Rewriting for Fitch style natural deductions

Herman Geuvers, Rob Nederpelt

Nijmegen University, The Netherlands
Eindhoven University of Technology, The Netherlands

Abstract. Logical systems in natural deduction style are usually presented in the Gentzen style. A different definition of natural deduction, that corresponds more closely to proofs in ordinary mathematical practice, is given in [Fitch 1952]. We define precisely a Curry-Howard interpretation that maps Fitch style deductions to simply typed terms, and we analyze why it is not an isomorphism. We then describe three reduction relations on Fitch style natural deductions: one that removes *garbage* (subproofs that are not needed for the conclusion), one that removes *repeats* and one that *unshares* shared subproofs. We also define an equivalence relation that allows to *interchange* independent steps. We prove that two Fitch deductions are mapped to the same λ-term if and only if they are equal via the congruence closure of the aforementioned relations (the reduction relations plus the equivalence relation). This gives a Curry-Howard isomorphism between equivalence classes of Fitch deductions and simply typed λ-terms. Then we define the notion of cut-elimination on Fitch deductions, which is only possible for deductions that are completely unshared (normal forms of the unsharing reduction). For conciseness, we restrict in this paper to the implicational fragment of propositional logic, but we believe that our results extend to full first order predicate logic.

1 Introduction

For Gentzen style natural deduction, ([Gentzen 1969]) there is a well-known notion of rewriting: cut-elimination. This is a procedure for eliminating 'detours' in a logical derivation that arise from first applying an introduction rule and then an elimination rule for a connective. This notion of reduction can be defined more concisely by associating typed λ-terms to natural deductions: there is the Curry-Howard isomorphism between natural deductions and simply typed terms and cut-elimination in the first corresponds to β-reduction in the latter (see [Howard 1980]). A different definition of natural deduction is *flag style natural deduction* defined by [Fitch 1952]. See [Bornat and Sufrin 1999] for a nice implementation of a proof assistant based on flag deductions. Here, deductions are linear, written vertically where every line consists of a formula and a motivation for the derivability of that formula (referring to previous lines). Furthermore, there is a notion of 'scope' (of an assumption) which is indicated by a flag. Apart from the closer correspondence to proofs in mathematical practice, a positive aspect is that subproofs can be shared. A negative aspect is that, due to the sharing, the notion of cut-elimination is blurred. Also, the order of the steps in a Fitch style deduction is somewhat arbitrary: a flag deduction can be seen as a linearization of a Gentzen style natural deduction tree and this linearization involves some arbitrary (bureaucratic) choices. This implies that one Gentzen style *tree deduction* corresponds to many flag deductions. We make this precise by using simply typed λ-calculus (see [Barendregt 1992]) and to define a Curry-Howard *formulas-as-types* (and *proofs-as-terms*) interpretation from flag deductions to typed terms. The Curry-Howard interpretation that we define here is not the only possible one. We will come back to this point briefly in Section 5. The interest of our interpretation lies in the fact that it ignores those aspects of flag deductions that can be seen as 'bureaucracy'. To keep things simple and for space restrictions, we restrict here to simply typed λ-calculus with just arrow types and for the logic to propositional logic with just implication.

2 Flag style Natural Deduction

We consider the implicational fragment, so the set of formulas, Form, is built up from a set of parameters, Par, using the implication $\rightarrow$. The rules have the same style as for Gentzen

style tree deduction, fixing the meaning of a connective by saying how to eliminate (an E-rule) it and how to introduce it (an I-rule).

Flag deduction, a first definition

Definition 2.1. *The rules for natural deductions in flag style are the following.*

$\to$-*Introduction*

$$\begin{array}{l|l|l} n & \vdots & \\ & \vdots & \\ m & B & \\ m+1 & A\to B & \to I,\ n,\ m \end{array}$$

$\to$-*Elimination*

$$\begin{array}{l|l|l} \vdots & \vdots & \\ n & A\to B & \\ \vdots & \vdots & \\ m & A & \\ \vdots & \vdots & \\ l & B & \to E,\ n,\ m \end{array}$$

Repeat

$$\begin{array}{l|l|l} \vdots & \vdots & \\ n & B & \\ \vdots & \vdots & \\ m & A & \\ \vdots & \vdots & \\ l & B & R,\ n \end{array}$$

Hypothesis m | A

Remark 2.2. The order in which the lines appear in a flag deduction should be exactly as suggested by the above diagrams, except for the rule $\to$-E, where $A\to B$ and A may be interchanged. All rules can be applied 'under a flag'. Furthermore, the repeat rule and the $\to$-E rule can take their premise (the B, resp. the A and the $A\to B$) from arbitrarily high, with the proviso that B must be 'in scope', i.e. it must not be 'under a closed flag'.

The repeat rule allows to use one sub-deduction several times. Its use will be discussed later. Some aspects of the definition are vague: we have not defined what it means for a formula to be 'in scope'. The idea is that the $\to$-I rule closes a sub-deduction and that the formulas in that sub-deduction are not in scope anymore. The definition of flag deductions above lacks precision, especially if we want to study their structure in detail. We will therefore make the definition more precise in Definition 2.6, but first we give some examples.

Example 2.3. The following two examples give an impression of flag deductions.

D_1

1	$A\to B$	
2	$\quad A$	
3	$\quad\quad A$	
4	$\quad\quad B$	$\to$E, 1, 2
5	$\quad A\to B$	$\to$I, 3, 4
6	$A\to A\to B$	$\to$I, 2, 5
7	$(A\to B)\to A\to A\to B$	$\to$I, 1, 6

D_2

1	A	
2	$\quad B$	
3	$\quad A$	R, 1
4	$B\to A$	$\to$I, 2, 3
5	$A\to B\to A$	$\to$I, 1, 4

D_3

n	A	
$\vdots$	$\vdots$	
m	$A\to A\to B$	
$m+1$	A	R, n
$m+2$	$A\to B$	$\to$E, m, $m+1$
$m+3$	B	$\to$E, $m+2$, $m+1$

D_4

1	$A\to B\to C$	
2	$\quad A\to B$	
3	$\quad\quad A$	
4	$\quad\quad B$	$\to$E, 2, 3
5	$\quad\quad B\to C$	$\to$E, 1, 3
6	$\quad\quad C$	$\to$E, 5, 4
7	$\quad A\to C$	$\to$I, 3, 6
8	$(A\to B)\to A\to C$	$\to$I, 2, 7
9	$(A\to B\to C)\to(A\to B)\to A\to C$	$\to$I, 1, 8

In D_1, there are two possible ways of deriving B on line 4 $\rightarrow$-E, 1, 2 or $\rightarrow$-E1, 3. As it is recorded in the motivation at the end of the line ($\rightarrow$-E, 1, 2 in this case), we can distinguish these two deductions: we will consider these two flag deductions as different. In D_2 and D_3 we see two possible uses of the repeat rule. D_4 is a derivation of a well-known axiom of Hilbert style deduction.

