigma(v)$). Note that t and $\sigma(t)$ are terms of the same type.

A substitution σ is HSN *if term $\sigma(v)$ is* HSN *for each variable v.*

A term t is strongly HSN *if term $\sigma(t)$ is* HSN *for each* HSN *substitution σ.*

Since each variable x or α is HSN, the identity substitution is HSN and so each strongly HSN term is HSN.

As usual, reduction rules are closed under substitution: if $t \longrightarrow t'$ then $\sigma(t) \longrightarrow \sigma(t')$. By definition, $t \longrightarrow t'$ if and only if t has a subterm s (possibly $s = t$ itself) that is a redex with reduct s' and t' is the result of replacing s in t by s'. Since, as usual, each free variable of s' is a free variable of s, there is no danger of unintended capturing of a free variable of s' by a surrounding abstraction inside t. Let σ' be the substitution that is like σ, except that $\sigma'(v) = v$ for each variable v that is bound by an abstraction surrounding the subterm s of t. Then for the corresponding reduction $\sigma(t) \longrightarrow \sigma(t')$, the subterm $\sigma'(s)$ of $\sigma(t)$ is replaced by $\sigma'(s)$.

The usually β-reduction $(\lambda x.t) \cdot a \longrightarrow t[x := a]$ has the special property that for each substitution σ, $\sigma((\lambda x.t) \cdot a) \longrightarrow \sigma'(t)$, where the substitution σ' is like σ except that $\sigma'(x) = \sigma(a)$.

▶ **Lemma 44.** *If proof term e is strongly* HSN*, then the abstractions $\lambda x.e$ and $\mu\alpha.e$ are strongly* HSN*.*

Proof. Let e be a strongly HSN proof term and σ an HSN substitution. We do the case for $\lambda x : T.e$. We have to show that $\sigma(\lambda x.e)$ is HSN. Note that $\sigma(\lambda x.e) = \lambda x.\sigma'(e)$ where substitution σ' is like σ, except that $\sigma'(x) = x$. For (1), we need that $\lambda x.\sigma'(e)$ is SN, which follows from the fact that substitution σ' is HSN and term e is strongly HSN. For (2), let $a : T$ be HSN. We have to show that $(\lambda x.\sigma'(e)) \cdot a$ is SN. Since $\lambda x.\sigma'(e)$ and a are SN, an infinite path from $(\lambda x.\sigma'(e)) \cdot a$ must contract a redex $(\lambda x.e') \cdot a'$ where $\sigma'(e) \longrightarrow^* e'$ and $a \longrightarrow^* a'$.

In Truth Table Logic, this reduct is $e'[x := a']$. Note that $\sigma'(e)[x := a'] \longrightarrow^* e'[x := a']$. Let σ'' be the substitution that is like σ and σ' except that $\sigma''(x) = a'$. Substitution σ'' is HSN since term $a' : T$ is HSN. So proof term $\sigma''(e)$ is SN. Now $\sigma''(e) = \sigma'(e)[x := a'] \longrightarrow^* e'[x := a']$, so the reduct $e'[x := a']$ is SN and cannot create an infinite path either. ◀

Note that the situation is different if we add reduction rules like those for Class-mc in which redex $(\lambda x.e') \cdot a'$ can have reducts of the form $(\lambda x.e'') \cdot a'$ where $e' \not\longrightarrow^* e''$ but $e'[x := a'] \longrightarrow^+ e''[x := a']$. Then $\sigma''(e) \longrightarrow^+ e''[x := a']$, so $e''[x := a']$ is SN. This implies that e'' is SN, so an infinite path from $(\lambda x.e'') \cdot a$ must contract a redex again. This cannot happen infinitely often: an infinite path $(\lambda x.e') \cdot a \longrightarrow (\lambda x.e'') \cdot a \longrightarrow (\lambda x.e''') \cdot a \longrightarrow \ldots$ would result in an infinite path $e'[x := a'] \longrightarrow^+ e''[x := a'] \longrightarrow^+ e'''[x := a'] \longrightarrow \ldots$.

▶ **Lemma 45.** *Each term t is strongly* HSN.

Proof. We prove this by induction on t.

