Addendum to “Proof terms for generalized natural deduction”

Herman Geuvers
Radboud University Nijmegen & Technical University Eindhoven, The Netherlands
herman@cs.ru.nl

Tonny Hurkens

hurkens@science.ru.nl

Abstract

This short note is a clarification of a proof in the paper “Proof terms for generalized natural deduction” [2]. In the original paper, some details are missing, which makes the proof unclear. In particular, this concerns the proof of Strong Normalization for the reduction $\longrightarrow_a$, the proof-reduction that contracts an introduction which is immediately followed by an elimination of the same connective. This is also called the β-rule for the connective. In [2], this is proved for generalized intuitionistic connectives, which are derived from the truth-table definition of the connective. In this note, we provide some additional details for the proof and we repair a few omissions in the definitions. We do not repeat the definitions of the derivation rules and of the reduction $\longrightarrow_a$, so this note can only be read along with the original paper [2].

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 constructive logic, natural deduction, detour conversion, permutation conversion, normalization Curry-Howard isomorphism

Digital Object Identifier 10.4230/LIPIcs...

Acknowledgements We thank Andreas Abel for pointing out some unclarities in the original paper [2], that has led us to write this addendum.

In [1] it has been shown how to generate natural deduction rules for propositional connectives from truth tables, both for classical and constructive logic. The paper [2] extends this for the constructive case with proof-terms, thereby extending the Curry-Howard isomorphism to these new connectives. A general notion of conversion of proofs is defined, both as a conversion of derivations and as a reduction of proof-terms. Conversions come in two favors: either a *detour conversion*, $\longrightarrow_a$, arising from a *detour convertibility*, where an introduction rule is immediately followed by an elimination rule, or a *permutation conversion*, $\longrightarrow_b$, arising from an *permutation convertibility*, an elimination rule nested inside another elimination rule. In the paper [2], both are defined for the general setting, as conversions of derivations and as reductions of proof-terms. One of the main contributions of [2] is that detour conversion, $\longrightarrow_a$, is strongly normalizing. Other results are that permutation conversion, $\longrightarrow_b$, is strongly normalizing and that the combination of $\longrightarrow_a$ and $\longrightarrow_b$ is weakly normalizing. In [3], it is proven that the combination of $\longrightarrow_a$ and $\longrightarrow_b$ is strongly normalizing.

Definition 57 in Section 6.1 defines *saturated sets*, which are sets of strongly normalizing terms that are closed under *key-redex expansion* and it defines, given a connective c of arity n and saturated sets $X_1, \ldots, X_n$, a set $c(X_1, \ldots, X_n)$ (which is then shown to be saturated

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

as well).

Definition 57 (3) should be read as follows:
A set $X \subseteq \mathsf{Term}$ is *saturated* ($X \in \mathsf{SAT}$) if it satisfies the following properties

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

This is because otherwise each strongly normalizing t that doesn't have a key-redex would be in X.

Definition 57 (4) should be read as follows:
For a connective c of arity n and $X_1, \ldots, X_n \in \mathsf{SAT}$ we define the set $c(X_1, \ldots, X_n)$ as follows. Assume that $r_1, \ldots, r_m$ are the elimination rules for c.

$$\begin{array}{rl} c(X_1, \ldots, X_n) := \{t \mid \underline{t \in \mathsf{SN}} \wedge \forall r_i \in \{r_1, \ldots, r_m\} & \\ \forall D \in \mathsf{SAT}, \forall \overline{p}, \overline{q} \in \mathsf{Term} & \\ \forall k(p_k \in X_k) \wedge (\forall \ell \forall u_\ell \in X_\ell \, (q_\ell[y_\ell := u_\ell] \in D)) \implies & t \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}] \in D \ \} \end{array}$$

This is to make sure that the definition is also correct for a connective that has no elimination rules, like $\top$. In that case $c(X_1, \ldots, X_n) = \mathsf{SN}$.

Now, we re-check the main lemmas concerning these definitions: Lemma 58 and Lemma 61 of [2]. To clarify the proofs we have isolated two additional properties about key-redexes in Lemma 2.

▶ **Lemma 1** (Lemma 58 of [2].). *If* $X_1, \ldots, X_n \in \mathsf{SAT}$*, then* $c(X_1, \ldots, X_n) \in \mathsf{SAT}$.