In D_3 we can avoid the use of (repeat), because we can take premises for the $\rightarrow$-E rule from arbitrary high; the use of an explicit (repeat) is for readability only. In D_2, we can only avoid the use of (repeat) if we allow $\rightarrow$-introduction without explicitly writing the conclusion as the last formula. It's a matter of taste (and choice) whether one wants to allow this. If one does, we can omit the use of (repeat) completely from our deductions. We choose for this option, so we can change D_2 into the following correct deduction.

$$D_2 \quad \begin{array}{l|l|l} 1 & \boxed{A} & \\ 2 & \boxed{\begin{array}{l|l} & B \end{array}} & \\ 3 & B{\rightarrow}A & \rightarrow\text{I, 2, 1} \\ 4 & A{\rightarrow}B{\rightarrow}A & \rightarrow\text{I, 1, 3} \end{array}$$

Flag deduction, an improved definition We have already pointed out that Definition 2.1 is a bit informal, especially as to which formulas are 'in scope' (and may henceforth be repeated or used in the $\rightarrow$-E rule). To define mappings between flag deductions and typed λ-terms, we give a more precise definition of flag deductions. In this definition, a flag deduction consists of a sequence of tuples $\langle l_0, A_0, m_0\rangle, \langle l_1, A_1, m_1\rangle, \ldots, \langle l_n, A_n, m_n\rangle$. Here each tuple is of the form $\langle l, A, m\rangle$, where $l \in \mathcal{I}$ is the *label*, taken from a countable set $\mathcal{I}$ (usually $\mathbb{N}$) A is a formula (the formula derived at that line) and m is the *motivation* for the derivation. This motivation contains the following information: which rule has been applied to derive the formula and on which lines the derivation is based. The motivations m will be of the following forms. (We write $-$ to mean 'nothing', to be used for hypotheses.)

m	meaning
-	*hypothesis* ("raising" a flag)
$\rightarrow$-E, l_1, l_2	$\rightarrow$*-elimination* on the formulas on the lines with labels l_1, l_2
$\rightarrow$-I, l_1, l_2	$\rightarrow$*-introduction* on the formulas on the lines with labels l_1, l_2
R, l	*repeat* the formula on the line with label l

Definition 2.4. *1. If Σ is a sequence of tuples $\langle l, A, m\rangle$ as above where each label occurs at most once as a line number, we call it a* pre-deduction.
2. Given a pre-deduction Σ and an $l' \in \mathcal{I}$, we say that l' is Σ-fresh *if l' does not yet occur in any of the tuples $\langle l, A, m\rangle$ of Σ.*
3. We say that the formula A is on line l in Σ *(or just A is on line l, if Σ is clear from the context), if the tuple $\langle l, A, \ldots\rangle$ occurs in Σ.*

We will write a pre-deduction in the shape of a flag deduction, like the ones in Example 2.3. So a line $\langle m, A, -\rangle$ is depicted as

$$m \quad \begin{array}{l|l} & \boxed{A} \end{array}$$

This raises a flag, whose "flagpole" we extend to the *last* line with and $\rightarrow$-I, m, l motivation. Crucial notions in the definition of flag deductions are the *scope* of a deduction and the *flag* of a deduction. Given a deduction Σ, Scope(Σ) will be the set of lines in Σ from which we can 'use' formulas (at the end of Σ). Flag(Σ) is the line l of the 'last open flag'.

Definition 2.5. *For a pre-deduction Σ, we define the* scope *of Σ,* Scope(Σ)*, as follows*[1]

$$\text{Scope}\left(\begin{array}{l|l|l} \vdots & \Sigma_1 & \\ m & \boxed{\begin{array}{l|l} & A \\ \hline & \Sigma_2 \end{array}} & \\ \vdots & & \\ l & B & \rightarrow\text{-I, } m,n \end{array}\right) = \text{Scope}(\Sigma_1) \cup \{(l, B)\}$$

[1] For clarity, we give this definition by writing the deductions graphically. We could equivalently give it on the basis of the pre-deductions of Definition 2.4, but that just blurs the presentation.

$$\text{Scope}\left(\begin{array}{c|ll}\vdots & \Sigma & \\ l & A & m\end{array}\right) = \text{Scope}(\Sigma) \cup \{(l, A)\} \text{ if } m \neq [\rightarrow\text{-}I, \ldots, \ldots]$$

In the first clause, we have $[\rightarrow\text{-}I, m, n]$ on the last line and we "search" upward for the line m containing a hypothesis (flag); if there is no line m or if line m contains no hypothesis, the scope is $\emptyset$.
For Σ a pre-deduction, we define Flag(Σ) *as follows.*

$$\text{Flag}\left(\begin{array}{c|l}\vdots & \Sigma \\ l & \underline{\lfloor A} \end{array}\right) = (l, A)$$

$$\text{Flag}\left(\begin{array}{c|ll}\vdots & \Sigma_1 & \\ m & \underline{\lfloor A} & \\ \vdots & \lfloor \Sigma_2 & \\ l & B & \rightarrow\text{-}I,\ m,n\end{array}\right) = \text{Flag}(\Sigma_1)$$

$$\text{Flag}\left(\begin{array}{c|l}\vdots & \Sigma \\ l & A \end{array}\right) = \text{Flag}(\Sigma) \text{ otherwise}$$

Note that Scope *may yield an empty set and that* Flag *may be undefined.*

Definition 2.6. *We inductively define the notion of* flag deduction.

1. *(Flag raising) If Σ is a flag deduction or $\Sigma = \emptyset$ and l is a fresh label, then*

$$\begin{array}{c|l}\vdots & \Sigma \\ l & \underline{\lfloor A} \end{array}$$

   is a flag deduction.
2. *($\rightarrow$-E) If Σ is a flag deduction with $(l_1, A{\rightarrow}B), (l_2, A) \in \text{Scope}(\Sigma)$ and l is fresh, then*

$$\begin{array}{c|ll}\vdots & \Sigma & \\ l & B & \rightarrow\text{-}E, l_1, l_2\end{array}$$

   is a flag deduction.
3. *($\rightarrow$-I) If Σ is a flag deduction with* Flag$(\Sigma) = (l_1, A)$, $(l_2, B) \in \text{Scope}(\Sigma)$ *and l is a fresh label, then*

$$\begin{array}{c|ll}\vdots & \Sigma & \\ l & A{\rightarrow}B & \rightarrow\text{-}I, l_1, l_2\end{array}$$

   is a flag deduction.
4. *(Repeat) If Σ is a flag deduction with $(l', A) \in \text{Scope}(\Sigma)$ and l is a fresh label, then*

$$\begin{array}{c|ll}\vdots & \Sigma & \\ l & A & R,\ l'\end{array}$$

   is a flag deduction.

Definition 2.7. *Let Σ be a flag deduction.*

1. *The* conclusion *of Σ is the formula on the last line of Σ.*
2. *The* assumptions *of Σ are the formulas in the flags in Σ that are open (i.e. that have not been closed by an $\rightarrow$-I rule). More formally, the assumptions of Σ are the formulas A for which*
$$\exists l \in \mathcal{I}(\langle l, A, -\rangle \in \Sigma \wedge (l, A) \in \text{Scope}(\Sigma)).$$
3. *For Δ a set of formulas and A a formula, $\Delta \vdash_f A$ if there is a flag-deduction Σ with conclusion A and assumptions in Δ.*

2.1 The problem with cut-elimination

In natural deduction, we speak of a *cut* if we first introduce a connective (via an intro rule) and then immediately eliminate it (via an elim rule for that same connective). In our case, that means: doing an $\rightarrow$-I to introduce, say $A \rightarrow B$ and then doing an $\rightarrow$-E to derive B. (In full proposition logic there are more notions of cut: "commuting cuts", see [Prawitz 1965]) that allow deduction rules to be interchanged, but that's not of interest here.) In Fitch style, this would amount to the following, where we denote the process of cut-elimination by $\Longrightarrow_c$.

$$
\begin{array}{l|l|l}
1 & \boxed{\begin{array}{l} A \\ \hline \Sigma \\ B \end{array}} & \\
\vdots & & \\
n & & \\
n+1 & A \rightarrow B & \rightarrow\text{I}, 1, n \\
\vdots & \Theta & \\
m & A & \\
\vdots & \Pi & \\
l & B & \rightarrow\text{E}, n+1, m
\end{array}
\quad \Longrightarrow_c \quad
\begin{array}{l|l}
\vdots & \Theta \\
m & A \\
\vdots & \Sigma[m/1] \\
n & B \\
\vdots & \Pi
\end{array}
$$

The problem here is that the right derivation may not be well-formed: if in the sub-derivations Θ or Π the formula $A \rightarrow B$ on line $n+1$ is used, the right hand side derivation has invalid line references (referring to non-existent lines). The problem with cut-elimination is due to *sharing* of sub-derivations. To give a precise definition of cut-elimination we first define the formulas-as-types interpretation from flag deductions to simply typed λ-calculus.

