On Transforming Intuitionistic Matrix Proofs into Standard-Sequent Proofs

Stephan Schmitt Christoph Kreitz

Fachgebiet Intellektik, Fachbereich Informatik,
Technische Hochschule Darmstadt
Alexanderstraße 10, D–64283 Darmstadt, Germany
e-mail: {steph,kreitz}@intellektik.informatik.th-darmstadt.de

Abstract. We present a procedure transforming intuitionistic matrix proofs into proofs within the intuitionistic standard sequent calculus. The transformation is based on L. Wallen's proof justifying his matrix characterization for the validity of intuitionistic formulae. Since this proof makes use of Fitting's *non-standard* sequent calculus our procedure consists of two steps. First a non-standard sequent proof will be extracted from a given matrix proof. Secondly we transform each non-standard proof into a standard proof in a structure preserving way. To simplify the latter step we introduce an extended standard calculus which is shown to be sound and complete.

1 Introduction

According to the *proofs-as-programs* paradigm theorems proven in a constructive manner can be interpreted as specifications of programs which are contained in the proof. Therefore proof tools for constructive logics are very important for the development of verifiably correct software. Because of the expressiveness of the underlying calculus these tools are essentially interactive proof editors supported by a *tactic* mechanism for programming proofs on the meta-level. On the other hand theorem provers like Setheo [9], Otter [16], or KoMeT [3] show that reasoning about *classical* predicate logic can be automated sufficiently well. It would therefore be desirable to have a procedure which automatically generates the purely logical parts of a proof during a session with a proof editor for a rich *constructive* theory. This would liberate its user from having to deal with rather tedious but boring parts of the proof. The proof created by such a procedure should be expressed within the calculus underlying the proof development tool to allow the extraction of programs.

Proof editors like the NuPRL system [4] are based on a sequent calculus supporting the construction of proofs which are comprehensible for mathematicians and programmers. It includes a calculus for predicate logic similar to Gentzen's calculus for intuitionistic logic [8]. This calculus, which contains at most one formula in the succedent of a sequent, will be considered a *standard* sequent calculus $\mathcal{LJ}_S$.

P. Baumgartner, R. Hähnle & J. Posegga, ed., 4^{th} *International Workshop on Theorem Proving with Analytic Tableaux and Related Methods (TABLEAUX 95)*, LNAI 918, pp. 106–121,

In [15] L. Wallen successfully created a matrix characterization $\mathcal{MJ}$ for the validity of intuitionistic formulae. His theoretical framework is based on Fitting's [5] *non-standard* sequent calculus $\mathcal{LJ}_{NS}$ which allows the occurrence of more than one succedent formula. Because of this characterization it is possible to construct the purely logical parts of a NuPRL-proof in two steps. First a matrix proof in $\mathcal{MJ}$ has to be found by some effective proof procedure. Wallen suggested extending Bibel's connection method [1, 2] for this purpose. Secondly the matrix proof has to be transformed back into a valid standard sequent proof.

In this paper we shall focus on the second step, i.e. on a procedure transforming a proof which was derived efficiently in $\mathcal{MJ}$ into a proof within the standard sequent calculus $\mathcal{LJ}_S$. Because of Wallen's investigations we can be sure that such a $\mathcal{LJ}_S$–proof must exist but there is not yet an efficient method to construct it from a a given $\mathcal{MJ}$–proof. In order to do this we again proceed in two steps. First we represent the $\mathcal{MJ}$–proof in the non-standard calculus $\mathcal{LJ}_{NS}$. Secondly we convert the resulting non-standard sequent proof into a standard proof. We will show, however, that because of the strong differences between the rules of the two calculi it is not possible to transform every $\mathcal{LJ}_{NS}$–proof into a corresponding $\mathcal{LJ}_S$–proof without changing the proof structure. To solve this problem we have developed an extended standard calculus $\mathcal{LJ}_S^\star$ which is able to represent each $\mathcal{LJ}_{NS}$–proof in a structure preserving way. We have proven $\mathcal{LJ}_S^\star$ to be sound and complete and implemented its rules as tactics of the NuPRL system. Therefore we can transform intuitionistic matrix proofs into extended NuPRL proofs without any additional search.

In the following section we shall briefly review the sequent calculi $\mathcal{LJ}_S$ and $\mathcal{LJ}_{NS}$ and summarize the notation which is necessary to understand Wallen's matrix characterization $\mathcal{MJ}$. Section 3 will discuss the procedure transforming matrix proofs into non-standard sequent proofs. In section 4 we shall present an $\mathcal{LJ}_{NS}$–proof which cannot be converted into an equivalent $\mathcal{LJ}_S$–proof in a structure preserving way and introduce the extended standard calculus $\mathcal{LJ}_S^\star$. Section 5 will present the transformation from $\mathcal{LJ}_{NS}$ into $\mathcal{LJ}_S$. We conclude with a few remarks on implementation issues and efficient search procedures for $\mathcal{MJ}$–proofs.

2 Preliminaries

Our transformation procedure relates intuitionistic proofs in three entirely different calculi: a matrix characterization $\mathcal{MJ}$ [15], a non-standard sequent calculus $\mathcal{LJ}_{NS}$ [5], and the standard sequent calculus $\mathcal{LJ}_S$ [8]. In this section we shall briefly review these calculi.

Wallen's basic idea for developing the matrix characterization $\mathcal{MJ}$ which we shall describe below was to use Schütte's embedding of intuitionistic logic $\mathcal{J}$ into the modal logic $S4$ [12] together with the Kripke semantic similarity between $S4$ and $\mathcal{J}$ [13]. Therefore his investigations were based on an intuitionistic sequent calculus which has a structure similar to the one for $S4$. Contrary to Gentzen's (standard) sequent calculus $\mathcal{LJ}_S$ this calculus allows more than one formula in

$\mathcal{LJ}_{NS}$:

axiom A: $$\frac{}{\Gamma, A \vdash A, \Delta}$$

$\vee$*–intro:* $$\frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}$$

$\wedge$*–intro:* $$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}$$

$\neg$*–intro:* $$\frac{\Gamma, A \vdash}{\Gamma \vdash \neg A, \Delta}$$

$\rightarrow$*–intro:* $$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B, \Delta}$$

$\forall$*–intro a:* $$\frac{\Gamma \vdash A[x\backslash a]}{\Gamma \vdash \forall x.\, A, \Delta} \quad (a \text{ Eigenvar.})$$

$\exists$*–intro t:* $$\frac{\Gamma \vdash A[x\backslash t], \exists x.\, A, \Delta}{\Gamma \vdash \exists x.\, A, \Delta}$$

$\vee$*–elim:* $$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}$$

$\wedge$*–elim:* $$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}$$

$\neg$*–elim:* $$\frac{\Gamma, \neg A \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}$$

$\rightarrow$*–elim:* $$\frac{\Gamma, A \rightarrow B \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \rightarrow B \vdash \Delta}$$

