

Article

# Synthetic Tableaux with Unrestricted Cut for First-Order Theories

Dorota Leszczyńska-Jasion \*  and Szymon Chlebowski \*

Department of Logic and Cognitive Science, Faculty of Psychology and Cognitive Science, Adam Mickiewicz University, ul. Szamarzewskiego 89a, 60-568 Poznań, Poland

\* Correspondence: Dorota.Leszczyńska@amu.edu.pl (D.L.-J.); Szymon.Chlebowski@amu.edu.pl (S.C.)

Received: 14 August 2019; Accepted: 15 November 2019; Published: 29 November 2019



**Abstract:** The method of synthetic tableaux is a cut-based tableau system with synthesizing rules introducing complex formulas. In this paper, we present the method of synthetic tableaux for Classical First-Order Logic, and we propose a strategy of extending the system to first-order theories axiomatized by *universal axioms*. The strategy was inspired by the works of Negri and von Plato. We illustrate the strategy with two examples: synthetic tableaux systems for identity and for partial order.

**Keywords:** synthetic tableaux; principle of bivalence; cut; first-order theory; universal axiom

---

## Cut? Don't eliminate, introduce!

Gentzen's *Hauptsatz* is rightly considered to be a milestone in the development of structural proof theory. For decades, it was thought that cut-elimination, yielding analyticity of the system, is a goal per se. However, today, it is well-known that eliminating cuts frequently increases the size and length of proofs. One can find examples showing that, in the worst case, cut-elimination produces non-elementarily larger and longer proofs [1–3]. For this reason, techniques of cut-introduction are being studied in order to shorten proofs (see [4,5]). In [4], it is shown that the technique of atomic cut-introduction is able to provide an exponential compression in the length of proofs. In [5], the authors studied the introduction of non-atomic formulas by the rule of cut.

Minimizing proofs is not the only reason to study proof systems with the rule of cut. Cut-formulas represent lemmatas, introduced to a proof in order to improve its structure or to bring in a new concept. It makes the proof more legible for a human. Instead of introducing cut to cut-free proofs (as is done in the above-mentioned papers), it may be useful, or perhaps more natural, to study proofs constructed within a cut-based system, i.e., a system in which the rule of cut cannot be eliminated. This approach is present in [6,7], where the authors introduced sequent calculi which are cut-based. The calculi are actually sequent-variants of tableau system **KE**.

## What is the method of synthetic tableaux?

This paper presents the system of synthetic tableaux for First-Order Logic, which is an extension of the method for Classical Propositional Logic presented by Urbański [8] and Urbański [9], but the inspiration for the first-order version comes from D'Agostino [10] and Mondadori [11]. The method was explored for some cases of propositional logics by Urbański [12] and Urbański [13]; however, an extension to the first-order level substituted a challenging task. On the propositional level, the closest proof-theoretical relative of the method of synthetic tableaux seems to be the calculus **KI**, which is an “inversion” of **KE**

(see [10,11] for **KI** and [14,15] for **KE**). However, the calculus for First-Order Logic presented here differs substantially from the version of **KI** for First-Order Logic.

The system of synthetic tableaux presented here is equipped with the so-called rule of the *Principle of Bivalence*, which is a form of cut. This rule is not eliminable from the system. In the case of system **KE**, non-eliminable cut is the remedy the authors propose for the computational collapse of analytic tableau system (in fact, of any proof system deprived of a representation of the principle of bivalence, see [14,15]). The situation is exactly analogous in the case of the system of synthetic tableaux for the propositional level. However, as may be expected, the situation complicates on the level of first-order; we demonstrate by examples the problems with restricting the applications of cut to analytic applications. Possible solutions to this problem will be examined in the future.

In Section 1, we describe the method of synthetic tableaux for the propositional case. In Section 2, we make a reference to system **KI**, which is an inversion of the more famous **KE**. The completeness proof presented in Section 3 was inspired by the completeness proofs of **KI** and **KE** with respect to an axiomatic system. In Section 4, we sketch some results concerning relative complexity of proof systems. The results motivate our research on proof systems like synthetic tableaux. In Sections 5–7, we describe the system of synthetic tableaux for the First-Order Logic, together with the proofs of soundness and completeness. Finally, Section 8 presents our strategy of extending the synthetic tableaux to first-order theories.

## 1. The Method of Synthetic Tableaux for CPL

Below, we present the synthetic tableaux system (ST-system, for short) for Classical Propositional Logic (CPL, for short). We describe the rules of tableau construction and define the notion of proof. In the case of CPL, there is exactly one binary branching rule and a collection of linear rules called *synthetic* or *synthesizing* since they build complex formulas from their subformulas or from their negations. As the reader shall see, there is a clear analogy between the synthesizing rules and natural-deduction rules introducing a connective, or sequent-calculus rules.

We use  $\mathcal{L}_{\text{CPL}}$  for the language of CPL with  $\neg, \rightarrow, \vee, \wedge$ . By  $A, B, C, F$ , we refer to formulas of  $\mathcal{L}_{\text{CPL}}$ .  $\mathcal{VAR}$  is for the set of propositional variables,  $\text{Sub}(A)$  for the set of all subformulas of formula  $A$ , understood in the usual manner. By  $\neg\text{Sub}(A)$ , we mean the set of negations of the subformulas of  $A$ , that is,  $\neg\text{Sub}(A) = \{\neg F : F \in \text{Sub}(A)\}$ .

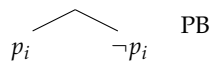
The number of linear rules depends on the number of logical connectives in the language; in the account presented in [16], which we follow here, there is 10 such rules (displayed below in Table 1).

**Table 1.** Linear synthesizing rules of the ST-system for CPL.

$$\begin{array}{cccccc} \frac{\neg B}{B \rightarrow C} \mathbf{r}_{\rightarrow}^1 & \frac{C}{B \rightarrow C} \mathbf{r}_{\rightarrow}^2 & \frac{B}{\neg C} \mathbf{r}_{\rightarrow}^3 & \frac{B}{B \vee C} \mathbf{r}_{\vee}^1 & \frac{C}{B \vee C} \mathbf{r}_{\vee}^2 & \frac{\neg B}{\neg C} \mathbf{r}_{\vee}^3 \\ \frac{\neg B}{\neg(B \wedge C)} \mathbf{r}_{\wedge}^1 & \frac{\neg C}{\neg(B \wedge C)} \mathbf{r}_{\wedge}^2 & \frac{B}{C} \mathbf{r}_{\wedge}^3 & \frac{B}{\neg \neg B} \mathbf{r}_{\neg} \end{array}$$

The premises of  $\mathbf{r}_{\rightarrow}^3$ ,  $\mathbf{r}_{\vee}^3$ , and  $\mathbf{r}_{\wedge}^3$  in Table 1 may occur in any order. If one wonders where, for example, this “ $C$ ” in  $\mathbf{r}_{\rightarrow}^1$  comes from, the following proviso comes to the rescue. A linear rule may be applied in the construction of a synthetic tableau for formula  $A$  provided that each premise and conclusion of the rule belongs to the set  $\text{Sub}(A) \cup \neg\text{Sub}(A)$ . Thus, in the case of rule  $\mathbf{r}_{\rightarrow}^1$ , any  $C$  such that  $B \rightarrow C$  is in  $\text{Sub}(A) \cup \neg\text{Sub}(A)$  is fine.