3 Formulas-as-types for Fitch style Natural Deduction

For flag deductions, we define an interpretation into the set of simply typed λ-terms. For the mapping back, we will need *labeled* simply typed λ-terms (to be able to restore all the labels in the flag deduction). Labeled terms are terms where all sub-terms (except the variables) are labeled with a label from $\mathcal{I}$:

$$\mathsf{LTerm} ::= \mathsf{Var}^{\mathsf{Typ}}_{\mathcal{I}} \mid (\mathsf{LTerm}\ \mathsf{LTerm})^{\mathcal{I}} \mid (\lambda \mathsf{Var}^{\mathsf{Typ}}_{\mathcal{I}}.\mathsf{LTerm})^{\mathcal{I}}$$

The typing rules are the usual: we allow any $l \in \mathcal{I}$ to be used as a label. So we have
- $x_i^A : A$,
- $(MN)^l : B$ if $M : A \rightarrow B$, $N : A$ and $l \in \mathcal{I}$,
- $(\lambda x_i^A.M)^l : A \rightarrow B$ if $M : B$ and $l \in \mathcal{I}$.

If we want to denote the label of a sub-term explicitly (in an application or an abstraction) we write $(M^{l_1} N^{l_2})^l$ or $(\lambda x_i^A.M^{l_1})^l$.

Definition 3.1. *The set of labels and variable indices of a labeled λ-term M will be denoted by $lab(M)$. So,*
- $lab(x_i^A) = \{i\}$,
- $lab((MN)^l) = \{l\} \cup lab(M) \cup lab(N)$,
- $lab((\lambda x_i^A.M)^l) = \{l, i\} \cup lab(M)$.

Similarly, we will denote the set of lines of a flag deduction Σ by $lab(\Sigma)$. (These are the 'line numbers' that occur in Σ, which are taken from the same index set $\mathcal{I}$ of the labels and indices of λ-terms.)

There is a straightforward mapping from the labeled to the unlabeled terms, by just erasing labels. We denote it by $|\_|$, so if $M \in \mathsf{LTerm}$, then $|M| \in \mathsf{Term}$. We view a labeled term as a specific representation of an unlabeled term. Stated otherwise, we consider the terms *modulo relabeling*. However, we do not just allow any kind of relabeling, but only *injective* ones, because we want to be able to distinguish, e.g. $\lambda f^{A \rightarrow A \rightarrow B}.\lambda g^{B \rightarrow A}.\lambda x^B.((f(gx)^4)^6(gx)^5)^7$ and $\lambda f^{A \rightarrow A \rightarrow B}.\lambda g^{B \rightarrow A}.\lambda x^B.((f(gx)^4)^6(gx)^4)^7$.

Definition 3.2. *A* relabeling *is an injective map $r : \mathcal{I} \rightarrow \mathcal{I}$. By* adding a labeling *to a simply typed λ-term N, we mean to construct a labeled simply typed term M such that $|M| = N$ and all labels and variable-indices in M are unique (i.e. distinct sub-terms of M have different labels and similar for variables).*

A relabeling $r : \mathcal{I} \to \mathcal{I}$ extends immediately to a map on labeled simply typed terms and to a map on flag deductions. If the labeled term M arises from P as a result of some relabeling r, we say that M is a *relabeling of* P.
For mapping simply typed λ-terms to flag deductions, we first add a labeling and then we define the associated flag deduction. The mapping of flag deductions to simply typed λ-terms can be defined directly (without using labeled terms). As we will be using the labeled terms later, we define the map from flag deductions to labeled terms.

Definition 3.3. *The interpretation of a flag deduction Σ as a labeled simply typed λ-term, $[\![\Sigma]\!]^L$, is defined as follows. (We use the cases of Definition 2.6.)*

1. *(Flag raising) If the last rule in Σ is a flag, then $[\![\Sigma]\!]^L = x_l^A$.*
2. *($\to$-E) If the last rule in Σ is $\to$-E, then then $[\![\Sigma]\!]^L = ([\![\Sigma{\leq}l_1]\!]^L [\![\Sigma{\leq}l_2]\!]^L)^l$.*
3. *($\to$-I) If the last rule in Σ is $\to$-I then $[\![\Sigma]\!]^L = (\lambda x_{l_1}^A.[\![\Sigma{\leq}l_2]\!]^L)^l$.*
4. *(Repeat) If the last rule in Σ is R, then $[\![\Sigma]\!]^L = [\![\Sigma{\leq}l']\!]^L$.*

Here $\Sigma{\leq}l$ denotes the pre-deduction Σ up to and including l. The interpretation of a flag deduction as a simply typed term is defined by just erasing the labels, so

$$[\![\Sigma]\!] := |[\![\Sigma]\!]^L|.$$

When treating examples, we will be using the usual notation for flag deductions, with real flags and flag poles to indicate the scope of a flag. It should be clear how we can transform the formal notation of flag deductions (as used above) into a deduction with 'real' flags and flag poles.

Example 3.4. We show the interpretation of two flag deductions as typed λ-terms. For readability, we write the type labels of the variables only in the λ-abstraction.

1	$A{\to}A{\to}B$	
2	| A	
3	| $A{\to}B$	$\to$E, 1, 2
4	| B	$\to$E, 3, 2
5	$A{\to}B$	$\to$I, 2, 4
6	$(A{\to}A{\to}B){\to}(A{\to}B)$	$\to$I, 1, 5

1	A	
2	| B	
3	| A	R, 1
4	$B{\to}A$	$\to$I, 2, 3
5	$A{\to}B{\to}A$	$\to$I, 1, 4

The typed λ-term associated to these terms are $\lambda x_1^{A\to A\to B}.\lambda x_2^A.(x_1 x_2)x_2$ and $\lambda x_1^A.\lambda x_2^B.x_1$.

Theorem 3.5 (Soundness of $[\![-]\!]$). *If Σ is a flag deduction with conclusion A, then $[\![\Sigma]\!] : A$. Moreover, if the assumptions of Σ are $B_1, \ldots, B_n$, with labels $i_1, \ldots, i_n$, respectively, then $FV([\![\Sigma]\!]) = \{x_{i_1}^{B_1}, \ldots, x_{i_n}^{B_n}\}$.*

Proof. By induction on the flag deduction Σ. □

In the definition of the opposite embedding, we assume that the terms are *uniquely labeled*, that is, distinct sub-terms of M have different labels and all the indices of the bound variables are distinct and distinct from the labels. To define the embedding, we first ignore the free variables (i.e. we don't raise flags), defining $(\![-]\!)^p$. Then we raise flags for all free variables, defining $(\![-]\!)$. There are basically two versions of the embedding: a simple minded one that just creates a sub-derivation for every sub-term and uses a lot of instances of the repeat rule, and a more sophisticated one that does not create any repeat rule. We only treat the second one.