$\forall$*–elim t:* $$\frac{\Gamma, \forall x.\, A, A[x\backslash t] \vdash \Delta}{\Gamma, \forall x.\, A \vdash \Delta}$$

$\exists$*–elim a:* $$\frac{\Gamma, A[x\backslash a] \vdash \Delta}{\Gamma, \exists x.\, A \vdash \Delta} \quad (a \text{ Eigenvar.})$$

$\mathcal{LJ}_{S}$:

axiom A: $$\frac{}{\Gamma, A \vdash A}$$

$\vee$*–intro 1:* $$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}$$

$\vee$*–intro 2:* $$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$$

$\wedge$*–intro:* $$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

$\neg$*–intro:* $$\frac{\Gamma, A \vdash}{\Gamma \vdash \neg A}$$

$\rightarrow$*–intro:* $$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}$$

$\forall$*–intro a:* $$\frac{\Gamma \vdash A[x\backslash a]}{\Gamma \vdash \forall x.\, A} \quad (a \text{ Eigenvar.})$$

$\exists$*–intro t:* $$\frac{\Gamma \vdash A[x\backslash t]}{\Gamma \vdash \exists x.\, A}$$

$\vee$*–elim:* $$\frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C}$$

$\wedge$*–elim:* $$\frac{\Gamma, A, B \vdash C}{\Gamma, A \wedge B \vdash C}$$

$\neg$*–elim:* $$\frac{\Gamma, \neg A \vdash A}{\Gamma, \neg A \vdash C}$$

$\rightarrow$*–elim:* $$\frac{\Gamma, A \rightarrow B \vdash A \quad \Gamma, B \vdash C}{\Gamma, A \rightarrow B \vdash C}$$

$\forall$*–elim t:* $$\frac{\Gamma, \forall x.\, A, A[x\backslash t] \vdash C}{\Gamma, \forall x.\, A \vdash C}$$

$\exists$*–elim a:* $$\frac{\Gamma, A[x\backslash a] \vdash C}{\Gamma, \exists x.\, A \vdash C} \quad (a \text{ Eigenvar.})$$

Fig. 1. The rules of the calculi $\mathcal{LJ}_{NS}$ and $\mathcal{LJ}_{S}$

the succedent of a sequent. We therefore call it a *non-standard* calculus and denote it by $\mathcal{LJ}_{NS}$. A proof for its correctness and completeness can be found in [5]. Figure 1 presents the rules of both calculi simultaneously. Note that the antecedents and succedents of sequents are *sets* of formulae instead of sequences which allows to omit structural rules like weakening and contraction. We shall use these rules in an *analytic* manner, i.e. for reasoning from the conclusion to the premises. Thus the starting point of a derivation is a *goal sequent* of the form $\vdash A$ where A is the formula which has to be proven valid. To support this kind of reasoning the rules of $\mathcal{LJ}_{S}$ are a slight modification of the original ones given by Gentzen in [8].

For the matrix characterization $\mathcal{MJ}$ each formula A will be represented by its *formula tree*. In this tree each node is attached with a *position* k which uniquely describes a sub-formula of A which we denote by $lab(k)$. The formula tree corresponds to an irreflexive and non-transitive *ordering relation* $\ll^\mu$: a so-called *transition* $(a,b) \in \ll^\mu$ describes the fact that $lab(b)$ is a direct sub-formula of $lab(a)$. Besides $lab(k)$ some other informations about this sub-formula are assigned to the node k. For the following definition we assume the reader to be familiar with positive and negative occurrences of sub-formulae in a formula A.

Definition 1. *(Polarity, signed formula)*
Let k be a position in $\ll^\mu$ representing a sub-formula $lab(k)$ of A. Then the polarity *$pol(k)$ of k is defined by*

$$pol(k) = \begin{cases} 0 \text{ if } lab(k) \text{ occurs positively in } A \\ 1 \text{ if } lab(k) \text{ occurs negatively in } A \end{cases}$$

A signed sub-formula *of A is a pair $\langle B, n\rangle$ where $n = 0$ if the sub-formula B occurs positively in A and $n = 1$ otherwise. The* signed sub-formula *$\langle lab(k), pol(k)\rangle$ related to a position k is denoted by $sform(k)$.*

If $pol(k) = 1$ the corresponding sub-formula $lab(k)$ will become an antecedent formula in the $\mathcal{LJ}_{NS}$–proof and $pol(k) = 0$ denotes the membership to the succedent formulae, respectively.

The *multiplicity* $\mu(k) = \langle \mu_Q(k), \mu_J(k)\rangle$ denotes the number of instances of sub-formulae $lab(k)$ which are needed for completing the matrix proof. Using several instances of $lab(k)$ may be due to multiple instantiations of quantifiers (as in classical deduction), denoted by $\mu_Q(k)$ or to multiple instantiations $\mu_J(k)$ which are necessary for intuitionistic reasoning. In the tree ordering $\ll^\mu$ of A these multiplicities are represented by 'duplicated' labels corresponding to the same sub-formula.

As in classical deduction multiplicities are required for constructing a global *substitution* $\sigma = \langle \sigma_Q, \sigma_J\rangle$ which simultaneously makes all '*connected*' literals equal. To support this construction both parts of a substitution are defined as mappings on the positions in $\ll^\mu$ which are divided into *variables* (marked with an over bar) and *constants*. Similar to the multiplicities μ_Q and μ_J one substitution (σ_Q) corresponds to the instantiation of quantifiers and the other (σ_J) to the instantiation of *intuitionistic variables*. The former realizes the well known Eigenvariable restrictions on an $\mathcal{LJ}_{NS}$–proof which depend on the polarity of the quantifier sub-formulae. The latter is a new aspect in the characterization $\mathcal{MJ}$ which encodes the *non-permutability* of applying intuitionistic rules. σ_J has to unify the so-called *prefixes* of the connected literals in the matrix proof where a prefix of a position k is a string consisting of all the preceding *special* positions[1] in the tree ordering. σ_J takes into account the Kripke semantics of the special operators $\forall, \neg, \rightarrow$ (for details see [5, 15]) and defines intuitionistic restrictions on the corresponding $\mathcal{LJ}_{NS}$–proof.

[1] A formula A is called *special* iff its is of the form $\forall x.B, \neg B, B \rightarrow C$ or A is an atom. A position k is called special iff $lab(k)$ is a special formula.

Both substitutions induce relations $\sqsubset_Q$ and $\sqsubset_J$ consisting of pairs $(\sigma_Q(\bar{a}), a)$ and $(\sigma_J(\bar{a}), a)$ respectively. $(b, a) \in \sqsubset_Q$ indicates that $lab(b)$ has to be reduced before $lab(a)$ in a $\mathcal{LJ}_{NS}$–proof, which essentially is the Eigenvariable condition. The union of the tree ordering $\ll^\mu$ with the substitution relation $\sqsubset = \sqsubset_Q \cup \sqsubset_J$ yields a *reduction ordering* on the positions. Each position k of $\ll^\mu$ uniquely corresponds to a $\mathcal{LJ}_{NS}$–rule which depends on the actual sub-formula $lab(k)$ and its polarity $pol(k)$. The relation $\sqsubset$ represents the restrictions given by the non-permutability of applying rules in a $\mathcal{LJ}_{NS}$–proof. The complete reduction ordering is given by the transitive closure $\lhd = (\ll^\mu \cup \sqsubset)^+$.