The branching rule is simply the rule of cut on literals. Following the insight of D’Agostino (see [14,15]) we call it the “PB-rule”, or simply “PB”, from the *Principle of Bivalence*.



Further, we say that the branching is *performed on*  $p_i$  or that the PB-rule was applied *with respect to*  $p_i$ . As in the case of the synthetic rules, applications of the PB-rule are subject to a restriction: the PB-rule may be applied with respect to  $p_i$  in the construction of a synthetic tableau for formula  $A$  provided that  $p_i \in \text{Sub}(A)$ . As a matter of fact, the restriction is built in the definition of a synthetic tableau (see Definition 1). With this restriction, the branching rule permits only analytic atomic cuts (for the notion of analytic cut, see [17–19]).

In the original account (see [8,16]), Mariusz Urbański defined the notion of *synthetic inference*, which is a sequence of formulas regulated by the above rules in a well-defined manner. Then, the notion of proof comes as a family of such interconnected sequences. The family-of-sequence account was motivated by the close relationship between synthetic tableaux and the so-called erotetic search scenarios, which were also defined as families of sequences, the so-called *erotetic derivations*. (*Erotetic search scenario* is a concept defined on the grounds of the logic of questions called *Inferential Erotetic Logic*. The reader can find more information about the relationship in [16]. The books by Urbański [9] and Urbański [20] contain broad exposition of the matter but are written in Polish. For erotetic search scenarios, see [21], and, for Inferential Erotetic Logic, the best recommendation is [22].) However, this is out of proof-theoretical tradition, where proofs are usually trees or sequences, and nowadays erotetic search scenarios may also be defined as trees (see [22,23]) so there is no need to stand by the sequence format. Below, we adopt the more common account of trees.

**Trees and tableaux.** We assume the proof-theoretic account of trees as partially ordered sets. *Branch* of a tree is its subset, which is a chain maximal with respect to inclusion. Below, in the proof of Lemma 2, we use the notion of *size of a finite tree*, which is the number of nodes of the tree.

As is practiced in structural proof theory, tableaux (derivations, proofs) are defined as trees labeled with formulas (or sequents, see ([24], p. 8), compare also [25]). Thus a (synthetic) tableau comes as a labeled tree  $\langle X, R, \ell \rangle$ , where  $\ell$  is a function assigning formulas to nodes of  $\langle X, R \rangle$ . If  $\mathcal{T} = \langle X, R, \ell \rangle$  is a labeled tree, then by  $r_{\mathcal{T}}$  we mean the *root of*  $\mathcal{T}$ . The notion of proof in the ST-system for CPL is given by the following definitions.

**Definition 1.** A synthetic tableau for formula  $A$  is a finite labeled tree

$$\mathcal{T} = \langle X, R, \ell \rangle$$

generated by the rules:  $r_{\rightarrow}^1, r_{\rightarrow}^2, r_{\rightarrow}^3, r_{\vee}^1, r_{\vee}^2, r_{\vee}^3, r_{\wedge}^1, r_{\wedge}^2, r_{\wedge}^3, r_{\neg}, \text{PB}$ , and such that

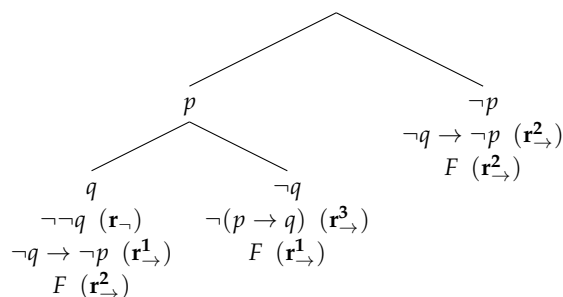
$$\ell : X \setminus \{r_{\mathcal{T}}\} \longrightarrow \text{Sub}(A) \cup \neg \text{Sub}(A)$$

and each leaf is labeled either with  $A$  or with  $\neg A$ .

**Definition 2.** A proof of  $A$  in the ST-system is a synthetic tableau  $\mathcal{T}$  for  $A$  such that each leaf of  $\mathcal{T}$  is labeled with  $A$ .

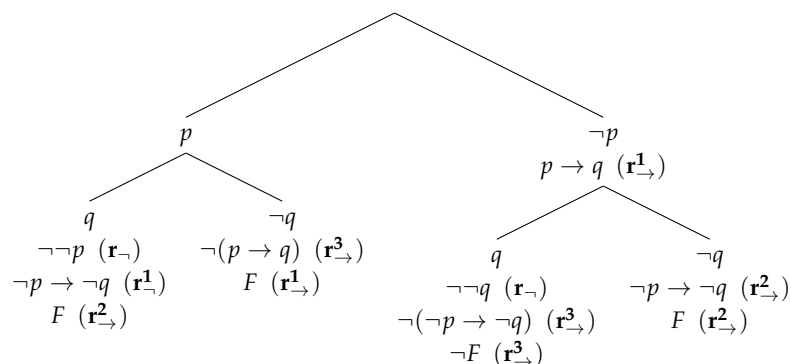
Here are two examples of synthetic tableaux. The first one is a synthetic tableau for formula  $(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$  and is a proof of the formula in the ST-system. If a formula was obtained by a linear rule, we indicate the rule to the right.

**Example 1.** Let  $F = (p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$ .



The second example is a synthetic tableau for  $(p \rightarrow q) \rightarrow (\neg p \rightarrow \neg q)$  and is not its proof, as the third (from the left) branch of the tableau ends with the negation of this formula.

**Example 2.** Let  $F = (p \rightarrow q) \rightarrow (\neg p \rightarrow \neg q)$ .



The above synthetic tableaux start with an application of the PB-rule. This is a general feature of the system—since PB is the only no-premise rule, it must be the one starting the construction of a synthetic tableau. Consequently, each synthetic tableau has more than one branch.

The system is sound and complete with respect to standard semantics for CPL; the most detailed proofs of these facts may be found in [9] (in Polish). Soundness is proved indirectly by the use of the concept of *minimal error point*. The idea is that, if  $A$  is not valid although it has a proof, then there must be an element in the tableau, which “introduces error” into the structure. We search for the element trying to establish the highest point in the structure of the tableau where the error appears. Then, a contradiction is derived, as it occurs that every error must have some erroneous predecessor. Completeness is proved by establishing a procedure of construction of a special kind of synthetic tableau, called *canonical synthetic tableau*. Canonical synthetic tableau for formula  $A$  starts with successive branchings on all propositional variables of  $A$ . If the number of distinct variables of  $A$  is  $k$ , then the number of branches of the canonical synthetic tableau for  $A$  is  $2^k$ . In the margin, this result shows that the ST-system for CPL is a *standard proof system* in the sense of D’Agostino and Mondadori (see, e.g., [14]), which is not at all surprising, as synthetic tableaux for CPL in the canonical version constitute a formal representation of the familiar truth-tables method.