Definition 3.6 (Simply typed terms $\to$ Flag deductions). *The interpretation of a labeled simply typed λ-term M as a pre-deduction, $(\![M]\!)^p$, is defined as follows.*

1. *For $(\![(M^{l_1} N^{l_2})^l]\!)^p$ distinguish cases according to the shape of M and N. Let B be the type of $(M^{l_1} N^{l_2})^l$.*
   (a) M and N are variables:

$$(\![(x_i^{A\to B}\, x_j^A)^l]\!)^p = \begin{array}{l} l \\ \hline B \end{array} \quad \to\text{-}E,i,j$$

(b) M is a variable, N is not:

$$(\!|(x_i^{A\to B}N^{l_2})^l|\!)^p = \begin{array}{l|ll} \vdots & (\!|N^{l_2}|\!)^p & \\ l & B & \to\text{-}E,i,l_2 \end{array}$$

(c) N is a variable, M is not:

$$(\!|(M^{l_1}x_j^A)^l|\!)^p = \begin{array}{l|ll} \vdots & (\!|M^{l_1}|\!)^p & \\ l & B & \to\text{-}E,l_1,j \end{array}$$

(d) Neither M nor N is a variable:

$$(\!|(M^{l_1}N^{l_2})^l|\!)^p = \begin{array}{l|ll} \vdots & (\!|M^{l_1}|\!)^p & \\ \vdots & (\!|N^{l_2}|\!)^p & \\ l & B & \to\text{-}E,l_1,l_2 \end{array}$$

2. *For $(\!|(\lambda x_i^A.M)^l|\!)^p$ distinguish cases according to whether M is a variable or not. Let B be the type of M.*
(a)

$$(\!|(\lambda x_i^A.x_j^B)^l|\!)^p = \begin{array}{l|ll} i & \lfloor A & \\ l & A\to B & \to\text{-}I,i,j \end{array}$$

(b)

$$(\!|(\lambda x_i^A.M^{l_1})^l|\!)^p = \begin{array}{l|ll} i & \left|\begin{array}{l} A \\ \hline (\!|M^{l_1}|\!)^p \end{array}\right. & \\ \vdots & & \\ l & A\to B & \to\text{-}I,i,l_1 \end{array}$$

The interpretation of a labeled simply typed λ-term M as a full flag deduction, $(\!|M|\!)$, is defined by adding flags for the free variables:
If $FV(M) = \{x_{i_1}^{B_1}, \ldots, x_{i_n}^{B_n}\}$, then $(\!|M|\!)$ is

$$\begin{array}{l|ll} i_1 & \left|\begin{array}{l} B_1 \\ \hline \vdots \end{array}\right. & \\ \vdots & & \\ i_n & \left|\begin{array}{l} B_n \\ \hline (\!|M|\!)^p \end{array}\right. & \\ \vdots & & \end{array}$$

Example 3.7. Compare with 3.4. We define the interpretation of two labeled λ-terms as flag deductions. For readability, we write the type labels only in the λ-abstractions. Consider the labeled typed λ-terms $(\lambda x_1^{A\to A\to B}.(\lambda x_2^A.((x_1\,x_2)^3\,x_2)^4)^5)^6$ and $(\lambda x_1^A.(\lambda x_2^B.x_1)^3)^4$. Their interpretations are as follows.

$$\begin{array}{l|ll} 1 & A\to A\to B & \\ \hline 2 & \left|\begin{array}{l} A \\ \hline A\to B \\ B \\ \hline A\to B \end{array}\right. & \begin{array}{l} \\ \to\text{E, 1, 2} \\ \to\text{E, 3, 2} \\ \to\text{I, 2, 4} \end{array} \\ 6 & (A\to A\to B)\to(A\to B) & \to\text{I, 1, 5} \end{array} \qquad \begin{array}{l|ll} 1 & \left|\begin{array}{l} A \\ \hline B \\ \hline B\to A \end{array}\right. & \to\text{I, 2, 1} \\ 4 & A\to B\to A & \to\text{I, 1, 3} \end{array}$$

Observe that the (repeat) rule does not occur anymore in the resulting deductions. In the first case, we get the same deduction as the 'original' one of 3.4. In the second deduction, we obtain a different deduction from the one of 3.4: the (repeat) rule has been removed. So in general, it is *not* the case that $(\!|-|\!) \circ [\![-]\!]$ is the identity: if we start from a flag deduction, map it to a λ-term and back, we will sometimes arrive at a different flag deduction then the one we started from. Apart from the repeat rule, there are more interesting cases sources for the non-isomorphism. This will be discussed in detail in Section 4. The other way around, it *is* the case that $[\![-]\!] \circ (\!|-|\!)$ is the identity on simply typed λ-terms. First we give the soundness theorem for the second interpretation of flag deductions as typed λ-terms.

Theorem 3.8 (Soundness of $(\![-]\!]$). *If M is a labeled term of type A, then $(\![M]\!]$ (following Definition 3.6) is a flag deduction with conclusion A. Moreover, if $FV(M) = \{x_{i_1}^{B_1}, \ldots, x_{i_n}^{B_n}\}$ then the assumptions of $(\![M]\!]$ are $B_1, \ldots, B_n$, with labels $i_1, \ldots, i_n$, respectively.*

Proof. By induction on M. □

Theorem 3.9. *Given a simply typed λ-term M,*

$$[\![(\![M]\!)]\!] \equiv M$$

Proof. By induction on M. □

4 The fine structure of flag deductions

We have already pointed out that there is no isomorphism between simply typed terms and flag deductions, because $(\![-]\!] \circ [\![-]\!]$, is not the identity. There are various origins for this non-isomorphism and we categorize them.
Given a term M, we refer to $(\![M]\!]$ as the *canonical flag deduction* for M. If Σ is a flag deduction, we will also call $(\![[\![\Sigma]\!]]\!)$ the *canonical form of Σ*, because it is the canonical flag deduction for $[\![\Sigma]\!]$. This yields the class of *canonical flag deductions*. There is an obvious isomorphism between the typed λ-terms and the canonical flag deductions (modulo relabeling). We want the equivalence relation on flag deductions, that we alluded to before, to be defined in such a way that the class of canonical flag deductions forms a complete set of representatives. This characterizes the canonical forms from a different point of view, purely in terms of flag deductions.

R: Repeat Rule In flag deductions, the (repeat) rule can be applied everywhere. We don't want to distinguish two deductions that only differ in the applications of the (repeat) rule. The interpretation $[\![-]\!]$ maps such deductions to the same λ-term.

G: Garbage (dead ends) In a flag deduction, there may be parts that do not contribute to the final result. We can call these parts 'garbage' or 'dead ends'. Garbage can be detected by looking at the set of lines that the conclusion depends on (following the 'motivations', collecting all lines starting from the conclusion); all lines that are not encountered in this way are garbage. Note that one can not just remove one "garbage line", because there may be other (garbage) lines depending on it. (So one has to start removal with the last garbage line.)

I: Permutation of independent steps The precise order of the deduction steps is somewhat arbitrary. See the following two examples.

1	A			1	A		
2	$A \to B$			2	$A \to C$		
3	$A \to C$			3	C	$\to$E, 2, 1	
4	$B \to C \to D$			4	$A \to B$		
5	C	$\to$E, 3, 1		5	B	$\to$E, 4, 1	
6	B	$\to$E, 2, 1		6	$B \to C \to D$		
7	$C \to D$	$\to$E, 4, 6		7	$C \to D$	$\to$E, 6, 5	
8	D	$\to$E, 7, 5		8	D	$\to$E, 7, 3	

The conclusion C, now given in line 5, can be located anywhere between lines 3 and 8. Two consecutive flags can also be permuted, as long as they are not closed. A flag can also move over a formula, if the derivability of the formula does not depend on the flag. This is shown in the second deduction. We view all these changes in a deduction as a *permutation of independent steps*.