For the characterization $\mathcal{MJ}$ we call $\sigma = \langle \sigma_Q, \sigma_J \rangle$ a *combined* substitution and $\mu = \langle \mu_Q, \mu_J \rangle$ a multiplicity. Then $\langle A, n \rangle^\mu$ ($n \in \{0, 1\}$) is an indexed signed formula, i.e. a signed formula $\langle A, n \rangle$ where all multiplicities of sub-formulae of A are given. On this basis Wallen [15] defines the notions of *paths* through an indexed signed formula $\langle A, n \rangle^\mu$ and of *σ-complementary* connections between formulae. Both extend the classical notions given in [2]. Wallen also defines *$\mathcal{J}$–admissibility* of a combined substitution which essentially means that the induced ordering $\lhd$ is irreflexive and can thus be represented by a directed acyclic graph. A set of connections *spans* a formula $\langle A, 0 \rangle^\mu$ if every *atomic* path through it contains a connection from this set. Based on these notions a characterization for the validity of an intuitionistic formula can be formulated as follows:

Theorem 1. *(Characterization theorem for $\mathcal{MJ}$ [15])*
An intuitionistic formula A is $\mathcal{J}$–valid if and only if there is a multiplicity μ, for the signed formula $\langle A, 0 \rangle$, a $\mathcal{J}$–admissible combined substitution σ, and a set of σ–complementary connections that span $\langle A, 0 \rangle^\mu$.

The correctness and completeness proofs of $\mathcal{MJ}$ are given in [15]. For detailed reading of the characterization see chapter 5 and 8.

3 The transformation $\mathcal{MJ} \longmapsto \mathcal{LJ}_{NS}$

To describe the first step of our proof transformation procedure we assume that an efficient proof procedure based on the characterization theorem for $\mathcal{MJ}$ has generated a matrix proof $M(A) = (A, \mu, \sigma)$ for a given (valid) formula A. To convert this proof into a non-standard sequent proof we have realized the correctness and completeness proof of theorem 1 in an algorithmic manner. Since the original proof is already very long and many details have to be considered in this algorithm we restrict ourselves to presenting our method informally.

Our starting point is the reduction ordering $\lhd$ which can be constructed from (A, μ, σ) as described above. The key idea of our procedure is to traverse this relation such that each transition (a, b) of the tree ordering $\ll^\mu$ of A results in a reduction of the formula $lab(a)$ represented by the position a. The type of this reduction depends on the main operator of $lab(a)$ and its polarity $pol(a)$. These informations uniquely identify the $\mathcal{LJ}_{NS}$–rule to be applied for reducing this formula during the proof. The non-permutability of $\mathcal{LJ}_{NS}$–rules is taken into account by the relations $\sqsubset_Q$ and $\sqsubset_J$ which are extracted from the substitutions

```
TOTAL(∝⋆)
  For all positions k do compute the label wait₁(k) od
  while not proven(∝⋆) do
    select (z,k) ∈ (≪^μ)⋆ with solved(z) and not solved(k)
    compute the label wait₂(k)
    if not wait₁(k) and not wait₂(k) then
      set solved(k) := ⊤
      delete the labels wait₁(y) of all positions {y | (k,y) ∈ ⊏}
      if k is no intuitionistic position² then
        reduce the sequent with the unique LJ_NS–rule from ⟨lab(k), pol(k)⟩
        if the axiom–rule has been applied then
          set proven(∝⋆) := ⊤
        else
          if k is a β position then
            split the relation ∝⋆ into ∝⋆₁ and ∝⋆₂
            recursively call TOTAL(∝⋆₁) and TOTAL(∝⋆₂)
            set proven(∝⋆) := ⊤
  od
```

Fig. 2. The main procedure transforming $\propto^\star$ into a linear sequence of $\mathcal{LJ}_{NS}$–rules

σ_Q and σ_J. The transformation is finished when all produced sequents contain the axiom $F \vdash F$.

In the following we focus on the tree structure of $\lhd$ together with the relation $\sqsubset$. We delete the transitivity elements of the relation $\lhd$ and call the result $\propto$. We refine this relation to $\propto^\star$ by eliminating all irrelevant positions, i.e. all positions which have no connection position in their transitive successors in $\ll^\mu$. Then $(\ll^\mu)^\star$ is the tree ordering related to $\propto^\star$. For the purpose of proof construction the relation $\sqsubset$ can be interpreted as defining *wait labels* on the positions which represent the non-permutability of the $\mathcal{LJ}_{NS}$–rules. For a position k there are two types of wait labels $wait_1(k)$ and $wait_2(k)$. The first can be computed in a preprocessing step for all positions k of $\propto^\star$ which are restricted by $\sqsubset$: $wait_1(k)$ holds iff $(y,k) \in \sqsubset$ for an arbitrary position y. The second type of wait labels depends on the actual order of traversing the relation $\propto^\star$: $wait_2(k)$ holds iff the corresponding $\mathcal{LJ}_{NS}$–rule would cause a deletion of a relevant succedent formula (represented by another position y) in the $\mathcal{LJ}_{NS}$–proof. So the wait labels of secondary type have to be computed for each position k *after* its selection for the next reduction step. This guarantees that the actual state of the relation $\propto^\star$ representing the actual sequent in the $\mathcal{LJ}_{NS}$–proof is taken into account. The basic structure of the main procedure $TOTAL(\propto^\star)$ transforming the partial ordering $\propto^\star$ into a linear sequence of $\mathcal{LJ}_{NS}$–rules is presented in figure 2.

[2] Each special position has a predecessor in $(\ll^\mu)^\star$ which is called an *intuitionistic* position. Such positions are necessary because of the relation to the matrix characterization for modal logic $S4$, i.e. the Kripke semantic similarity between the two logics [15]. The assignments at an intuitionistic position are equal to the ones at the corresponding special position. For extracting $\mathcal{LJ}_{NS}$–rules they are not relevant and must be ignored.

At the beginning all positions k except the root position of $(\ll^{\mu})^{\star}$ are marked as unsolved (i.e. not yet reduced): we set $solved(k) := \bot$. Furthermore we set $proven(\propto^{\star}) := \bot$. The goal sequent on part of the $\mathcal{LJ}_{NS}$–proof is initialized with $\vdash A$. The condition $proven(\propto^{\star})$ becomes true if either the sequent reduction has been completed by an *axiom*–rule or both recursive calls after a *split* have terminated successfully. The selection of the position k to be reduced next guarantees the fact that the formula $lab(k)$ is an isolated element of the actual sequent, i.e. that it is not any longer a sub-formula. This is true because the predecessor position of k (the position z) has already been solved. Further the labels $wait_1(y)$ of positions y for which $(k, y) \in \sqsubset$ have to be deleted if k is marked as solved. The termination of the whole procedure is given by the two cases for $proven(\propto^{\star}) := \top$. The selection of the position k to be considered next terminates under the condition that there exists a "fair strategy" for this process, i.e. all positions which fulfill the selection condition are taken into account in a finite time. So it is not possible that a position k for which $wait_1(k)$ or $wait_2(k)$ holds will be selected continuously.