In the case of ST-system for CPL, soundness and completeness may be proved by quite simple techniques using, e.g., Hintikka sets and Hintikka’s Lemma. However, since we aim at FOL, we use some more general tools. Completeness is proved with respect to axiomatic account, i.e., by simulating *Modus Ponens*—this idea is taken from the works by D’Agostino and Mondadori, which we briefly refer in

the next section. The proof of soundness is inspired by the use of *semantic trees* in the proofs for resolution systems. Here, we rely on the version of this technique presented in [26].

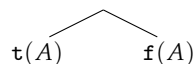
There is one important difference between the version presented here and the original one by Urbański; however, the difference does not influence the metalogical properties mentioned above. In the case of the original system, the definition of synthetic inference warrants that, if  $p_i$  is its term, then  $\neg p_i$  is not. On the level of a synthetic tableau, it means that the branching rule (our PB-rule) is never applied on the same branch more than once with respect to the same propositional variable. This condition warrants consistency of each synthetic inference. In the account presented here, this condition is neglected as it proved to be a hindrance in designing the ST-system for FOL. There has been an attempt to generalize the ST-system to the first-order case in a way which saves the property of consistent branch. The outcomes are presented in the research report [27]; however, the basic metalogical results—soundness and completeness—are missing.

## 2. System KI for CPL

Before synthetic tableaux were independently designed by Urbański, a system similar in spirit, called **KI**, was considered by D’Agostino and Mondadori. Interestingly, both synthetic tableaux and **KI** were to some extent inspired by Kalmár’s work (see [28]). Even the notion of *synthetic* rule occurs both in [10] and in Urbański’s work.

**KI** is a system which satisfies Prawitz’ *inversion principle* with respect to a much better known system **KE**. The latter was developed by Marco Mondadori in the late 1980s, and analyzed carefully by D’Agostino and Mondadori [14,15]. Information about system **KI** can be found in Section 3.7 of [10] and in [11].

System **KI** for CPL is expressed in a language with truth signs ([10], the unsigned version is also considered.) It contains introduction rules and the following version of the PB-rule:



which is *not* restricted to propositional variables. As mentioned above, “PB” is for the *Principle of Bivalence*, as the rule clearly embodies the idea that  $A$  is either true or false. When one accepts *arbitrary* formulas to be introduced by the PB-rule, one must also accept inconsistencies on branches. This is the price to be paid for the unrestricted use of cut. One of the foundational ideas of the method of synthetic tableaux by Urbański was that they formalize reasoning in which the final conclusion is derived from all the possible consistent sets of atoms that build it (this is the Kalmár’s inspiration). Hence, the restriction of the PB-rule to syntactical atoms gains an additional justification, irrespective of efficiency of this kind of system.

On the other hand, if inconsistent branches are the price to be paid for unrestricted (and possibly more efficient) use of cut, it is the price we can bear, especially when we realize that *the price for consistent branches* is extremely high. The problem is that it is highly improbable to describe in the framework of a restricted ST-system any logic which is not both propositional and extensional. Apparently, any finitely valued logic that may be characterized by finite matrices, *and only such logics*, may be successfully described in this framework. It explains why the only successful attempts to describe ST-systems where the cases of CPL, three-valued extensional L3 (see [12]) and paraconsistent CLuN, which may be fully characterized by semi-valuations (see [13]). Consistency of every branch is rather the property of these logics than a desired property of ST-systems.

Going back to **KI**, since the system permits inconsistent branches, it needs the notions of an open and a closed branch. A branch is called *closed* if, for some formula  $F$ , it contains both “ $t(F)$ ” and “ $f(F)$ ”, otherwise it is called *open*.

The introduction rules of **KI** for CPL are displayed in Table 2. We quote the names of the rules after [10].

**Table 2.** Introduction rules of **KI**.

$\frac{f(B)}{t(B \rightarrow C)} \text{It} \rightarrow 1$	$\frac{t(B)}{t(B \vee C)} \text{It} \vee 1$	$\frac{f(B)}{f(B \wedge C)} \text{If} \wedge 1$	$\frac{t(B)}{f(\neg B)} \text{If} \neg$
$\frac{t(C)}{t(B \rightarrow C)} \text{It} \rightarrow 2$	$\frac{t(C)}{t(B \vee C)} \text{It} \vee 2$	$\frac{f(C)}{f(B \wedge C)} \text{If} \wedge 2$	$\frac{f(B)}{t(\neg B)} \text{It} \neg$
$\frac{t(B)}{f(B \rightarrow C)} \text{If} \rightarrow$	$\frac{f(B)}{f(B \vee C)} \text{If} \vee$	$\frac{t(B)}{t(B \wedge C)} \text{It} \wedge$	

As the reader can see in Table 2, the introduction rules of **KI** for CPL are signed versions of the linear (synthetic) rules of ST-system for CPL. In [11], the author considered system called *canonical restriction* of **KI**, where the use of the PB-rule is restricted to atoms and a tree for  $A$  is built from its subformulas. Thus, the canonical restriction of **KI** fully corresponds to ST-system by Urbański, modulo truth signs. (More specifically, it is easy to see that the two systems for CPL, namely the canonical version of **KI** and the ST-system of Urbański, polynomially simulate each other. We say more about this issue in Section 4. For the notion of  $p$ -simulation, see [10].)

The notion of proof is introduced as follows. Let  $S$  be a finite set of signed formulas.  $S$  can be empty. A **KI**-tree for  $S$  is an expansion tree regulated by the rules of **KI**, starting from the elements of  $S$ . When  $S$  is empty, the origin of the tree is labeled with  $\emptyset$ .

Now, let  $\Gamma$  be a set of formulas (without the truth signs) and let  $A$  be a formula (with no truth sign). A **KI**-proof of  $A$  from  $\Gamma$  is a **KI**-tree for  $\{t(B) : B \in \Gamma\}$ , such that  $t(A)$  occurs in every open branch. Finally,  $A$  is a **KI**-theorem, symbolically  $\vdash_{\mathbf{KI}} A$ , if  $A$  is provable from the empty set of formulas.

Completeness of this system has been proved by the authors with respect to the axiomatic system for CPL. This has inspired us to use the same technique in proving completeness of the first-order version. It is also worth mentioning that completeness of both **KI** and **KE** may be proved exactly by the same argument.

### 3. Completeness Proof with Respect to the Axiomatic Account of CPL

Here, we present how the completeness-proof strategy works for the case of **KI** and CPL in order to use the same pattern in the completeness proof of the ST-system for FOL (see the next section).

The axiom schemes and the rule of *Modus Ponens* (*MP*, for short) presented in Table 3 constitute the axiomatic proof system  $\mathcal{F}$  for CPL. In the presentation, we rely on the conventions introduced in [29].