S: Sharing of subproofs Consider the following two flag deductions

1	B	
2	$(A{\to}B){\to}(A{\to}B){\to}C$	
3	A	
4	B	R, 1
5	$A{\to}B$	$\to$I, 3, 4
6	$(A{\to}B){\to}C$	$\to$E, 2, 5
7	C	$\to$E, 6, 5

1	B	
2	$(A{\to}B){\to}(A{\to}B){\to}C$	
3	A	
4	$A{\to}B$	$\to$I, 3, 1
5	$(A{\to}B){\to}C$	$\to$E, 2, 4
6	A	
7	$A{\to}B$	$\to$I, 6, 1
8	C	$\to$E, 5, 7

In the left deduction, the conclusion $A{\to}B$ is used twice (in lines 6 and 7), to do an $\to$-elimination. The subproof of $A{\to}B$ (lines 3–5) is 'shared' by the two applications of $\to$-E on lines 6 and 7. Although the example above is obviously quite trivial (for reasons of exposition), this is clearly a great advantage of flag deductions over ordinary natural deduction (and simply typed λ-terms): a result that has been derived in a certain context can be reused (i.e. as a source it can be 'shared' by consecutive other rules). In ordinary natural deduction, the result $A{\to}B$ would have to be derived twice. In terms of simply typed λ-calculus, this means that one and the same sub-term will occur several times. We illustrate this by computing the λ-term for this flag deduction, which is

$$y_2(\lambda z^A.x_1)(\lambda z^A.x_1) : C,$$

where x_1^B is the variable associated to line 1 and $y_2^{(A\to B)\to(A\to B)\to C}$ is the variable associated to line 2. The flag deduction associated with this λ-term is the deduction to the right above. Observe that the subproof of $A{\to}B$ has been copied and occurs twice in this flag deduction. Also the repeat rule has been removed.

We now define an equivalence relation $\simeq_i$ on flag deductions that equates flag deductions that only differ as a consequence of permutation of independent steps. We also define 3 reduction relations, $\longrightarrow_r$ that removes repeat rules, $\longrightarrow_g$ that removes garbage and $\longrightarrow_s$ that unshares deductions.

Definition 4.1. *Define the equivalence relation $\simeq_i$ on flag deductions as the reflexive, symmetric, transitive closure of the following relations.*

1. *(Interchange of lines) If $\Sigma \neq \emptyset$, then*

$\vdots$	Θ		$\simeq_i$	$\vdots$	Θ	
l	A	m		l'	B	m'
l'	B	m'		l	A	m
$\vdots$	Σ			$\vdots$	Σ	

   Note that, as we assume both sides of the $\simeq_i$ to be well-formed flag deductions, it must be the case that $l \notin m'$, and $l' \notin m$.

2. *(Interchange of blocks) If $\Sigma \neq \emptyset$, then*

$\vdots$	Θ		$\simeq_i$	$\vdots$	Θ	
n	A			l	B	m'
$\vdots$	Π			n	A	
m	$A{\to}C$	$\to$-I,n,p		$\vdots$	Π	
l	B	m'		m	$A{\to}C$	$\to$-I,n,p
$\vdots$	Σ			$\vdots$	Σ	

   Note that, as we assume both sides of the $\simeq_i$ to be well-formed flag deductions, it must be the case that $l \notin m'$, and $l' \notin \Pi$.

Remark 4.2 (to the Definition). Note that the well-formedness of the right hand side is not automatically implied by the well-formedness of the left hand side. We assume $\Sigma \neq \emptyset$, because if $\Sigma = \emptyset$, then the deduction on the left of the $\simeq_i$ may have a different conclusion from the one to the right.

Definition 4.3. *The reduction relations $\longrightarrow_g$, $\longrightarrow_r$ and $\longrightarrow_s$ on flag deductions are defined as follows.*

1. *If $\Sigma \neq \emptyset$ and $l \notin \Sigma$, then*

$$\begin{array}{l|ll} \vdots & \Theta & \\ n & \begin{array}{|l} A \\ \hline \Pi \end{array} & \\ \vdots & & \\ l & A{\to}C & \to\text{-}I,n,p \\ \vdots & \Sigma & \end{array} \longrightarrow_g \begin{array}{l|l} \vdots & \Theta \\ \vdots & \Sigma \end{array}$$

2. *If $\Sigma \neq \emptyset$, $l \notin \Sigma$ and $m \neq [I,-,-]$, then*

$$\begin{array}{l|ll} \vdots & \Theta & \\ l & A & m \\ \vdots & \Sigma & \end{array} \longrightarrow_g \begin{array}{l|l} \vdots & \Theta \\ \vdots & \Sigma \end{array}$$

3. *If $\Sigma \neq \emptyset$, then*

$$\begin{array}{l|ll} \vdots & \Theta & \\ l & A & R,m \\ \vdots & \Sigma & \end{array} \longrightarrow_r \begin{array}{l|l} \vdots & \Theta \\ \vdots & \Sigma[l/m] \end{array}$$

4.

$$\begin{array}{l|ll} \vdots & \Theta & \\ m & \begin{array}{|l} A \\ \hline A \end{array} & \\ l & & R,\ m \end{array} \longrightarrow_r \begin{array}{l|l} \vdots & \Theta \\ m & \begin{array}{|l} A \\ \hline \end{array} \end{array}$$

5. *If $m = [R,k]$ or $m = [{\to}\text{-}E, k_1, k_2]$, then*

$$\begin{array}{l|ll} \vdots & \Theta_1 & \\ l & A & m \\ \vdots & \Theta_2 & \\ l_0 & B_1 & \ldots,l\ldots \\ \vdots & \Theta_3 & \\ l_1 & B_2 & \ldots,l\ldots \\ \vdots & \Sigma & \end{array} \longrightarrow_s \begin{array}{l|ll} \vdots & \Theta_1 & \\ l & A & m \\ l' & A & m \\ \vdots & \Theta_2 & \\ l_0 & B_1 & \ldots,l\ldots \\ \vdots & \Theta_3 & \\ l_1 & B_2 & \ldots,l'\ldots \\ \vdots & \Sigma & \end{array}$$

6.

$$\begin{array}{l|ll} \vdots & \Theta_1 & \\ n & \begin{array}{|l} A \\ \hline \Pi \end{array} & \\ \vdots & & \\ l & A{\to}C & \to\text{-}I,n,p \\ \vdots & \Theta_2 & \\ l_0 & B_1 & \ldots,l\ldots \\ \vdots & \Theta_3 & \\ l_1 & B_2 & \ldots,l\ldots \\ \vdots & \Sigma & \end{array} \longrightarrow_s \begin{array}{l|ll} \vdots & \Theta_1 & \\ n & \begin{array}{|l} A \\ \hline \Pi \end{array} & \\ \vdots & & \\ l & A{\to}C & \to\text{-}I,n,p \\ n' & \begin{array}{|l} A \\ \hline \Pi' \end{array} & \\ \vdots & & \\ l' & A{\to}C & \to\text{-}I,n',p' \\ \vdots & \Theta_2 & \\ l_0 & B_1 & \ldots,l\ldots \\ \vdots & \Theta_3 & \\ l_1 & B_2 & \ldots,l'\ldots \\ \vdots & \Sigma & \end{array}$$

where Π' is just Π with fresh line numbers.

The transitive reflexive symmetric closure of $\longrightarrow_g, \longrightarrow_r, \longrightarrow_s$ *and* $\simeq_i$ *will be denoted by* $\simeq_{gris}$.

Remark 4.4. In the reduction rules, apart from *lines* being repeated, garbage or interchangeable, we also have to take into accounts *blocks*: parts of a proof that are 'guarded' by an $\rightarrow$-I-rule.

Theorem 4.5 (Preservation of equality under $[\![-]\!]^L$ and $[\![-]\!]$). *Let the two flag deductions* Σ *and* Θ *be given. If* $\Sigma \simeq_i \Theta$ *or* $\Sigma \longrightarrow_{gr} \Theta$ *, then* $[\![\Sigma]\!]^L \equiv [\![\Theta]\!]^L$. *If* $\Sigma \longrightarrow_s \Theta$, *then* $[\![\Sigma]\!] \equiv [\![\Theta]\!]$.