- If t is a variable α or x, then for each HSN substitution σ, $\sigma(t)$ is HSN by definition.
- If t is an abstraction $\lambda x.e$ or $\mu\alpha.e$, then the proof term e is strongly HSN by induction, so t is strongly HSN by Lemma 44.
- If t is a pattern $\{\overline{a}; \overline{f}\}_r$ or $[\overline{a}\ ;\ \overline{f}]_r$, then all terms $\overline{a}, \overline{f}$ are strongly HSN by induction. Let σ be an HSN substitution. We have to show that $\sigma(t)$ is an HSN term. For (1), we need that $\sigma(t)$ is SN, which follows from the fact that all terms $\sigma(a_i)$ and $\sigma(f_j)$ are SN. For (3), we need that all all terms $\sigma(a_i)$ and $\sigma(f_j)$ are HSN, which holds by induction.
- If t is a proof term $f \cdot a$, then the terms f and a are strongly HSN by induction. Let σ be an HSN substitution. We have to show that $\sigma(t)$ is an HSN term. Since $\sigma(t) = \sigma(f) \cdot \sigma(a)$, we only need to show (1): proof term $\sigma(f) \cdot \sigma(a)$ is SN. Both $\sigma(f)$ and $\sigma(a)$ are HSN and thus SN. An infinite path from $\sigma(f) \cdot \sigma(a)$ must contract a redex $f' \cdot a'$ where $\sigma(f) \longrightarrow^* f'$ and $\sigma(a) \longrightarrow^* a'$. Now both f' and a' are HSN terms. Since $f' \cdot a'$ is a redex in Truth Table Logic, either f' is an abstraction or both f' and a' are patterns. If f' is an HSN abstraction, then by (1), since a' is HSN, $f' \cdot a'$ is SN. If both f' and a' are HSN patterns for some formula $\Phi = c(A_1, \ldots, A_n)$ and the reduct is $f'' \cdot a''$, then f'' and a'' are HSN terms for some subformulas A_i, so f'' and a'' are SN. So an infinite path from $f'' \cdot a''$ must contract a redex $f''' \cdot a'''$ again, where $f'' \longrightarrow^* f'''$ and $a'' \longrightarrow^* a'''$. Since f''' has the same type as f'' (and a''' has the same type as a''), related to formula A_i, this cannot happen infinitely often. ◀

▶ **Theorem 46.** *All proof terms in Truth Table Logic are* SN.

Proof. By Lemma 45, each (proof) term is strongly HSN, so HSN, so SN. ◀

7 Conclusion

We have shown a couple of basic results for general classical logic derived from truth tables. Most surprisingly maybe is that one classical connective makes the whole logic classical: it is not possible to combine e.g. a classical implication with a constructive negation, as the negation becomes classical due to the fact that implication is classical. Truth Table Natural Deduction, TT-ND, provides the good setting for studying these properties as it gives generic deduction rules for connectives "in isolation", i.e. without explaining one connective in terms of the other.

Then we have studied the proof theory of classical TT-ND, and we have introduced proof terms for classical deductions that use both assumption variables (hypotheses) and conclusion variables. This has enabled us to study proof normalization, with the aim that

proofs in normal form satisfy the sub-formula property. The use of conclusion variables turns out to be useful in general, also for the intuitionistic case, where it enables a reduction rule that unifies detour steps and permutation steps. Conclusion variables also naturally enable multi-conclusion natural deduction. Classical multi-conclusion TT-ND is the most general of these systems, where other systems can be embedded into. For this system we prove strong normalization. Classical multi-conclusion TT-ND also emphasizes that there are basically four term-formers: λ-abstraction, μ-abstraction, intro patterns and elim patterns, where the latter two are derived from the truth table. Based on this we define Truth Table Logic as a unifying system.

8 Future and Related Research

For future work, we see the further study of Truth Table Logic as unifying framework for TT-ND. Truth Table Logic also emphasizes the interpretation of proofs of a negated formula as a continuation, where $\sim\Phi$ denotes the type of continuations over Φ. This relates to the general question of the computational interpretation of classical proofs, which has been studied in various research works, like [13, 2, 4], and it is to be studied how our generic computation rules relate to the concrete ones studied in these papers for implication.

Also, the system TT-ND derives rules from a truth table, but that doesn't cover all possible connectives. E.g. if one defines a constructive connective in terms of other, the truth table one obtains generates constructive rules that are sometimes stronger and sometimes weaker than the constructive formula. For example, if we consider the truth table for $c(A, B) := \neg A \to B$, the constructive rules we derive for c are exactly the ones for $A \vee B$, which is stronger than $\neg A \to B$. On the other hand, if we consider the truth table for $d(A, B) := \neg A \vee B$, the constructive rules we derive for c are exactly the ones for $A \to B$, which is weaker than $\neg A \vee B$.

Olkhovikov and Schroeder-Heister [12] also show examples of this, e.g. for $A \vee (B \to C)$ one can write down the truth table and derive constructive rules, but they are weaker than the constructive formula $A \vee (B \to C)$ (because one basically obtains $B \to (A \vee C)$). The TT-ND derived rules give so called "flat elimination" rules [15], and it is likely that it defines exactly the connectives with flat elimination rules. We conjecture that if $c(A_1, \ldots, A_n)$ is a formula defined in terms of the standard connectives, and we derive constructive rules form the truth table for c, then we get a formula equivalent to $c(A_1, \ldots, A_n)$ if in every subformula (of $c(A_1, \ldots, A_n)$) of the shape $P \vee Q$, $Q \vee P$, $P \to Q$ or $\neg P$, P does not contain negation or implication. Or put differently: in P we only have monotone connectives.