**Table 3.** Axiom schemes of  $\mathcal{F}$ .

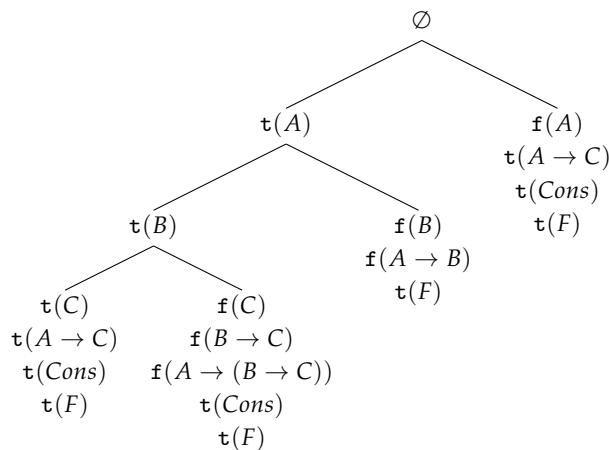
- |                                                                                                      |                                                                                |
|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| 1. $A \rightarrow (B \rightarrow A)$                                                                 | 6. $(A \rightarrow B) \rightarrow ((A \rightarrow \neg B) \rightarrow \neg A)$ |
| 2. $(A \rightarrow B) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))$ | 7. $\neg \neg A \rightarrow A$                                                 |
| 3. $A \rightarrow A \vee B$                                                                          | 8. $A \wedge B \rightarrow A$                                                  |
| 4. $B \rightarrow A \vee B$                                                                          | 9. $A \wedge B \rightarrow B$                                                  |
| 5. $(A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow (A \vee B \rightarrow C))$          | 10. $A \rightarrow (B \rightarrow A \wedge B)$                                 |

$$\frac{A \rightarrow B \quad A}{B} \text{MP}$$

The proof of completeness requires a demonstration that every axiom scheme has its scheme of proof. As an example, here is a scheme of **KI**-proof of Axiom 2. For conciseness, let:

$$F = (A \rightarrow B) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))$$

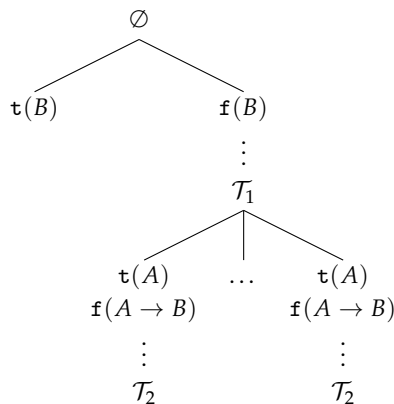
$$Cons = (A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C)$$



All the other axioms of  $\mathcal{F}$  may be easily proved in **KI**. The next theorem states that **KI** simulates the only rule of  $\mathcal{F}$ .

**Theorem 1.** If  $\vdash_{\mathbf{KI}} A$  and  $\vdash_{\mathbf{KI}} A \rightarrow B$ , then  $\vdash_{\mathbf{KI}} B$ .

**Proof.** (After [10,11].) Suppose that  $\vdash_{\mathbf{KI}} A$  and  $\vdash_{\mathbf{KI}} A \rightarrow B$ , and let  $\mathcal{T}_1$  and  $\mathcal{T}_2$  stand for the proofs of, respectively, formulas  $A$  and  $A \rightarrow B$ . Then, the following tree:



is a **KI**-proof of  $B$ , since each branch, except for the leftmost one, goes through  $f(A \rightarrow B)$  and ends with  $t(A \rightarrow B)$ , thus the leftmost branch is the only open branch of the tree.  $\square$

As mentioned above, this strategy of completeness proof requires unrestricted usage of the PB-rule in the system. That would be a drawback of the system obviously, but we truly adore the Gentzen-like spirit of the proof. We believe that the proof reveals important connections between axiomatic systems, Gentzen sequent systems with cut, and **KI** or ST-systems but with unrestricted cut. We elaborate on this topic in the next section.



#### 4. Synthetic Tableaux and Other Deductive Systems for CPL: A Note on Relative Complexity

There are another reasons to consider axiomatic systems, Gentzen sequent systems with cut, **KI**, and ST-system with unrestricted cut, as belonging to one family of deductive systems. Below, we cite the results established in [10,11], concerning the relative complexity of deductive systems for CPL. However, it is not the aim of this paper to describe the formal tools and report all the details needed to establish the quoted results. All this may be found in [30,31] (for the background), [10,11]. Our aim here is to indicate the context in which the method of synthetic tableaux appears especially attractive.

That a proof system  $S'$  *linearly/polynomially simulates* another proof system  $S$  means, informally, that there is a function computable in linear/polynomial time which maps every proof of a formula  $A$  in  $S$  to a proof of the same formula in  $S'$ .

Systems **KE** and **KI** (for CPL) can linearly simulate each other. We also know (see [11]) that the canonical restriction of **KI** (that is, **KI** with the use of the PB-rule restricted to atoms) can linearly simulate truth-tables, but not vice versa. Hence, canonical **KI**, as well as ST-system for CPL, are actually *improvements* on truth-tables in terms of systems complexity. On the other hand, analytic restriction of **KI**, where the PB-rule is restricted to subformulas of the formula to be proved, but is not restricted to atoms, can polynomially simulate both truth-tables and the analytic tableau system (see [32]) but, again, not vice versa. Finally, canonical **KI**, and thus also ST-system for CPL, cannot polynomially simulate the analytic tableau system. Here is what we lose when cut is restricted to atoms.

Again, let  $S$  and  $S'$  stand for propositional proof systems. Following D'Agostino [10], we write:  $S' \leq_p S$  to mean that  $S$  polynomially simulates (p-simulates)  $S'$ . The inscription  $S' \equiv_p S$  means that the relation of p-simulation holds in both directions, and  $S' <_p S$  means that  $S$  p-simulates  $S'$  but not vice versa. The following holds (let us recall that the systems are considered as pertaining to CPL):

1. Gentzen system with cut  $\equiv_p$  Natural Deduction  $\equiv_p$  Frege systems
2. Resolution  $<_p$  any system from (1)
3. Cut-free Gentzen system  $<_p$  any system from (1)

It can also be shown that **KE**, and thus also **KI**, can linearly simulate Natural Deduction.

To sum up, it is clear that the presence of a version of cut increases efficiency of proof systems. All the above pertains to systems for CPL; however, we believe that the relation between ST-systems, especially with cut unrestricted, and axiomatic systems is worthy of further research. The use of unrestricted cut can be beneficial if the system was used to support automated deduction with FOL. Clearly, the issue needs further research.

#### 5. The First-Order Case

The presentation of syntax, semantics, and axiomatic account of FOL is based on the conventions introduced in [29].