The detailed procedure is rather complicated because of the complex definition of the wait labels and the extended *split* procedure at *β–positions* which represent branching points in the sequent proof and correspond to the rules ∧–*intro*, ∨–*elim* and →–*elim*. The *split* is realized by first deleting the successor relation β_2 of the β–position in the sub-relation $\propto_1^{\star}$ and β_1 in $\propto_2^{\star}$, respectively. Afterwards an optimization of the remaining sub-relations is performed by iterating reduction procedures on $\propto_i^{\star}, i \in \{1, 2\}$ until all positions in both relations which are not longer relevant for the proof are eliminated. After the split the recursive calls have to compute wait labels again because they may have changed considerably. A simple example shall illustrate the construction of an $\mathcal{LJ}_{NS}$–proof. For a detailed presentation of the transformation procedure and a proof of its correctness and completeness we refer to the authors technical report [11].

Example 1. Consider the propositional formula $\neg A \vee B \rightarrow B \vee \neg A$. Assume that a matrix proof $M(A)$ has already been constructed and that $\sqsubset = \sqsubset_J = \{(a_7, \bar{a}_2^1), (a_3^1, \bar{a}_8^1), (a_6, \bar{a}_4^1)\}$ (containing the two connections) has been created from the substitution $\sigma = \sigma_J$. The complete relation $\propto^{\star}$ is shown on the left side of figure 3. Here all b–positions are intuitionistic positions. The assignments between (non intuitionistic) positions and formulae are shown below.

position k	a_0	a_1	$\bar{a}_2^1$	a_3^1	$\bar{a}_4^1$	a_5	a_6	a_7	$\bar{a}_8^1$
$lab(k)$	$\neg A \vee B \rightarrow B \vee \neg A$	$\neg A \vee B$	$\neg A$	A	B	$B \vee \neg A$	B	$\neg A$	A
$pol(k)$	0	1	1	0	1	0	0	0	1

We now motivate several possibilities for constructing a $\mathcal{LJ}_{NS}$–proof from $\propto^{\star}$. We have two wait labels relevant for the proof. First, position $\bar{a}_2^1$ (¬–*elim*) has to be reduced after a_7 (¬–*intro*) because applying ¬–*elim* before ¬–*intro* would make the succedent formula A disappear ($wait_1(\bar{a}_2^1)$). So one would have to repeat the ¬–*elim* step which does not occur in $\propto^{\star}$. Hence the transformation would end up in a blind alley and the $\mathcal{LJ}_{NS}$–proof could not be found. Second, the

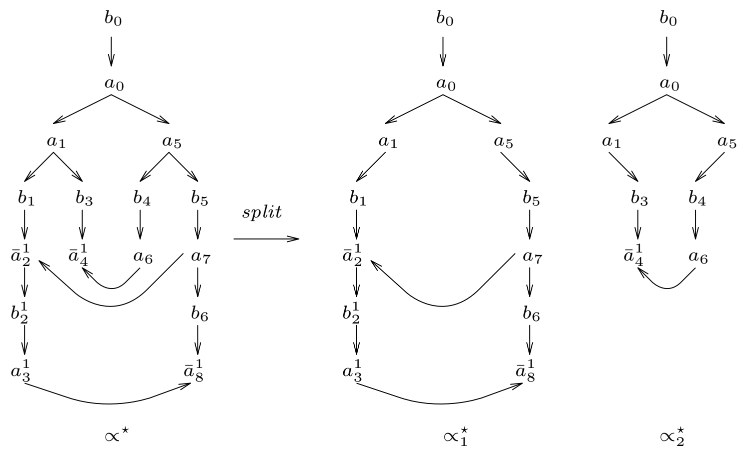

Fig. 3. The split of $\propto^\star$ at position a_1 (example 1)

branching $\vee$–*elim* at the only β–position given by a_1 has to be performed before reducing a_7 ($wait_2(a_7)$). Otherwise the relevant succedent formula B would be deleted and the proof could not be completed. The latter wait label shows that the reduction ordering is not complete, i.e. the relation $\sqsubset$ does not encode *all* restrictions which have to be taken into account. A possible selection and solve ordering is given by

$$[a_0, a_5, a_1, a_7, \bar{a}_2^1, a_3^1, \bar{a}_8^1, a_6, \bar{a}_4^1].$$

The *split* at a_1 is performed after solving a_5 in order to apply the $\vee$–*intro* rule only once. If we would branch before a_5 is solved we would have to copy the $\vee$–*intro* step into each subproof. The split together with the elimination positions and sub-relations which are not longer proof relevant in $\propto_1^\star$ and $\propto_1^\star$ is shown on the right side of Figure 3. The complete $\mathcal{LJ}_{NS}$–proof extracted from the above ordering is presented below.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg A, A \vdash A}\;\textit{axiom } A\ (a_3^1, \bar{a}_8^1)}{\neg A, A \vdash}\;\neg\text{–}\textit{elim}\ (\bar{a}_2^1)}{\neg A \vdash B, \neg A}\;\neg\text{–}\textit{intro}\ (a_7) \qquad \dfrac{}{B \vdash B, \neg A}\;\textit{axiom } B\ (a_6, \bar{a}_4^1)}{\neg A \vee B \vdash B, \neg A}\;\vee\text{–}\textit{elim}\ (a_1)}{\neg A \vee B \vdash B \vee \neg A}\;\vee\text{–}\textit{intro}\ (a_5)}{\vdash \neg A \vee B \rightarrow B \vee \neg A}\;\rightarrow\text{–}\textit{intro}\ (a_0)$$

4 An extended standard–sequent calculus $\mathcal{LJ}_S^\star$

In this section we prove that there cannot be a proof transformation which maps every $\mathcal{LJ}_{NS}$–proof into an equivalent $\mathcal{LJ}_S$–proof having the same structure. To preserve the proof structure of the given $\mathcal{LJ}_{NS}$–proof it will therefore be necessary to introduce an extension $\mathcal{LJ}_S^\star$ of the standard calculus $\mathcal{LJ}_S$. First, however, let us define what we mean by a structure preserving proof transformation.

Definition 2. *($\mathcal{K}$–proof)*
Let $\mathcal{K}$ be a sequent calculus which is sound and complete. A sequence S of rules is called a $\mathcal{K}$ proof*, if an application of the rules in the given order is a proof of a valid formula A representing the goal sequent.*
The length *of a $\mathcal{K}$ proof $|S|$ is given by the number of elements of S. For the set of all $\mathcal{K}$ proofs we write $\mathcal{S}_{\mathcal{K}}$.*

Recall that a sequence of rules S from the calculus $\mathcal{K}$ is applied in an analytic way, i.e. from the conclusions to the premises. If a rule splits a proof into two independent subproofs we represent the left branch in S before the right one.