Proof. For every base step in the definition of $\longrightarrow_{gr}$ or $\simeq_i$, we show that $[\![-]\!]^L$ maps $\simeq$ flag deductions to $\equiv$ λ-terms. Similarly for $\longrightarrow_s$. □

Lemma 4.6 (Closure under $\longrightarrow_r$, $\longrightarrow_g$, $\longrightarrow_s$). *If* Σ *is a flag deduction and* $\Sigma \longrightarrow_g \Sigma'$ *or* $\Sigma \longrightarrow_r \Sigma'$ *or* $\Sigma \longrightarrow_g \Sigma'$, *then* Σ' *is a flag deduction with the same conclusion.*

Proof. For every reduction step of Definition 4.3, we easily verify the statement of the Lemma. □

Lemma 4.7. *The reductions* $\longrightarrow_g$, $\longrightarrow_r$ *and* $\longrightarrow_{gr}$ *(the union of* $\longrightarrow_g$ *and* $\longrightarrow_r$*) are strongly normalizing.*

Proof. The $\longrightarrow_r$ and $\longrightarrow_g$ rules return a shorter flag deduction. □

It can also be proved that $\longrightarrow_{gr}$ is confluent. We will obtain this property only as a consequence of uniqueness of normal forms, which we will prove by defining the *gr*-normal form of a flag deduction directly (without reducing).
The end goal of this section is to prove the reverse of Theorem 4.5. To do that we consider the $\longrightarrow_{grs}$ normal forms, so we first have to show that these exist, which we do by proving that $\longrightarrow_s$ terminates on $\longrightarrow_{gr}$-normal forms. We first introduce some useful notions and state some auxiliary Lemmas.
For l a line in Σ, we want to define $\Sigma \lceil l$ as the flag deduction that arises from Σ by restricting to all those lines that are 'needed' by l. This is the transitive closure of the 'refers to' relation, where a line l refers to l' if l' is used in the motivation (the m) of line l. In taking this transitive closure we skip the applications of the repeat rule. The set of needed lines is inductively defined as follows.

Definition 4.8. *Given a flag deduction* Σ *and a line* $\langle l, A, m\rangle$ *in* Σ, *we define the set of* needed lines *for* $\langle l, A, m\rangle$, $lines_\Sigma(\langle l, A, m\rangle)$, *by*

$$\begin{aligned} lines_\Sigma(\langle l, A, -\rangle) &= \{l\} \\ lines_\Sigma(\langle l, A, [R, l_0]\rangle) &= lines_\Sigma(l_0) \\ lines_\Sigma(\langle l, A{\rightarrow}B, [{\rightarrow}\text{-}I, l_1, l_2]\rangle) &= \{l, l_1\} \cup lines_\Sigma(l_2) \\ lines_\Sigma(\langle l, B, [{\rightarrow}\text{-}E, l_1, l_2]\rangle) &= \{l\} \cup lines_\Sigma(l_1) \cup lines_\Sigma(l_2) \end{aligned}$$

We will usually omit the formula A *and the motivation* m *from this notation, writing* $lines_\Sigma(l)$ *instead. We write* $\Sigma \lceil l$ *for* $\Sigma \lceil lines_\Sigma(l)$, *the restriction of* Σ *to the lines that are in* $lines_\Sigma(l)$.

Lemma 4.9. *If* Σ *is in gr-normal form, then* $\Sigma \lceil l = \Sigma$ *(with* l *the last line of* Σ*).*

Proof. Obviously, $l' \in \Sigma \lceil l \Rightarrow l' \in \Sigma$, so we only have to prove that all lines that occur in Σ also appear in $\Sigma \lceil l$. This is an easy consequence of the Definition of needed lines. □

Lemma 4.10. *The set of 'needed lines' of a flag deduction is preserved under gr-reduction. That is, for* $\Sigma \longrightarrow_{gr} \Sigma'$, *with last lines* l *and* l' *respectively,*

$$lines_\Sigma(l) = lines_{\Sigma'}(l').$$

Which implies immediately that $\Sigma \lceil l = \Sigma' \lceil l'$.

Proof. By distinguishing cases according to the reduction step $\Sigma \longrightarrow_{gr} \Sigma'$.

Corollary 4.11. *For Σ a flag deduction with last line l,*

$$gr\text{-}normal\ form(\Sigma) = \Sigma \restriction l.$$

Proof. Σ has a gr-normal form, say $\Sigma \longrightarrow_{gr} \Sigma_1 \longrightarrow_{gr} \ldots \longrightarrow_{gr} \Sigma_n$ with Σ_n in gr-normal form. From Lemma 4.10 it follows that $\Sigma \restriction l = \ldots = \Sigma_i \restriction l_i = \ldots = \Sigma_n \restriction l_n$ where l_i is the last line of Σ_i. From Lemma 4.9, it follows that $\Sigma_n \restriction l_n = \Sigma_n$, so $\Sigma \restriction l$ is Σ_n, the gr-normal form of Σ.

Lemma 4.12. *If Σ in gr-normal form, then $lab(\Sigma) = lab([\![\Sigma]\!]^L)$.*

Proof. For $\mathrm{lab}(\Sigma) \subseteq \mathrm{lab}([\![\Sigma]\!]^L)$, we prove that all labels of $\Sigma \restriction l$ (l the last line of Σ) occur in $[\![\Sigma]\!]^L$. (Then we are done, because, by Lemma 4.9, $\Sigma \restriction l = \Sigma$ for Σ in gr-normal form.) Let $l' \in \Sigma \restriction l$, i.e. $l' \in \mathrm{lines}_\Sigma(l)$. We now prove $l' \in \mathrm{lab}([\![\Sigma]\!]^L)$ by an immediate induction on $\mathrm{lines}_\Sigma(l)$. For the reverse, $\mathrm{lab}([\![\Sigma]\!]^L) \subseteq \mathrm{lab}(\Sigma)$, we prove $\mathrm{lab}([\![\Sigma]\!]^L) \subseteq \mathrm{lab}(\Sigma \restriction l)$ by induction on $[\![\Sigma]\!]^L$.

Lemma 4.13. *If Σ in gr-normal form, and $\Sigma \longrightarrow_s \Sigma'$, then Σ' is also in gr-normal form.*

Proof. Let Σ be in gr-normal form with last line l, so $l' \in \mathrm{lines}_\Sigma(l)$ for all $l' \in \mathrm{lab}(\Sigma)$. In $\Sigma \longrightarrow_s \Sigma'$ new lines are introduced, but they are also in $\mathrm{lines}_\Sigma(l)$, as is easily checked by analyzing the two possible s-reduction steps. □

Lemma 4.14. *If Σ in gr-normal form and $\Sigma \longrightarrow_s \Sigma'$, then $\#lab([\![\Sigma]\!]^L) < \#lab([\![\Sigma']\!]^L)$*

Proof. In an s-reduction step, new labels are added. Now the Lemma follows immediately from Lemmas 4.12 and 4.13. □

Corollary 4.15 (Termination of $\longrightarrow_s$ on gr-normal forms). *If Σ is in gr-normal form, then unsharing ($\longrightarrow_s$) terminates on Σ.*

Proof. The number of labels in $[\![\Sigma]\!]^L$ strictly increases under $\longrightarrow_s$ (Lemmas 4.14 and 4.13). On the other hand, the λ-term $[\![\Sigma]\!]$ (without labels) does not change under $\longrightarrow_s$. There is a maximum to the number of labels that can occur in a labeled version of $[\![\Sigma]\!]$, so $\longrightarrow_s$ terminates. □

We now study the i-equality on grs-normal forms. The main result is that, for Σ and Θ in grs-normal form, if $[\![\Sigma]\!]^L = [\![\Theta]\!]^L$, then $\Sigma \simeq_i \Theta$. The main technique for establishing this result is 'merging' two flag deductions into one. The main property about merging is that, if Σ is a grs-normal form containing the two *independent* lines l_1 and l_2, then merging $\Sigma \restriction l_1$ and $\Sigma \restriction l_2$ yields a well-formed flag-deduction Σ'. This only works for Σ in grs-normal form.