FOL is expressed in language called  $\mathcal{L}_{\text{FOL}}$ , containing:

- propositional connectives:  $\neg, \wedge, \vee, \rightarrow$ ;
- infinite set of variable symbols; we use  $x, y, z, \dots$  as metasymbols for variables;
- quantifiers  $(\exists x), (\forall x)$ ;
- function symbols of arbitrary arities;  $f, g, h, \dots$  are used as metasymbols, function symbols of arity 0 are called *constant symbols*; and
- relation symbols of arbitrary arities;  $P, Q, R, \dots$  are used as metasymbols.

The notions of *term*, *atomic formula*, and *formula* are defined in the usual manner, similarly for *free* and *bound occurrence of variable*, and *sentence*.  $\text{FORM}_{\mathcal{L}_{\text{FOL}}}$  is for the set of all formulas of the language.



We use  $t, s, \dots$  as metasympols for terms. The result of *substitution of term  $t$  for  $x$  in  $A$*  is denoted by  $A(t/x)$ , and defined in the usual manner; *int.al.*, whenever  $A(t/x)$  is considered, it is assumed that term  $t$  is free for  $x$  in  $A$ . After [29], we adopt the simplifying conventions for denoting substitutions, according to which:

- If “ $A(x)$ ” and “ $A(t)$ ” are written in the same context, this means that  $A = A(x)$  is a formula and that  $A(t)$  is  $A(t/x)$ ; using “ $A(x)$ ” neither presupposes that  $x$  occurs free in  $A$  nor that it occurs in  $A$  at all.
- If “ $A(t)$ ” and “ $A(s)$ ” are written in the same context, then this is to mean that  $A$  is a formula,  $x$  is a variable and  $A(t) = A(t/x)$ ,  $A(s) = A(s/x)$ .

For semantics of FOL, we use:

- $\mathcal{M} = \langle M, f^{\mathcal{M}} \rangle$  for an interpretation of  $\mathcal{L}_{\text{FOL}}$ , where  $M$  is the domain of  $\mathcal{M}$  and  $f^{\mathcal{M}}$  is the interpreting function; and
- $\sigma, \sigma^*$  for object assignments, that is, mappings from the set of variables to the domain  $M$  of  $\mathcal{M}$ .

We write “ $\mathcal{M} \models A[\sigma]$ ” for “formula  $A$  is satisfied in  $\mathcal{M}$  under  $\sigma$ ”, and “ $\mathcal{M} \models A$ ” for “ $A$  is true in  $\mathcal{M}$ ”.

### 5.1. Axiomatic System

Axiomatic proof system  $\mathcal{F}_{\text{FO}}$  for FOL contains the 10 axiom schemes listed in Table 3, and, additionally, the following two axiom schemes, in which  $A$  is any formula and  $t$  is any term free for  $x$  in  $A$ .

11.  $A(t) \rightarrow (\exists x)A(x)$ ; and
12.  $(\forall x)A(x) \rightarrow A(t)$ .

The set of rules of  $\mathcal{F}_{\text{FO}}$  contains *MP* and the following two quantifier rules of inference, where  $x$  does not appear freely in  $C$ ; “*GC*” is for the generalization over consequent and “*GA*” for the generalization over antecedent.

$$\frac{C \rightarrow A(x)}{C \rightarrow (\forall x)A(x)} \text{ GC} \quad \frac{A(x) \rightarrow C}{(\exists x)A(x) \rightarrow C} \text{ GA}$$

$\mathcal{F}_{\text{FO}}$  is sound and complete with respect to the model semantics referred to above (see [29]).

### 5.2. Synthetic-Tableaux System

The ST-system for FOL consists of the linear (synthesizing) rules, the PB-rule and the notion of proof. In this case, the PB-rule is not a subject to any restrictions, thus:

$$\begin{array}{c} \diagup \quad \diagdown \\ F \quad \neg F \end{array}$$

may be applied at any time, in a tableau constructed for a formula  $A$ , and  $F$  is arbitrary. This unrestricted form of cut (PB-rule) is necessary, as we have seen, to prove completeness of **KI** with respect to CPL in the “Gentzen-way”. In the ST-system for FOL, the rule is left unrestricted for the same reason. However, one can think of restrictions for practical applications. For example, it seems that there are no obstacles to restrict  $F$  to be an element of  $\text{Sub}(A)$ , but we do not consider this restriction here. Needless to say, no other counterpart of cut elimination, except for possible restrictions of applicability of the rule, is possible in the ST-system.

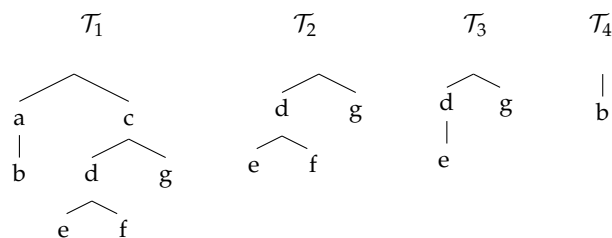
The linear rules of the system are those for propositional connectives, listed in Table 1, and the following two rules for existential quantification, where  $t$  is a term free for  $x$  in  $A$ :

$$\frac{A(t)}{(\exists x)A(x)} \text{ r}_{\exists} \quad \frac{\neg A(t)}{\neg(\forall x)A(x)} \text{ r}_{\neg\forall}$$

As in the propositional case, the applications of linear rules *could* be restricted to the set of suitably defined subformulas of the formula to be proved. The problem, however, lies in the notion of *subformula*. First, it must take into account substitutions, to the effect that, e.g.,  $A(t)$  is a subformula of  $(\exists x)A(x)$ . What is more, however, the problem gets more complicated in the case of the global rules introduced below. For this reason, we resign from the subformula restrictions in this version of the system.

For the next step, we need the notion of a subtableau of a tableau. Let  $R|_Y$  stand for the restriction of relation  $R$  to set  $Y$ . If  $\mathcal{T} = \langle X, R \rangle$  is a tree, then a *subtree* of  $\mathcal{T}$  is any  $\mathcal{S} = \langle Y, R|_Y \rangle$  such that: (i)  $Y \subseteq X$ ; (ii) if  $w$  is in  $Y$ , then each  $R$ -successor of  $w$  is in  $Y$ ; and (iii)  $\mathcal{S}$  has a root, that is,  $\mathcal{S}$  is itself a tree. If  $\mathcal{T}$ , together with a labeling function, is a synthetic tableau, then its subtrees, together with suitably restricted labeling functions, may be considered as *subtableaux* of  $\mathcal{T}$ ; however, we want the subtableaux to start with a branching, just as tableaux do. Moreover, a restriction of the labeling function must leave the root empty. The following example illustrates this idea.

**Example 3.** The second tree,  $\mathcal{T}_2$ , is a subtree of tree  $\mathcal{T}_1$ , and a subtableau of tableau  $\mathcal{T}_1$ . The third,  $\mathcal{T}_3$ , is not a subtree of  $\mathcal{T}_1$  since the node labeled with “f”, an  $R$ -successor of the node labeled with “d”, is missing. Finally,  $\mathcal{T}_4$  is a subtree of tree  $\mathcal{T}_1$ , but is not a subtableau of tableau  $\mathcal{T}_1$ , as it does not start with branching.