Definition 3. *(Initial formula set)*
Let S be a $\mathcal{K}$–proof. Then the multiset[3]

$$\{I(S) = \{F \mid F \vdash F \text{ is part of a initial sequent in } S.\}$$

is called the initial formula set *of the $\mathcal{K}$–proof S.*

The initial formula set $I(S)$ describes the knowledge used within the proof S.

Definition 4. *(Structure preserving proof transformation)*
Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be two sound and complete calculi. A mapping $f : \mathcal{S}_{\mathcal{K}_1} \longmapsto \mathcal{S}_{\mathcal{K}_2}$ is a proof transformation *iff for all valid formulae A and an arbitrary $\mathcal{K}_1$–proof $S \in \mathcal{S}_{\mathcal{K}_1}$ for A the result $f(S) \in \mathcal{S}_{\mathcal{K}_2}$ is a proof for A in the calculus $\mathcal{K}_2$.*
A proof transformation $f : \mathcal{S}_{\mathcal{K}_1} \longmapsto \mathcal{S}_{\mathcal{K}_2}$ is structure preserving *w.r.t. $\mathcal{K}_1$ iff $I(S) = I(f(S))$ for every $\mathcal{K}_1$–proof S.*

Note that soundness and completeness of $\mathcal{K}_1$ and $\mathcal{K}_2$ already guarantees that there must be a transformation between $\mathcal{K}_1$–proofs and $\mathcal{K}_2$–proofs. This concept of structure preservation realizes the idea that the proofs S and $f(S)$ are founded on the same knowledge. This, however, cannot be achieved when transforming $\mathcal{LJ}_{NS}$–proofs into $\mathcal{LJ}_S$–proofs.

Theorem 2.
There is no structure preserving proof transformation $f_{\mathcal{LJ}_{NS}} : \mathcal{S}_{\mathcal{LJ}_{NS}} \longmapsto \mathcal{S}_{\mathcal{LJ}_S}$.

Proof: We give an counterexample, i.e. a proof $S_1 \in \mathcal{S}_{\mathcal{LJ}_{NS}}$ which cannot be represented by a standard-proof $S_2 \in \mathcal{S}_{\mathcal{LJ}_S}$ having the same structure. Consider the formula $(\forall x.A(x) \vee B(x)) \wedge (\exists y.A(y) \rightarrow \exists z.\neg A(z)) \rightarrow \exists x.B(x)$. Figure 4 shows a non-standard proof S_1. The corresponding standard proof[4] S_2 is given in Figure 5.

---

[3] A *multiset* is an unordered collection of elements in which elements may appear more than once. Multisets are denoted by the brackets **{** and **}** whereas sets are denoted as usual by { and }. The operations $\dot\subseteq$, $\dot\cup$, and $\dot-$ denote the multiset extensions of the usual set operations $\subseteq$, $\cup$, and $-$.

[4] For better overview we present both proofs in the usual proof style without writing the *axiom*–rules. The list form described in Definition 2 can easily be constructed from the proof figures.

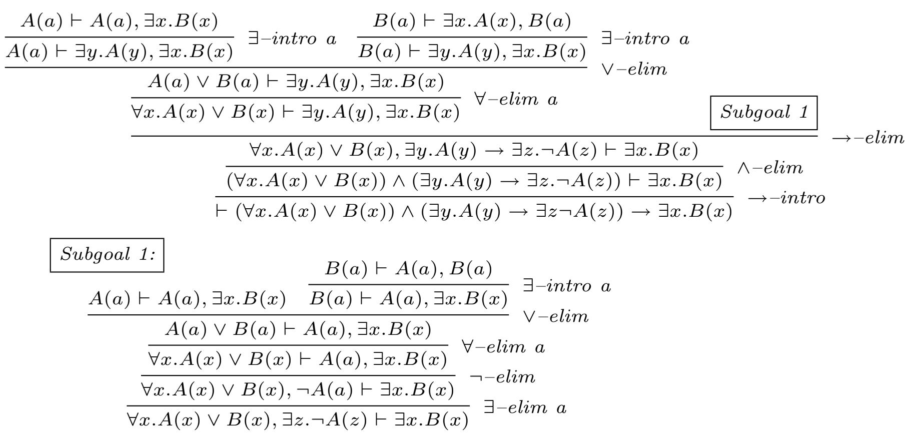

Fig. 4. A non-standard $\mathcal{LJ}_{NS}$–proof S_1 for the formula of the above proof.

The initial formula sets of the proofs are obviously not equal:

$$I(S_1) = \{A(a), B(a), A(a), B(a)\} \neq \{A(t), A(a), B(a), B(t)\} = I(S_2)$$

It is easy to see that the knowledge contained in S_2 *must* be different from that in S_1. Because of the Eigenvariable in the sub-formula $\exists z.\neg A(z)$ two different instantiations of the sub-formula $\forall x.A(x) \vee B(x)$ are *needed* in the $\mathcal{LJ}_S$–proof S_2. This holds for *all* $\mathcal{LJ}_S$–proofs of the given formula since no other rule than $\forall$–*elim t* is applicable in the proof at this time. Thus a transformation from $\mathcal{LJ}_{NS}$ to $\mathcal{LJ}_S$ preserving the structure of S_1 cannot exist. (Q.e.d.)

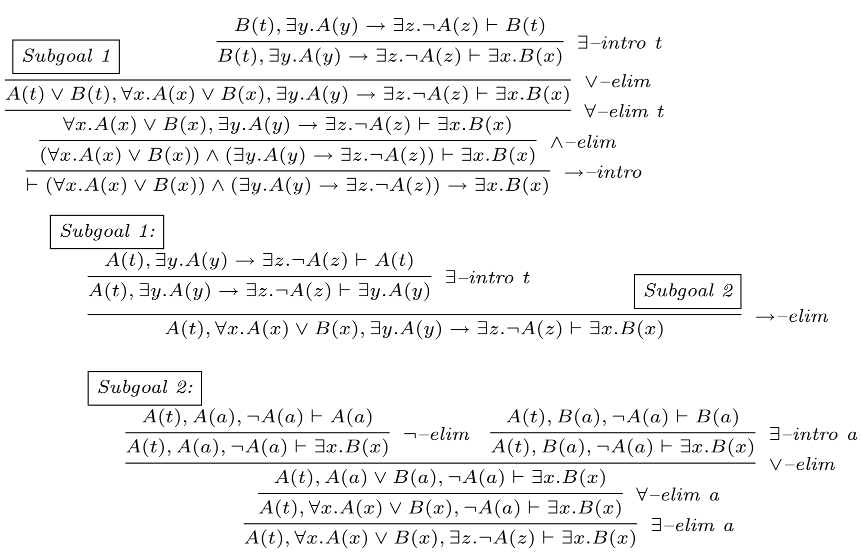

Fig. 5. A standard $\mathcal{LJ}_S$–proof S_2 for the formula of the above proof.

