A Tableau-Like Representation Framework for Efficient Proof Reconstruction

Stephan Schmitt
Department of Computer Science, Cornell University
Ithaca, NY 14853, USA
steph@cs.cornell.edu

Abstract. We present a tableau-like framework, so-called \( \beta \)-proofs, for a uniform representation of analytic tableaux or matrix proofs. \( \beta \)-proofs suggest split structures for sequent proofs but still violate non-permutabilities of sequent rules. Two operations on \( \beta \)-proofs stepwisely solve these violations while removing all redundant inference steps from the \( \beta \)-proofs. This process provides a general basis for a search-free reconstruction of sequent proofs, independent from concrete proof calculi. Our framework is uniformly applicable to classical logic and to all the non-classical logics that have a matrix characterization of validity.

1 Introduction

Automated theorem proving in non-classical logics has become important in many branches of Artificial Intelligence and Computer Science. As a result, the resolution principle [25] and the connection method [3], which both have led to efficient theorem provers for classical logic [31][16][4], have been extended to characterizations of logical validity in modal logics, intuitionistic logic, and fragments of linear logic [21][30][12][18]. On this basis, uniform and efficient proof search procedures have been developed for all these logics [22][12][13][19].

In many applications of theorem proving it is not sufficient to show that a theorem is valid; for instance, when integrating theorem provers as inference engines into interactive program development systems [15] or other problem-oriented applications [6]. The need for further processing, e.g. generating programs from proofs, or a deeper understanding of the proof requires that proof details can be presented in a comprehensible form. Since the efficiency of automated proof methods strongly depends on a machine-oriented and compact characterization of logical validity, the reconstruction of sequent proofs or natural deduction proofs from automatically generated machine proofs becomes necessary.

Proof reconstruction in classical [20][23][24] and non-classical logics [27][28][12][15] provides uniform procedures for constructing sequent proofs from machine-generated matrix proofs, i.e. within matrix calculi or analytic tableaux. Thesequent proof for the input formula is created by traversing its formula tree in an order which respects a reduction ordering induced by the matrix proof. It selects an appropriate sequent rule for each visited node by consulting tables.
that represent the peculiarities of the different logics. At nodes which cause the sequent proof to branch, the reduction ordering has to be divided appropriately and certain redundancies need to be eliminated in order to ensure completeness.

As discussed in [29, 15], redundancy elimination is the most crucial aspect during proof reconstruction: performing a complete redundancy deletion ensures a reconstruction process without additional search. This avoids redundant proof steps in the sequent proofs, which are impossible to identify when search is involved. Thus, we obtain an efficient method for a goal oriented construction of the sequent proofs. In order to explore all kinds of redundancy, the inference steps of a matrix proof need to be represented, encoding additional information about its internal structure. Usually, the inference steps are determined by the required proof steps for proving the input formula valid.

In this paper we present a uniform framework for the representation of matrix proofs, by generalizing the concept of inference steps. The framework is independent from the selected logic and, in addition, from the underlying proof calculus. Thus, it will be uniformly applicable to various proof methods for classical and non-classical logics, based on matrix or tableau calculi. We will introduce \( \beta \)-proofs as a tableau-like representation formalism, and investigate its correspondence to sequent proofs: \( \beta \)-proofs determine possible “split structures” for sequent proofs, but may still violate non-permutabilities of sequent rules. We will develop two elegant operations, which stepwisely solve these violations while removing all redundant inference steps from the \( \beta \)-proofs. For this, an intrinsic relation between \( \beta \)-proofs and the partially constructed sequent proofs will be established. More precisely, the inference steps in a \( \beta \)-proof encode the smallest set of proof steps that are required for all possible completions of the partial sequent proof. We show how this process “approximates” \( \beta \)-proofs towards sequent proofs in order to guide a search-free proof reconstruction.

Using \( \beta \)-proofs within the proof reconstruction approach provides us with a new dimension of uniformity. In addition to the dimension of logic-independent uniformity, our representation framework does neither depend on a particular proof search strategy nor on the proof reconstruction method itself. Thus, \( \beta \)-proofs realize a general interface for building efficient proof reconstruction components and combine them with a variety of concrete proof search procedures.

Section 3 gives a brief summary of matrix characterizations and proof reconstruction in non-classical logics. Section 4 summarizes the requirements for redundancy deletion during proof reconstruction. In Section 5, \( \beta \)-proofs as representation framework are presented. We develop the “approximation” operations and illustrate the integration into the reconstruction process. Section 6 discusses complexity issues and completeness of the refined proof reconstruction method.

## 2 Preliminaries

Matrix characterizations of logical validity were introduced for classical logic \([8]\) and later extended to intuitionistic and modal logics \([30]\), and fragments of linear logic \([12, 19]\). For the purpose of this paper, it suffices to consider the class \( \mathcal{L} \) of logics consisting of classical, intuitionistic, and modal logics as presented in \([30]\).