As in **KI**, we also need the following notion:

**Definition 3.** If for certain formula  $F$ , a branch of a labeled tree carries both  $F$  and  $\neg F$ , then we say that the branch is closed, otherwise we say that the branch is open.

Every linear rule of ST-system for FOL considered so far is *local* in that it acts on a *single* branch extending it with one node carrying the conclusion of the rule. The following rules for universal generalization are *global*, since they act on *each* open branch of a subtableau which, in particular, may be the whole tableau. Global rules are not necessary at the propositional level, but here we need it to incorporate universal generalization. For clarity, we present the rules in a descriptive form.

- UG1** If  $\mathcal{T}^*$  is a subtableau of  $\mathcal{T}$  such that every open branch of  $\mathcal{T}^*$  ends with formula  $C \rightarrow A(x)$ , where  $x$  does not occur freely in  $C$ , **and no formula in  $\mathcal{T}^*$  has been synthesized with the use of a premise which is not on  $\mathcal{T}^*$** , then  $\mathcal{T}$  may be extended by adding  $C \rightarrow (\forall x)A(x)$  to each open branch of  $\mathcal{T}^*$ .
- UG2** If  $\mathcal{T}^*$  is a subtableau of  $\mathcal{T}$  such that every open branch of  $\mathcal{T}^*$  ends with formula  $A(x) \rightarrow C$ , where  $x$  does not occur freely in  $C$ , **and no formula in  $\mathcal{T}^*$  has been synthesized with the use of a premise which is not on  $\mathcal{T}^*$** , then  $\mathcal{T}$  may be extended by adding  $(\exists x)A(x) \rightarrow C$  to each open branch of  $\mathcal{T}^*$ .

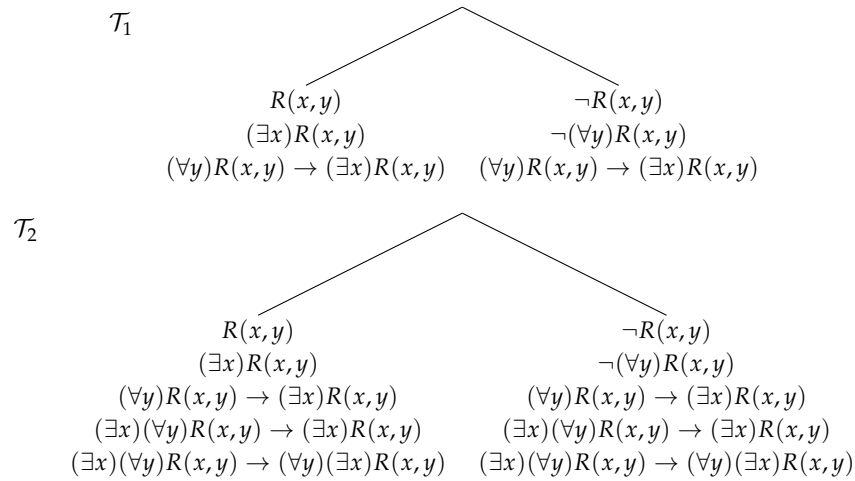
For preciseness, let us state the proviso written in bold in a more detailed way (we call it “the bold proviso”, although it is rather quite moderate):

**the bold proviso** if a formula lying on a branch of  $\mathcal{T}^*$  gets here by a local rule, then the premises necessary to derive it precede it on the same branch of  $\mathcal{T}^*$

Naturally, it can happen that a premise is present somewhere above the subtableau  $\mathcal{T}^*$ , but nevertheless the same formula must be “available” at the appropriate branch of  $\mathcal{T}^*$ .

Here is an example of a synthetic tableau where the global rules are applied. The first tree,  $\mathcal{T}_1$ , is obtained by local linear rules ( $\mathbf{r}_\exists$ ,  $\mathbf{r}_\supset^2$  on the left branch, and  $\mathbf{r}_{\neg\forall}$ ,  $\mathbf{r}_\supset^1$  on the right branch). Since  $x$  is not free in  $(\exists x)R(x, y)$ , rule **UG2** is applicable. One obtains a tree with formula  $(\exists x)(\forall y)R(x, y) \rightarrow (\exists x)R(x, y)$  in both leaves, to which **UG1** is applied. In this way, tree  $\mathcal{T}_2$  is obtained from  $\mathcal{T}_1$  by **UG2** and **UG1**. In both cases, the subtableau  $\mathcal{T}^*$  specified in the **bold proviso** is identical to the whole tableau, hence the proviso is satisfied.

**Example 4.**



The above tableau illustrates also the difficulty with restricting the applications of the rules of the system to subformulas of the formula to be proved. As mentioned above, the problem is in defining the very notion of subformula in the case of a language with quantifiers. Considering the above case, the definition would have to make formula  $(\forall y)R(x, y) \rightarrow (\exists x)R(x, y)$  a subformula of  $(\exists x)(\forall y)R(x, y) \rightarrow (\forall y)(\exists x)R(x, y)$ . Summarizing, the following definition abandons the restriction to subformulas. Consequently, it abandons the reference of a synthetic tableau to a particular formula  $A$ . However, the reference to a particular formula occurs in the notion of a proof.

**Definition 4.** A synthetic tableau in the ST-system for  $\mathcal{L}_{\text{FOL}}$  is a finite labeled tree  $\mathcal{T} = \langle X, R, \ell \rangle$  generated by: the unrestricted PB-rule, the local linear rules  $\mathbf{r}_\supset^1, \mathbf{r}_\supset^2, \mathbf{r}_\supset^3, \mathbf{r}_\forall^1, \mathbf{r}_\forall^2, \mathbf{r}_\forall^3, \mathbf{r}_\wedge^1, \mathbf{r}_\wedge^2, \mathbf{r}_\wedge^3, \mathbf{r}_\neg, \mathbf{r}_\exists, \mathbf{r}_{\neg\forall}$  and/or the global rules **UG1**, **UG2**, and such that:

$$\ell : X \setminus \{\mathbf{r}_\mathcal{T}\} \longrightarrow \text{FORM}_{\mathcal{L}_{\text{FOL}}}$$

**Definition 5.** A proof of  $A$  in the ST-system for FOL is a synthetic tableau  $\mathcal{T}$  for  $A$  such that each leaf of an open branch of  $\mathcal{T}$  is labeled with  $A$ .

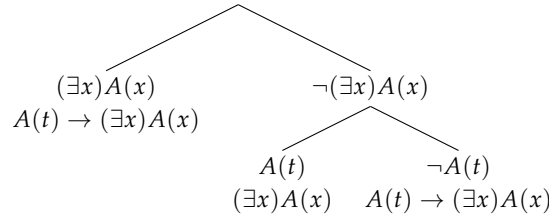
The first tree in Example 4 is a proof of formula  $(\forall y)R(x, y) \rightarrow (\exists x)R(x, y)$  in the ST-system, and the second is a proof of  $(\exists x)(\forall y)R(x, y) \rightarrow (\forall y)(\exists x)R(x, y)$  in the ST-system.