$$\frac{\Gamma \vdash A_i \vee (\Delta_S^i)}{\Gamma \vdash \Delta_S} \ \vee\text{–}change\ A_i$$

$$\frac{\Gamma, A \to B \vdash A \vee (\Delta_S) \quad \Gamma, B \vdash \Delta_S}{\Gamma, A \to B \vdash \Delta_S} \to(\vee)\text{–}elim \qquad \frac{\Gamma, \neg A \vdash A \vee (\Delta_S)}{\Gamma, \neg A \vdash \Delta_S} \neg(\vee)\text{–}elim$$

$$\frac{\Gamma \vdash A \vee (B \vee (\Delta_S))}{\Gamma \vdash (A \vee B) \vee (\Delta_S)} \vee(\vee)\text{–}intro \qquad \frac{\Gamma \vdash A \vee (\Delta_S) \quad \Gamma \vdash B \vee (\Delta_S)}{\Gamma \vdash (A \wedge B) \vee (\Delta_S)} \wedge(\vee)\text{–}intro$$

$$\frac{\Gamma \vdash A[x \backslash t] \vee ((\exists x.A) \vee (\Delta_S))}{\Gamma \vdash (\exists x.A) \vee (\Delta_S)} \exists(\vee)\text{–}intro\ t \qquad \frac{\Gamma \vdash A[x \backslash t] \vee \exists x.A}{\Gamma \vdash \exists x.A} \exists(\vee)^*\text{–}intro\ t$$

Fig. 6. The additional rules of the calculus $\mathcal{LJ}_S^\star$

A consequence of this result is that it is impossible to construct structure preserving $\mathcal{LJ}_S$–proofs from given $\mathcal{LJ}_{NS}$–proofs in general. The reason for this are different initial formula sets caused by the liberty to have more than one succedent formula in the $\mathcal{LJ}_{NS}$–proof. To simulate this feature in a corresponding standard proof it is necessary to represent the set of succedent formulae in *one* formula.

Definition 5. *(Standard succedent)*
Let $\Delta = A_1, A_2, \ldots, A_n, n \geq 1$ be a set of formulae denoting a succedent in a sequent $\Gamma \vdash \Delta$ of a $\mathcal{LJ}_{NS}$–proof. A formula $\Delta_S = A_{i(1)} \vee A_{i(2)} \vee \ldots \vee A_{i(n)}$ is called a standard succedent *of Δ where $i : \{1, 2, \ldots, n\} \longmapsto \{1, 2, \ldots, n\}$ is a bijective mapping (representing the associativity and commutativity of $\vee$).*
A standard succedent is called right–associative *iff $\Delta_S = A_{i(1)} \vee (A_{i(2)} \vee (\ldots \vee (A_{i(n-1)} \vee A_{i(n)}) \ldots)$ where i denotes an arbitrary permutation.*

Now we are able to create rules which allow to simulate the sets of formula in the $\mathcal{LJ}_{NS}$ succedents within standard proofs. These rules are added to $\mathcal{LJ}_S$ and result in a set of rules which we call the extended standard calculus $\mathcal{LJ}_S^\star$.

Definition 6. *(The calculus $\mathcal{LJ}_S^\star$)*
Let Δ_S be a standard succedent representing a $\mathcal{LJ}_{NS}$ succedent Δ with $|\Delta| = n \geq 1$. Define

$$\Delta_S^i = A_1 \vee (A_2 \vee (\ldots (A_{i-1} \vee (A_{i+1} \vee (\ldots (A_{n-1} \vee A_n) \ldots)))),$$

for all $2 \leq i \leq n$. Let $\mathcal{R}^*$ the set of the additional rules shown in figure 6 and $\mathcal{R}(\mathcal{LJ}_S)$ is the rule set of the standard calculus $\mathcal{LJ}_S$ (see figure 1). Then the extended standard calculus *$\mathcal{LJ}_S^\star$ is characterized by $\mathcal{R}(\mathcal{LJ}_S^\star) = \mathcal{R}(\mathcal{LJ}_S) \cup \mathcal{R}^*$.*

The brackets around the standard succedents (for example in $A \vee (\Delta_S)$) denote the right–associativity of the newly created formula. One can see that all standard succedents built by the new rules have this property. Thus a formula A_i in Δ_S which will be reduced next has to be the leftmost formula of this disjunction. To put the selected formula A_i to the leftmost position before reducing it we need the structural rule $\vee$–*change* A_i. Note that for the case $|\Delta| = 0$ we need an additional special rule $\exists(\vee)^*$–*intro* because an empty formula Δ_S does not exist and consequently the conclusion $(\exists x.A) \vee (\Delta_S)$ of the rule $\exists(\vee)$–*intro* would not be well formed.

Theorem 3. *([11])*
The extended standard calculus $\mathcal{LJ}_S^\star$ is sound and complete.

Soundness of $\mathcal{LJ}_S^\star$ can be proven by representing each of the additional rules given in figure 6 by a series of rules from $\mathcal{LJ}_S$ [5] (including the cut rule). Completeness follows from the fact that $\mathcal{LJ}_S^\star$ extends $\mathcal{LJ}_S$ which is already complete.

5 The transformation $\mathcal{LJ}_{NS} \longmapsto \mathcal{LJ}_S^\star$

In this section we use the calculus $\mathcal{LJ}_S^\star$ to construct standard proofs from $\mathcal{LJ}_{NS}$–proofs in a structure preserving way. This is done by simulating each rule in the calculus $\mathcal{LJ}_{NS}$ with a $\mathcal{LJ}_S^\star$–rule. In addition some rules are needed in the $\mathcal{LJ}_S^\star$–proof to identify a formula A in the disjunction Δ_S which has to be reduced next. On the part of the $\mathcal{LJ}_{NS}$–proofs this is not necessary because the formula $A \in \Delta$ is accessible in a direct way.

Definition 7. *(Rule mapping)*
Let Δ denote the succedent of an arbitrary $\mathcal{LJ}_{NS}$–proof ($|\Delta| \geq 0$). The mapping $\varphi : \mathcal{R}(\mathcal{LJ}_{NS}) \longmapsto \mathcal{R}(\mathcal{LJ}_S^\star)$ is defined by

1. $\varphi(r) = r$ *for all* $r \in \{\forall\text{–}elim\ t, \exists\text{–}elim\ a, \vee\text{–}elim, \wedge\text{–}elim\}$.
   (These rules do not affect the succedents.)
2. *For* $r \in \{\neg, \rightarrow\}$*:* $\varphi(r\text{–}elim) = \begin{cases} r\text{–}elim, & \text{if } |\Delta| = 0 \\ r(\vee)\text{–}elim, & \text{if } |\Delta| \geq 1. \end{cases}$
3. $\varphi(\wedge\text{–}intro) = \begin{cases} \wedge\text{–}intro, & \text{if } |\Delta| = 1 \\ \wedge(\vee)\text{–}intro, & \text{if } |\Delta| \geq 2. \end{cases}$
4. $\varphi(\exists\text{–}intro\ t) = \begin{cases} \exists(\vee)^*\text{–}intro\ t, & \text{if } |\Delta| = 1 \\ \exists(\vee)\text{–}intro\ t, & \text{if } |\Delta| \geq 2. \end{cases}$
5. $\varphi(\vee\text{–}intro) = \begin{cases} \varepsilon, & \text{if } |\Delta| = 1^6 \\ \vee(\vee)\text{–}intro, & \text{if } |\Delta| \geq 2. \end{cases}$
6. $\varphi(r) = r$ *for all* $r \in \{\forall\text{–}intro\ a, \neg\text{–}intro, \rightarrow\text{–}intro, axiom\ A\}$.
   (After an application of r in the $\mathcal{LJ}_{NS}$–proof all succedent formulae which are not involved in the reduction have lost their proof relevance.)