This is related to the general study of elimination rules [11, 19], the notion of higher level rules [16] and "harmony" in logic [15, 5]. It would be interesting to see which class of connectives can be defined using TT-ND, and whether the generic approach can be extended to more connectives, e.g. with higher level elimination rules.

Also, based on generalizing $\lambda\mu$ of Parigot [13], and with a semantic view on dualizing implication Crolard [3] has defined the $-$ connective, which has a constructive interpretation that is different from what we would get from a truth table. The interpretation in Kripke models "looks downward", which our interpretation doesn't do. It would be interesting whether the ideas of Crolard can be generalized to other connectives. The relation between Crolard's work and the work on generalized elimination rules and harmony in logic is also unclear.

Finally, there is the obvious question of how these results extend to predicate logic. We are working on extending the TT-ND ideas to predicate logic and define general rules, both classical and constructive for quantifiers, "in isolation", that is without explaining them in terms of other quantifiers.

References

1 Andreas Abel. On model-theoretic strong normalization for truth-table natural deduction. In Ugo de'Liguoro, Stefano Berardi, and Thorsten Altenkirch, editors, *26th International Conference on Types for Proofs and Programs, TYPES 2020, March 2-5, 2020, University of Turin, Italy*, volume 188 of *LIPIcs*, pages 1:1–1:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.TYPES.2020.1.

2 Z. Ariola and H. Herbelin. Minimal classical logic and control operators. In *ICALP*, volume 2719 of *LNCS*, pages 871–885. Springer, 2003.

3 T. Crolard. A formulae-as-types interpretation of subtractive logic. *J. Log. Comput.*, 14(4):529–570, 2004.

4 P.-L. Curien and H. Herbelin. The duality of computation. In *ICFP*, pages 233–243, 2000.

5 N Francez and R Dyckhoff. A note on harmony. *Journal of Philosophical Logic*, 41(3):613–628, June 2012. doi:10.1007/s10992-011-9208-0.

6 H Geuvers and T Hurkens. Deriving natural deduction rules from truth tables. In *Logic and Its Applications – 7th Indian Conference, ICLA 2017, Kanpur, India, January 5-7, 2017, Proceedings*, volume 10119 of *Lecture Notes in Computer Science*, pages 123–138, 2017. doi:10.1007/978-3-662-54069-5_10.

7 H Geuvers and T Hurkens. Proof terms for generalized natural deduction. In Andreas Abel, Fredrik Nordvall Forsberg, and Ambrus Kaposi, editors, *23rd International Conference on Types for Proofs and Programs, TYPES 2017, May 29-June 1, 2017, Budapest, Hungary*, volume 104 of *LIPIcs*, pages 3:1–3:39. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.TYPES.2017.3.

8 H. Geuvers, I. van der Giessen, and T. Hurkens. Strong normalization for truth table natural deduction. *Fundam. Inform.*, 170(1-3):139–176, 2019.

9 Tomoaki Kawano, Naosuke Matsuda, and Kento Takagi. Effect of the choice of connectives on the relation between classical logic and intuitionistic logic. *Notre Dame Journal of Formal Logic*, 63(2), 2022. doi:10.1215/00294527-2022-0016.

10 Peter Milne. Inversion principles and introduction rules. In Heinrich Wansing, editor, *Dag Prawitz on Proofs and Meaning*, pages 189–224. Springer International Publishing, Cham, 2015. doi:10.1007/978-3-319-11041-7_8.

11 S. Negri and J. von Plato. *Structural Proof Theory*. Cambridge University Press, 2001.

12 Grigory K. Olkhovikov and Peter Schroeder-Heister. On flattening elimination rules. *Rev. Symb. Log.*, 7(1):60–72, 2014. doi:10.1017/S1755020313000385.

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 S Read. Harmony and autonomy in classical logic. *Journal of Philosophical Logic*, 29(2):123–154, 2000. doi:10.1023/A:1004787622057.

16 Peter Schroeder-Heister. The calculus of higher-level rules, propositional quantification, and the foundational approach to proof-theoretic harmony. *Stud Logica*, 102(6):1185–1216, 2014. doi:10.1007/s11225-014-9562-3.

17 A.S. Troelstra and D. van Dalen. *Constructivism in Mathematics, volume I*. Number volume 121 in Studies in Logic and the Foundations of Mathematics. Elsevier Science, 1988.

18 I. van der Giessen. Natural deduction derived from truth tables, Master Thesis Mathematics, Radboud University Nijmegen, July 2018. URL: http://www.cs.ru.nl/~herman/PUBS/Masterscriptie_IrisvanderGiessen.pdf.

19 J. von Plato. Natural deduction with general elimination rules. *Arch. Math. Log.*, 40(7):541–567, 2001.