Finally, here is the announced proof of the completeness of the ST-system for FOL with respect to  $\mathcal{F}_{\text{FO}}$ .

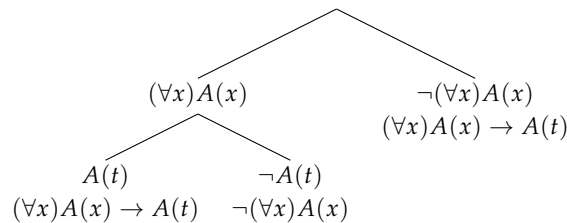
**Theorem 2.** Every axiom of  $\mathcal{F}_{\text{FO}}$  is provable in the ST-system for FOL.

**Proof.** As to axiom Schemes 1–10, one example is presented in Section 3. The other proofs are easy to find. Let us consider the quantifier cases.

For each axiom of the form  $A(t) \rightarrow (\exists x)A(x)$ , where  $t$  is free for  $x$  in  $A$ , there is the following tree:



where the application of  $R_{\exists}$  on the second branch is permitted because  $t$  is free for  $x$  in  $A$ . Similarly, the following tree:



constitutes a proof for an axiom of the form  $(\forall x)A(x) \rightarrow A(t)$ , where  $t$  is free for  $x$  in  $A$ .  $\square$

**Theorem 3.** The ST-system for FOL can simulate the inference rules of  $\mathcal{F}_{FO}$ . More specifically:

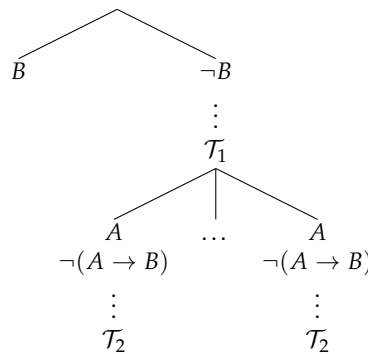
1. If a formula of the form  $C \rightarrow A(x)$ , where  $x$  is not free in  $C$ , has a proof in the ST-system for FOL, then  $C \rightarrow (\forall x)A(x)$  has it as well (rule GC).
2. If a formula of the form  $A(x) \rightarrow C$ , where  $x$  is not free in  $C$ , has a proof in the ST-system for FOL, then  $(\exists x)A(x) \rightarrow C$  has it as well (rule GA).
3. If  $A$  and  $A \rightarrow B$  have proofs in the ST-system for FOL, then there is also a proof of  $B$  (rule MP).

**Proof.** The result is obvious in the case of the rules of Universal Generalization—rules GC and GA are simulated by **UG1** and **UG2**, respectively. Suppose that a formula of the form  $C \rightarrow A(x)$ , where  $x$  is not free in  $C$ , has a proof  $\mathcal{T}$  in the ST-system for FOL. Since  $\mathcal{T}$  is its own subtableau, the “bold proviso” is satisfied. Thus, an application of **UG1** results in a proof of  $C \rightarrow (\forall x)A(x)$ . The reasoning is analogous in the case of rule GA.

For the case of *MP*, the proof of Theorem 1 may be repeated in the first-order setting. Thus, suppose that  $A$  and  $A \rightarrow B$  have proofs  $\mathcal{T}_1$  and  $\mathcal{T}_2$  (respectively) in the ST-system for FOL. Then, the tree displayed on Figure 1, page 13, is a proof of  $B$  in the ST-system for FOL, since the leftmost branch is the only open branch of the tree.  $\square$

Now, we can explain why the rules of universal generalization need the “bold proviso”. Consider the following simplified versions of **UG1** and **UG2**:

- UG1<sup>0</sup>** If  $\mathcal{T}$  is a proof of formula  $C \rightarrow A(x)$  in ST-system, where  $x$  does not occur freely in  $C$ , then each open branch of  $\mathcal{T}$  may be extended with  $C \rightarrow (\forall x)A(x)$ . The result is a proof of  $C \rightarrow (\forall x)A(x)$ .
- UG2<sup>0</sup>** If  $\mathcal{T}$  is a proof of formula  $A(x) \rightarrow C$  in ST-system, where  $x$  does not occur freely in  $C$ , then each open branch of  $\mathcal{T}$  may be extended with  $(\exists x)A(x) \rightarrow C$ . The result is a proof of  $(\exists x)A(x) \rightarrow C$ .



**Figure 1.** A simulation of MP.

The problem shows up when one tries to conduct the proof of Theorem 3 for MP. We cannot simply “glue” trees  $\mathcal{T}_1$ ,  $\mathcal{T}_2$  with the branch starting with  $\neg B$ , because the potential applications of  $\mathbf{UG1}^0$ ,  $\mathbf{UG2}^0$  in  $\mathcal{T}_1$ ,  $\mathcal{T}_2$  would miss their justification: with the left branch (with formula  $B$ ), the relevant tableaux would no longer be proofs of the respective premises  $C \rightarrow A(x)/A(x) \rightarrow C$ . Thus, the restriction concerning subtableaux is necessary to save our strategy of proving completeness via axiomatic system. It remains an open question whether the ST-system with  $\mathbf{UG1}^0$ ,  $\mathbf{UG2}^0$  instead of  $\mathbf{UG1}$ ,  $\mathbf{UG2}$  is complete.

**Theorem 4.** Suppose that  $A$  is a formula such that  $\mathcal{M} \models A$  for every interpretation  $\mathcal{M}$  of  $\mathcal{L}_{\text{FOL}}$ . Then,  $A$  has a proof in the ST-system for FOL.

**Proof.** Since  $A$  is valid, we know that  $A$  has a proof  $\mathcal{P}$  in  $\mathcal{F}_{\text{FO}}$ . By Theorems 2 and 3, and by induction on the length of  $\mathcal{P}$ ,  $A$  has a proof in the ST-system.  $\square$

### 5.3. Derivability of Universal Generalization

We have shown that the ST-system for FOL is complete with respect to  $\mathcal{F}_{\text{FO}}$ . However, it may seem more natural to use the more common form of universal generalisation:

$$\text{if } \vdash A(x), \text{ then } \vdash (\forall x)A(x)$$

instead of GC and GA. Consequently, one may consider a counterpart of universal generalization in the synthetic tableaux framework. Below, we show that such a rule is derivable.

**Theorem 5.** Let  $A(x)$  be a formula which is provable in the ST-system for FOL. Then,  $(\forall x)A(x)$  is also provable.