Proof. We check the 3 conditions of “saturated set” for $c(X_1, \ldots, X_n)$. The proof of the first condition is now trivial and that of the second one largely the same as in [2]; only the third part is interesting. Suppose $X_1, \ldots, X_n \in \mathsf{SAT}$.

c. Suppose $t \in \mathsf{SN}$ and t has a key-redex and $\forall t_0(t \longrightarrow_a^k t_0 \Rightarrow t_0 \in c(X_1, \ldots, X_n))$ (*). Let r_i be a rule for c and let $D \in \mathsf{SAT}$, $\overline{p}, \overline{q} \in \mathsf{Term}$ with $\forall k(p_k \in X_k)$ and $\forall \ell \forall u_\ell \in X_\ell \, (q_\ell[y_\ell := u_\ell] \in D)$. We need to prove that $t \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}] \in D$.
By Lemma 2(1) (see below) we know that all key-reduction steps from $t \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}]$ are of the form
$$t \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}] \longrightarrow_a^k t' \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}]$$
with $t \longrightarrow_a^k t'$ (for some t'). We know $t' \in c(X_1, \ldots, X_n)$, so $t' \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}] \in D$. So, we have $\forall u(t \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}] \longrightarrow_a^k u \implies u \in D)$. Also $t \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}]$ has a key-redex and $t \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}] \in \mathsf{SN}$ (by Lemma 2(3) below). So $t \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}] \in D$ and we are done.

◀

▶ **Lemma 2.** **1.** *If t has a key-redex and* $t \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}] \longrightarrow_a^k u$*, then* $u = t' \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}]$ *for some t' with* $t \longrightarrow_a^k t'$.
2. *If t has a key-redex and* $t \longrightarrow_a t' \longrightarrow_a^k q'$*, where the reduction* $t \longrightarrow_a t'$ *is not a key-reduction, then there is a q with* $t \longrightarrow_a^k q \twoheadrightarrow_a q'$.
3. *If all proper sub-terms of t are SN and* $\forall q(t \longrightarrow_a^k q \implies q \in \mathsf{SN})$*, then* $t \in \mathsf{SN}$

Proof. The first is simply by an analysis of the possible cases for $t \cdot_{r_i} [\overline{p} \,;\, \overline{\lambda y.q}] \longrightarrow_a^k u$. The second is by induction on the shape of t. The third is by proving $\forall t'(t \longrightarrow_a t' \implies t' \in \mathsf{SN})$, using an analysis of the possible cases for the structure of t and induction on the proof that the direct subterms of t are SN, using (2). ◀

For completeness, we also check Lemma 61 of [2], in particular the “introduction case”.

▶ **Lemma 3** (Lemma 61 of [2].). *If* $\Gamma \vdash t : A$, *and* $\rho \models \Gamma$, *then* $\langle t \rangle_\rho \in \langle A \rangle$.

Proof. By induction on the derivation of $\Gamma \vdash t : A$. Suppose $\rho \models \Gamma$. The (axiom) case and the (el) case are exactly as in [2], so we only consider the (in) case. We ignore ρ for the rest of the proof, as it gives a lot of notational overhead, so we just write t for $\langle t \rangle_\rho$.

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

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

  We need to prove $\{\overline{s}\, ;\, \overline{\lambda x.t}\}_r \in \Phi$ and we have as induction hypothesis $s_j \in A_j$ (for all j) and $t_i[x_i := a_i] \in \Phi$ for all t_i and $a_i \in A_i$. In particular, all these terms are SN. In case there are no elimination rules for Φ, the interpretation of Φ is SN and indeed, $\{\overline{s}\, ;\, \overline{\lambda x.t}\}_r \in \mathsf{SN}$, so we are done.

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

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

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

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

◀

References

1. H. Geuvers and T. Hurkens. Deriving natural deduction rules from truth tables. In *ICLA*, volume 10119 of *Lecture Notes in Computer Science*, pages 123–138. Springer, 2017.
2. Herman Geuvers and Tonny 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 für Informatik, 2017. URL: `https://doi.org/10.4230/LIPIcs.TYPES.2017.3`, `doi:10.4230/LIPIcs.TYPES.2017.3`.
3. Herman Geuvers, Iris van der Giessen, and Tonny Hurkens. Strong normalization for truth table natural deduction. *Fundam. Informaticae*, 170(1-3):139–176, 2019. URL: `https://doi.org/10.3233/FI-2019-1858`, `doi:10.3233/FI-2019-1858`.