For each rule $r \in \mathcal{R}(\mathcal{LJ}_{NS})$ the rule $\varphi(r)$ is the corresponding version in $\mathcal{LJ}_S^\star$. In some cases additional steps are necessary before using the rule $\varphi(r)$.

[5] This also shows how to implement these rules as a NuPRL tactic
[6] ε denotes the empty rule which leaves the sequent of the $\mathcal{LJ}_S^\star$–proof unchanged.

Definition 8.
Let $A_i \in \Delta$ the formula to be reduced next in the $\mathcal{LJ}_{NS}$–proof and let Δ_S be the corresponding standard succedent in the $\mathcal{LJ}_S^\star$–proof. The the sequence $R(r)$ of additional steps which have to be applied before $\varphi(r)$ is defined by

1. *If $\varphi(r) \in \{\wedge(\vee)\text{–}intro, \vee(\vee)\text{–}intro, \exists(\vee)\text{–}intro\ t\}$ and $i \geq 2$ then*
   $R(r) = [\vee\text{–}change\ A_i]$
2. *If $\varphi(r) \in \{\forall\text{–}intro\ a, \neg\text{–}intro, \rightarrow\text{–}intro, axiom\ A\}$ then*
   $$R(r) = \begin{cases} [\vee\text{–}intro\ 1], & \text{if } i = 1, \\ [\vee\text{–}change\ A_i, \vee\text{–}intro\ 1], & \text{if } i \geq 2. \end{cases}$$
   (The set $\Delta \setminus \{A_i\}$ is not longer proof relevant.)
3. *$R(r) = []$ otherwise.*

The rules $R(r)$ cause an insignificant expansion of the $\mathcal{LJ}_S^\star$–proof: $|S_2| \geq |S_1|$ for all $S_1 \in \mathcal{S}_{\mathcal{LJ}_{NS}}$ and $S_2 = f(S_1) \in \mathcal{S}_{\mathcal{LJ}_S^\star}$. Using the rule mapping and the sequence of additional steps we can define a transformation $f : \mathcal{S}_{\mathcal{LJ}_{NS}} \longmapsto \mathcal{S}_{\mathcal{LJ}_S^\star}$ such that all non-standard $\mathcal{LJ}_{NS}$–proofs will be transformed into standard $\mathcal{LJ}_S^\star$–proofs preserving the proof structure, i.e. $\forall S_1 \in \mathcal{S}_{\mathcal{LJ}_{NS}}.I(S_1) = I(f(S_1))$. Thus we have proven the following theorem.

Theorem 4.
There exists a structure preserving proof transformation $f : \mathcal{S}_{\mathcal{LJ}_{NS}} \longmapsto \mathcal{S}_{\mathcal{LJ}_S^\star}$.

We conclude this section with an example.

Example 2.
Consider the non-standard $\mathcal{LJ}_{NS}$–proof S_1 of figure 4. Using the transformation according to definitions 7 and 8 we get the following assignments:

S_1	S_3	
r	$R(r)$	$\varphi(r)$
$\rightarrow$–*intro*	[]	$\rightarrow$–*intro*
$\wedge$–*elim*	[]	$\wedge$–*elim*
$\rightarrow$–*elim*	[]	$\rightarrow(\vee)$–*elim*
$\forall$–*elim a*	[]	$\forall$–*elim a*
$\vee$–*elim*	[]	$\vee$–*elim*
$\exists$–*intro a*	[]	$\exists(\vee)$–*intro a*
axiom $A(a)$	[$\vee$–*intro 1*]	*by–axiom* $A(a)$
$\exists$–*intro a*	[$\vee$–*change* $\exists x.B(x)$]	$\exists(\vee)$–*intro a*
axiom $B(a)$	[$\vee$–*intro 1*]	*axiom* $B(a)$
$\exists$–*elim a*	[]	$\exists$–*elim a*
$\neg$–*elim*	[]	$\neg(\vee)$–*elim*
$\forall$–*elim a*	[]	$\forall$–*elim a*
$\vee$–*elim*	[]	$\vee$–*elim*
axiom $A(a)$	[$\vee$–*intro 1*]	*axiom* $A(a)$
$\exists$–*intro a*	[$\vee$–*change* $\exists x.B(x)$]	$\exists(\vee)$–*intro a*
axiom $B(a)$	[$\vee$–*intro 1*]	*axiom* $B(a)$

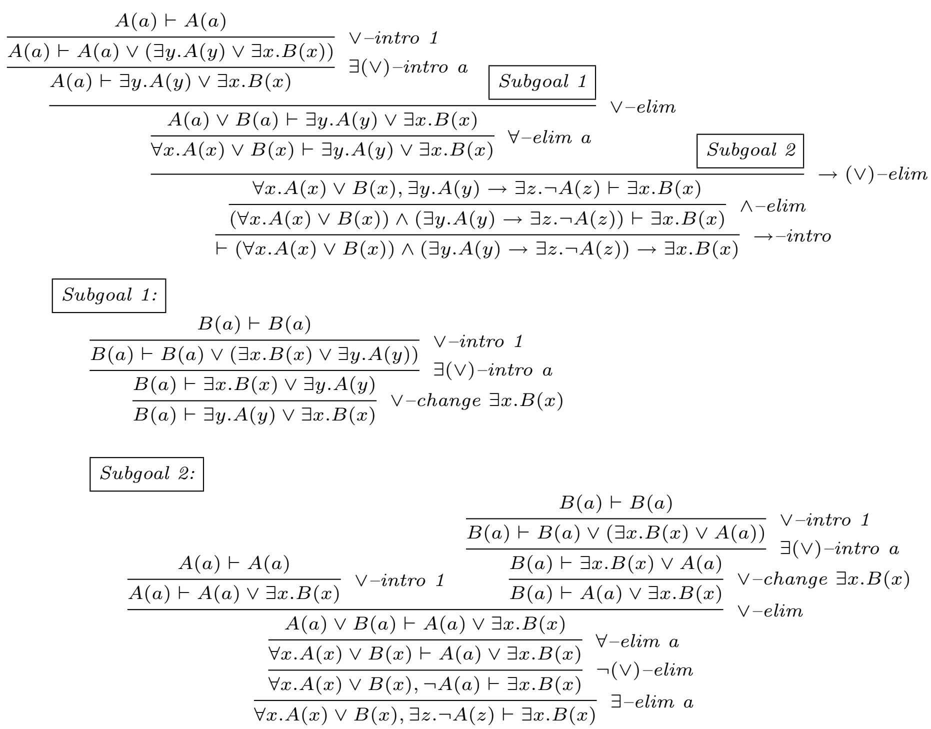

Fig. 7. The resulting $\mathcal{LJ}_S^\star$–proof of example 2

These assignments yield the following sequence of rules representing a standard proof $S_3 \in \mathcal{S}_{\mathcal{LJ}_S^\star}$:

$$S_3 = [\rightarrow\text{–}intro, \wedge\text{–}elim, \rightarrow(\vee)\text{–}elim, \forall\text{–}elim\ a, \vee\text{–}elim, \exists(\vee)\text{–}intro\ a,\\ \vee\text{–}intro\ 1, axiom\ A(a), \vee\text{–}change\ \exists x.B(x)\ , \exists(\vee)\text{–}intro\ a, \vee\text{–}intro\ 1,\\ axiom\ B(a), \exists\text{–}elim\ a, \neg(\vee)\text{–}elim, \forall\text{–}elim\ a, \vee\text{–}elim, \vee\text{–}intro\ 1,\\ axiom\ A(a), \vee\text{–}change\ \exists x.B(x)\ , \exists(\vee)\text{–}intro\ a, \vee\text{–}intro\ 1, axiom\ B(a)].$$

Recall that in a rule sequence the left subproof after a branching point is represented first A proof in the usual proof style is depicted in Figure 7. This proof has the same structure as $S_1 \in \mathcal{S}_{\mathcal{LJ}_{NS}}$ since $I(S_1) = \{A(a), B(a), A(a), B(a)\} = I(S_3)$. The expansion on the part of the $\mathcal{LJ}_S^\star$–proof is given by the additional proof steps $R(r)$, i.e. $|S_3| = |S_1| + 6$.

6 Conclusion

In this paper we have presented a two-step procedure transforming intuitionistic matrix proofs into proofs within a standard sequent calculus. In its first step the procedure converts an $\mathcal{MJ}$–proof – i.e. a proof according to L. Wallen's matrix characterization for the validity of intuitionistic formulae – into Fitting's non-standard sequent calculus $\mathcal{LJ}_{NS}$. This step essentially evaluates the reduction ordering $\lhd$ implicitly contained in the $\mathcal{MJ}$–proof and determines the $\mathcal{LJ}_{NS}$–rule to be applied at a given position k according to the information contained in its label $lab(k)$ and its polarity $pol(k)$. No search is involved in this step. To perform the second step – a conversion of $\mathcal{LJ}_{NS}$–proofs into standard sequent proofs while preserving the principal structure of the proof – we had to extend Gentzen's calculus $\mathcal{LJ}_S$ into an 'extended' standard calculus $\mathcal{LJ}_S^\star$ and to prove this calculus to be sound, complete, and compatible with $\mathcal{LJ}_S$. Given this calculus the conversion of $\mathcal{LJ}_{NS}$–proofs into standard proofs is very simple (for instance compared to the one presented in [14] or [6, p. 40]) and keeps the size of the resulting proof small. Again, no search is involved in the process. Thus it is possible to convert a matrix proofs which can be efficiently constructed by a machine into sequent proofs which are comprehensible for mathematicians and programmers.

In order to create such a matrix proof it will be necessary to extend Bibel's connection method [1, 2] for classical logic to one satisfying the additional requirements mentioned in theorem 1 and to develop an efficient algorithm for unifying prefixes. Both topics are currently being investigated (see [7, 10]). Among these the work of Otten and Kreitz (see [10]) attempts to integrate the transformation step $\mathcal{MJ} \longmapsto \mathcal{LJ}_{NS}$ (see section 3) into the extended connection method by traversing the reduction ordering $\propto^\star$ *during* the proof search. The proof procedure would thus construct the matrix proof and a sequent proof simultaneously. It is, however, not yet clear whether the integrated approach is more efficient than proceeding in two separate steps. Therefore both methods shall be investigated and compared in the future.

We have implemented the calculus $\mathcal{LJ}_S^\star$ with the NuPRL proof development system [4] by simulating $\mathcal{LJ}_S^\star$–rules via tactics. The implementation of our transformation procedures as meta-programs of NuPRL is progressing and will soon be finished. As a consequence matrix-based proof methods for first-order intuitionistic logic can be used within a larger environment for reasoning about programming and many other kinds of applied mathematics.

References

1. W. Bibel. On matrices with connections. *Jour. of the ACM*, 28, p. 633–645, 1981.
2. W. Bibel. *Automated Theorem Proving.* Vieweg Verlag 1987.
3. W. Bibel, S. Brüning, U. Egly, and T. Rath. Komet. In *Proceedings of the Conference on Automated Deduction*, Lecture Notes in Artificial Intelligence 814, pages 783–787. Springer Verlag, 1994.
4. R. L. Constable et al. *Implementing Mathematics with the NuPRL proof development system.* Prentice Hall,1986.
5. M. C. Fitting. *Intuitionistic logic, model theory and forcing.* Studies in logic and the foundations of mathematics. North–Holland, 1969.
6. J. Gallier. Constructive logics. Part I: A tutorial on proof systems and typed λ-calculi. Technical Report 8, Digital Equipment Corporation, 1991.

7. I. Gassmann. Eine Matrix-basierte Beweisprozedur für intuitionistische Logik. Master's thesis, TH Darmstadt, FG Intellektik, 1994.
8. G. Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935.
9. R. Letz, J. Schumann, S. Bayerl, and W. Bibel. Setheo: A high-performance theorem prover. *Journal of Automated Reasoning*, 8:183–212, 1992.
10. J. Otten and C. Kreitz. A connection based proof method for intuitionistic logic. In *Proceedings Tableaux Workshop 1995*, this volume.
11. S. Schmitt. Ein erweiterter intuitionistischer Sequenzenkalkül und dessen Anwendung im intuitionistischen Konnektionsbeweisen. Master's thesis, TH Darmstadt, FG Intellektik, 1994.
12. K. Schütte. *Vollständige Systeme modaler und intuitionistischer Logik*. Ergebnisse der Mathematik und ihrer Grenzgebiete. Springer Verlag, 1968.
13. R. M. Smullyan. A generalization of intuitionistic and modal logics. In H. Leblanc, ed., *Truth, Syntax and Modality*, pages 274–293. North–Holland, 1973.
14. G. Takeuti. *Proof Theory*. North–Holland, 1975.
15. L. A. Wallen. *Automated deduction in nonclassical logic*. MIT Press, 1990.
16. L. Wos et. al. Automated reasoning contributes to mathematics and logic. In M. E. Stickel, editor, *Proceedings of the 10th Conference on Automated Deduction*, *Lecture Notes in Computer Science* 449, pages 485–499. Springer Verlag, 1990.