Definition 4.16. *Given the flag deduction Σ, with $l_1 l_2 \in lab(\Sigma)$, l_1 and l_2 are* Σ-independent*, notation $l_1 \perp_\Sigma l_2$, if $l_1 \notin lines_\Sigma(l_2)$ and $l_2 \notin lines_\Sigma(l_1)$. We omit Σ when it is clear from the context.*

An important property of $\perp$ is the following.

Lemma 4.17. *If Σ is in s-nf and contains the line $\langle l, A, [\rightarrow\text{-}E, l_1, l_2]\rangle$, then $l_1 \perp l_2$.*

In the following we denote by $\mathrm{oflag}(\Sigma)$ the set of *open flags* of Σ.

Definition 4.18. *The flag deduction Σ and Θ are* compatible*, notation $comp(\Sigma, \Theta)$, if the following hold.*

1. *$lab(\Sigma) \cap lab(\Theta) \subset oflag(\Sigma) \cap oflag(\Theta)$, i.e. a label that occurs in both Σ and Θ must occur as an open flag in both.*
2. *If $i \in oflag(\Sigma) \cap oflag(\Theta)$ then it occurs as* the same flag *in both Σ and Θ (i,e. with the same formula).*

To define the merging of two flag deductions Σ and Θ, we view them as sequences. The goal is to prove that if both Σ and Θ are in *grs*-normal form, then the merging is a well-formed flag deduction. But this is only the case if we treat a part of a deduction that is 'under a flag' as a 'block' (one part of the sequence Σ), thus disallowing lines from Θ to be moved under a flag of Θ.

Definition 4.19. *If $comp(\Sigma, \Theta)$, we define the* merging *of Σ and Θ, notation $\Sigma||\Theta$, as the following flag deduction.*

1. *First remove $oflag(\Sigma)$ from Σ and $oflag(\Theta)$ from Θ, obtaining Σ', resp. Θ'.*
2. *Now interleave the sequences Σ' and Θ', starting at the end with an element of Σ'. In doing so, we consider a part $\langle i, A, F\rangle, \ldots, \langle l, A{\to}B, [I, i, l_2]\rangle$ as one element of the sequence.*
3. *Finally, put all elements of $oflag(\Sigma) \cup oflag(\Theta)$ on top of the sequence Δ in a canonical way (following a fixed ordering of $\mathcal{I}$).*

Lemma 4.20. *If $comp(\Sigma, \Theta)$ then $\Sigma||\Theta$ is a well-formed flag deduction.*

Lemma 4.21. *If $comp(\Sigma, \Theta)$ and $\Sigma \simeq_i \Sigma'$, then $\Sigma||\Theta \simeq_i \Sigma'||\Theta$ and $\Theta||\Sigma \simeq_i \Theta||\Sigma'$.*

Proposition 4.22. *Given two flag deductions Σ and Θ in grs-normal form, if $[\![\Sigma]\!]^L \equiv [\![\Theta]\!]^L$, then $\Sigma \simeq_i \Theta$.*

Proof. By induction on the structure of $[\![\Sigma]\!]^L$.

var $[\![\Sigma]\!]^L = x_i$. Then Σ and Θ are both $\langle i, A, [F]\rangle$.

app $[\![\Sigma]\!]^L = (M^{l_1} N^{l_2})^l$. Then Σ and Θ end with an $\to$-E rule. By induction hypothesis, $\Sigma \lceil l_1 \simeq_i \Theta \lceil l_1$ and $\Sigma \lceil l_2 \simeq_i \Theta \lceil l_2$ and so $\Sigma \simeq_i \Sigma \simeq_i ((\Sigma \lceil l_1)||(\Sigma \lceil l_2))\langle l, B, [E, l_1, l_2]\rangle \simeq_i ((\Theta \lceil l_1)||(\Theta \lceil l_2))\langle l, B, [E, l_1, l_2]\rangle \simeq_i \Theta$

abs $[\![\Sigma]\!]^L = (\lambda x_i{:}A.M^{l_2})^l$. Then Σ and Θ end with an $\to$-I rule. They can be of one of the following shapes.

$$\begin{array}{l|l|ll} \vdots & \ldots & & \\ i & \boxed{\begin{array}{l} A \\ \hline \ldots \\ B \end{array}} & & \\ \vdots & & & \\ l_2 & & & \\ l & A{\to}B & \to\text{I}, i, l_2 & \end{array} \qquad \text{or} \qquad \begin{array}{l|l|ll} \vdots & \ldots & & \\ l_2 & B & & \\ i & \boxed{\begin{array}{l} A \end{array}} & & \\ l & A{\to}B & \to\text{I}, i, l_2 & \end{array}$$

We distinguish cases according to whether $x_i \in \mathrm{FV}(M)$ (and then $i \in \mathrm{lines}(l_2)$) or $x_i \notin \mathrm{FV}(M)$ (and then $i \notin \mathrm{lines}(l_2)$). In the first case, Σ and Θ must have the first shape. By induction hypothesis, $\Sigma \lceil l_2 \simeq_i \Theta \lceil l_2$. Then also $\Sigma \lceil l_2 \simeq_i \Theta \lceil l_2$, where the last open flag is preserved and hence $\Sigma \simeq_i \Theta$. In the second case, $\Sigma \lceil l_2$ and $\Theta \lceil l_2$ do not contain i. Hence both Σ and Θ are $\simeq_i$ equal to a deduction of the second shape. By induction hypothesis, $\Sigma \lceil l_2 \simeq_i \Theta \lceil l_2$ and we can safely add the line $\langle i, A, F\rangle$ at the end and also the line $\langle l, A{\to}B, [I, i, l_2]\rangle$, and hence $\Sigma \simeq_i \Theta$. □

To prove the final theorem, we need to more Lemmas. They could have been proved before already, but were not yet needed. Therefore we state them only now. Both are proved by induction on the structure of Σ, using the fact that if l is the label of a line that is not a flag, then l occurs at most once in a motivation of Σ.

Lemma 4.23. *If Σ is a flag deduction in s-normal form, then $[\![\Sigma]\!]^L$ is a uniquely labelled simply typed term. (That is, every label occurs at most once in $[\![\Sigma]\!]^L$.)*

Lemma 4.24. *For r a relabelling, Σ a flag deduction and M a labelled simply typed term, if $[\![\Sigma]\!]^L = M$, then $[\![r(\Sigma)]\!]^L = r(M)$.*

Theorem 4.25. *If $[\![\Sigma]\!] \equiv [\![\Theta]\!]$, then $\Sigma \simeq_{gris} \Theta$.*

Proof. Suppose $[\![\Sigma]\!] \equiv [\![\Theta]\!]$. Consider the *grs*-normal forms of Σ and Θ: Σ' and Θ'. Then $[\![\Sigma]\!] \equiv [\![\Sigma']\!] \equiv [\![\Theta']\!] \equiv [\![\Theta]\!]$. This implies that $[\![\Sigma']\!]^L \equiv M$, $[\![\Theta']\!] \equiv N$ with $|M| \equiv |N|$. Moreover (by Lemma 4.23), all labels in M and N are unique, so we can find a relabelling r such that $r(M)$ is N. By Lemma 4.24, this means that $[\![r(\Sigma')]\!]^L \equiv r(M) \equiv N \equiv [\![\Theta']\!]^L$. From Proposition 4.22, it now follows that $r(\Sigma') \simeq_i \Theta'$. Hence (as we work modulo relabelling), $\Sigma \simeq_{grs} \Sigma' \simeq_i \Theta' \simeq_{grs} \Theta$. □

The following follows immediately from the Theorem and Theorem 3.9.

Corollary 4.26. *Given a flag deduction Σ, $(\!|[\![\Sigma]\!]|\!) \simeq_{gris} \Sigma$.*

4.1 Defining Cut-elimination

We can now define cut-elimination on flag deductions by first taking the $\longrightarrow_s$ normal form and then eliminating cuts as discussed in Section 2.1.

Definition 4.27. *We define cut-elimination on flag deductions as follows.*

$$
\begin{array}{l|l|l}
1 & \boxed{\begin{array}{l} A \\ \hline \Sigma \end{array}} & \\
\vdots & & \\
l & A{\to}B & {\to}\text{-}I\ 1{,}n \\
\vdots & \Theta & \\
k & A & \\
\vdots & \Pi & \\
l' & B & {\to}E,\ l,\ k
\end{array}
\quad \Longrightarrow_c \quad
\begin{array}{l|l|l}
\vdots & & \\
k & \Theta & \\
\vdots & A & \\
\vdots & \Sigma[k/1] & \\
\vdots & \Pi & \\
l' & B & R{,}n
\end{array}
$$

$$
\begin{array}{l|l|l}
k & A & \\
\vdots & \Pi_0 & \\
k' & \boxed{\begin{array}{l} A \\ \hline \Sigma \end{array}} & \\
\vdots & & \\
l & A{\to}B & {\to}\text{-}I\ k\text{'}{,}n \\
\vdots & \Pi_1 & \\
l' & B & {\to}E,\ l,\ k
\end{array}
\quad \Longrightarrow_c \quad
\begin{array}{l|l|l}
k & A & \\
\vdots & \Pi_0 & \\
\vdots & \Sigma[k/k'] & \\
\vdots & \Pi_1 & \\
l' & B & R{,}n
\end{array}
$$

As usual, these reduction rules can also be applied in a context. In the definition, we introduce repeat rules to make sure that B remains on the last line. These can again be removed via $\longrightarrow_r$ steps.

Remark 4.28. Different from cut-elimination in Gentzen natural deduction, a $\Longrightarrow_c$-step does not involve any duplication of subderivations, which may seem odd. However, a $\Longrightarrow_c$ step can introduce sharing of subproofs and the unsharing (via $\longrightarrow_s$) involves duplication of subderivations. This also implies that, to apply another cut-elimination step we first have to take the $\longrightarrow_s$-normal form of the result.

An example where a $\Longrightarrow_c$ step creates sharing is the following. (On the right hand side, line l is shared by lines 4 and 6.)

$$
\begin{array}{l|l|l}
1 & A{\to}(B{\to}C){\to}D & \\
2 & A{\to}C & \\
3 & \boxed{\begin{array}{l} A \\ \hline (B{\to}C){\to}D \end{array}} & {\to}\text{E, 1, 3} \\
4 & \boxed{\begin{array}{l} B \\ \hline C \end{array}} & {\to}\text{E, 2, 3} \\
5 & B{\to}C & {\to}\text{I, 5, 6} \\
6 & D & {\to}\text{E, 4, 7} \\
7 & A{\to}D & {\to}\text{I, 3, 8} \\
\vdots & \Theta & \\
l & A & \\
l+1 & D & {\to}\text{E, 9, } l
\end{array}
\quad \Longrightarrow_c \quad
\begin{array}{l|l|l}
1 & A{\to}(B{\to}C){\to}D & \\
2 & A{\to}C & \\
\vdots & \Theta & \\
l & A & \\
4 & (B{\to}C){\to}D & {\to}\text{E, 1, } l \\
5 & \boxed{\begin{array}{l} B \\ \hline C \end{array}} & {\to}\text{E, 2, } l \\
6 & B{\to}C & {\to}\text{I, 5, 6} \\
7 & D & {\to}\text{E, 4, 7} \\
8 & D & \text{R, 8}
\end{array}
$$

Lemma 4.29. *If Σ is a well-formed flag deduction in $\longrightarrow_s$-normal form and $\Sigma \Longrightarrow_c \Sigma'$, then Σ' is a well-formed flag deduction with the same conclusion.*

Proof. As all the lines (except for the flags) are referred to at most once in a $\longrightarrow_s$-normal form,we can safely move around the subparts (and remove some of them) as indicated above.

Theorem 4.30. *For Σ a well-formed flag deduction in $\longrightarrow_{gs}$-normal form and for M a uniquely labelled simply typed term,*

$$\Sigma \Longrightarrow_c \Sigma_1 \longrightarrow_{sr} \Sigma_2 \simeq_i \Theta \quad \textit{iff} \quad [\![\Sigma]\!] \longrightarrow_\beta [\![\Theta]\!]$$
$$M \longrightarrow_\beta N \quad \textit{iff} \quad (\![M]\!) \Longrightarrow_c \Sigma_1 \longrightarrow_{sr} \Sigma_2 \simeq_i (\![N]\!)$$

where the Σ_1 and Σ_2 are existentially quantified.

5 Future work

In this paper we restrict to the simplest fragment of logic: minimal proposition logic. But already there important aspects of flag deductions become visible, showing that in some way their structure (e.g. the order of the steps) is quite arbitrary but that in another way (e.g. the reusability of proven results, 'sharing'), their structure is quite useful and interesting. The Curry-Howard interpretation to simply typed λ calculus that we define here and the analysis of cut-elimination brings about this structure quite nicely.
We believe that the results of this paper can be extended to full first order predicate logic. This will be presented in forthcoming work which is a more detailed exposition of the results in this paper. A more intersting aspect is the definition of a term calculus for flag deductions directly. Simply typed λ-calculus ignores part of the structure of a flag deduction, extracting its 'computational content' and removing 'bureaucratic details'. But it also removes sharing and we don't consider that to be only a 'bureaucratic detail', but sometimes computationally relevant. It was suggested by the referees to use a λ-calculus with let-expressions to encode flag deductions faithfully. Then the reductions $\longrightarrow_{grs}$ and the congruence $\simeq_i$ can be described on these terms directly, giving a more perspicuous presentation. The 'sharing' example deduction in Section 4 then is interpreted as

$$\text{let } x_5 = (\lambda x_3.\text{let } x_4 = x_1 \text{ in } x_4) \text{ in } (\text{let } x_6 = x_2 x_5 \text{ in } (\text{let } x_7 = x_6 x_5 \text{ in } x_7))$$

This gives connections with the monadic presentation of λ-calculus, the (operational) CPS-translation, the (logical) A-translation.
Similarly, one can define a slightly different Curry-Howard embedding to simply typed terms and then $\simeq_i$ becomes σ-equivalence on λ-terms, as in the work of [Regnier 1994]. This gives a connection with proof nets. We will exploit these connections further and we thank the referees for their comments.
We note that the other suggested interpretations do not really follow the inductive structure of the flag deductions. It might be interesting to find a term-calculus for flag deductions where the basic constructors for flag deductions are the same as for the term calculus.

Acknowledgments

We greatly acknowledge the very insightful comments and suggestions of the referees on the first version of this paper. We are sorry that we don't have the space to discuss all their comments here.

References

[Barendregt 1992] H.P. Barendregt, Lambda calculi with Types. In *Handbook of Logic in Computer Science*, eds. Abramski et al., Oxford Univ. Press, pp. 117 – 309.

[Bornat and Sufrin 1999] R. Bornat and B. Sufrin, Animating Formal Proof at the Surface: The Jape Proof Calculator; *The Computer Journal*, Vol. 42, no. 3, pp. 177-192, 1999.

[Regnier 1994] L. Regnier, Une équivalence sur les lambda-termes, *TCS* 126(2), pp. 281–292, 1994.

[Fitch 1952] F. B. Fitch, *Symbolic Logic*, the Ronald Press Company, New York, 1952.

[Gentzen 1969] G. Gentzen. *Collected Works*. Edited by M.E. Szabo. North-Holland, Amsterdam, 1969.

[Howard 1980] W.H. Howard, The formulas-as-types notion of construction, in *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, eds. J.P. Seldin and J.R. Hindley, Academic Press 1980, pp. 479–490.

[Prawitz 1965] D. Prawitz. *Natural Deduction*. Almquist & Wiksell, Stockholm, 1965.