**Proof.** Let  $\mathcal{D}$  be an ST-proof of  $A(x)$ , and let  $d_1, \dots, d_n$  stand for the open branches of  $\mathcal{D}$  (by Definition 5, each open branch ends with the formula  $A(x)$ ). First, by using  $\mathbf{r}_{\rightarrow}^2$  with formula  $A(x)$  as the premise, we add  $(\exists x)A(x) \rightarrow A(x)$  to each  $d_i$  ( $1 \leq i \leq n$ ). Since  $x$  is not free in the antecedent, we transform the tree into a proof of  $(\exists x)A(x) \rightarrow (\forall x)A(x)$  by  $\mathbf{UG1}$ . The result is displayed on Figure 2. Finally, we extend each open branch of the tableau by means of the application of the following sequence of rules: PB-rule performed on  $(\forall x)A(x)$ , then on the right branch,  $(\exists x)A(x)$  is derived from  $A(x)$  by  $\mathbf{r}_{\exists}$ , with  $x$  as term  $t$  ( $x$  is free for  $x$  in  $A$ ). Then, the last formula is derived by  $\mathbf{r}_{\rightarrow}^3$ . Naturally, the branch is closed. The derivation is displayed on Figure 3.

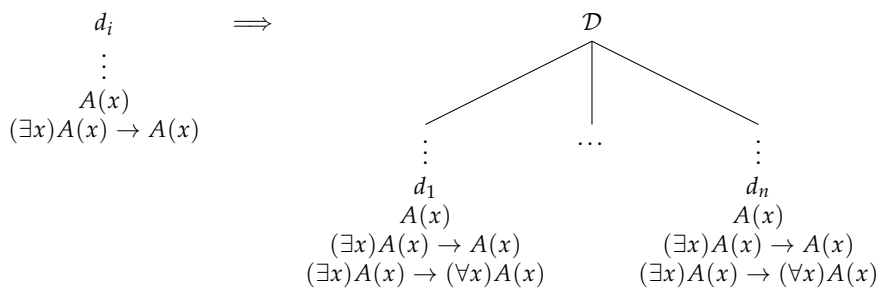


Figure 2. Universal Generalization, part 1.

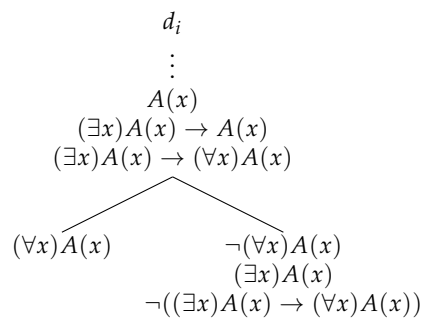


Figure 3. Universal Generalization, part 2.

The only possibly open branches in the modified tree are those going through  $d_i, A(x)$  and ending with  $(\forall x)A(x)$ . Thus, we have obtained an ST-proof of that formula.  $\square$

Theorem 5 shows that the following global rule is derivable in the ST-system for FOL:

**UG** If  $\mathcal{T}$  is a proof of formula  $A(x)$  in the ST-system for FOL, then each open branch of  $\mathcal{T}$  may be extended by adding  $(\forall x)A(x)$ . The result is a proof of  $(\forall x)A(x)$  in the system.

The rule is not a subject to any restriction akin to the bold proviso. Again, this follows from the strategy adopted at the metalevel, that is, from the fact that in the proof of Theorem 5 we act on the whole tableau of  $A(x)$ . We use this version of generalization in the formalization of first-order theories in Section 8.

#### 5.4. System **KI** for FOL

At the end of this section, let us refer briefly to the version of **KI** for FOL. The construction of the system is described in [11]. The author found the system by putting sequences of formulas into nodes of a **KI**-tree, which enables him to formulate restrictions on the rules introducing quantifiers on a branch. The rules presented there are local and they actually deal with quantifiers in a manner characteristic to analytic tableaux. For this reason, it is hard to compare the system of **KI** and the ST-system for FOL. Thus, we conclude with the modest remark that the main difference between the ST-system for FOL and **KI** for FOL lies in the treatment of quantifiers: **KI** treats quantifiers in a way characteristic to analytic tableau systems, whereas the ST-system deals with quantifiers by the use of global rules and unrestricted cut, which makes it more similar to natural-deduction or Gentzen systems.

### 6. Soundness of the ST System for FOL

Let us start with the following observation:

**Lemma 1.** Each local rule of the ST-system for FOL preserves the property of being satisfied under an object assignment  $\sigma$  in an interpretation  $\mathcal{M}$  of  $\mathcal{L}_{\text{FOL}}$ .

**Proof.** We consider  $\mathbf{r}_{\rightarrow}^1$  and  $R_{\exists}$ . The other cases are analogous.

Suppose that, for some interpretation  $\mathcal{M}$  and  $\sigma$  in  $\mathcal{M}$ ,  $\mathcal{M} \models \neg B[\sigma]$ . Then,  $\mathcal{M} \not\models B[\sigma]$  and thus  $\mathcal{M} \models A \rightarrow B[\sigma]$ . This argument shows that if the premise of rule  $\mathbf{r}_{\rightarrow}^1$  is satisfied under  $\sigma$  in  $\mathcal{M}$ , then the conclusion is satisfied as well.

Suppose that  $t$  is free for  $x$  in  $A$ , and that  $\mathcal{M} \models A(t)[\sigma]$ . Let  $u$  stand for the interpretation of term  $t$  in  $\mathcal{M}$  under  $\sigma$ . Let  $\sigma^*$  stand for an object assignment in  $\mathcal{M}$  such that  $\sigma^*(x) = u$  and  $\sigma^*(y) = \sigma(y)$  for every  $y \neq x$ . Then,  $\mathcal{M} \models A(x)[\sigma^*]$ . This means that  $\mathcal{M} \models (\exists x)A(x)[\sigma]$ .  $\square$

The following fact expresses the Principle of Bivalence with respect to the satisfaction relation in model-theoretic semantics.

**Fact 1.** Let  $\mathcal{M}$  and  $\sigma$  be arbitrary, and let  $F$  stand for an arbitrary formula.

$$\mathcal{M} \models F[\sigma] \text{ or } \mathcal{M} \models \neg F[\sigma].$$

The idea of the proof of Lemma 2 comes from using semantic trees in the proof of resolution systems (see [26], Section 3.8 for details).

We say that a branch  $\mathcal{B}$  of a synthetic tableau  $\mathcal{T}$  is *compatible with*  $\sigma$  in  $\mathcal{M}$  iff for every formula  $F$  that labels  $\mathcal{B}$ , it is the case that  $\mathcal{M} \models F[\sigma]$ . Let us observe that:

**Fact 2.** If a branch of a synthetic tableau is compatible with some object assignment  $\sigma$ , then the branch cannot be closed.

**Lemma 2.** Let  $\mathcal{T}$  be a synthetic tableau. Let  $\mathcal{M}$  be an arbitrary but fixed interpretation and let  $\sigma$  stand for an arbitrary but fixed object assignment in  $\mathcal{M}$ . There is a branch  $\mathcal{B}$  of  $\mathcal{T}$  which is compatible with  $\sigma$  in  $\mathcal{M}$ .

**Proof.** The proof is by induction on the size of  $\mathcal{T}$ , that is, the number of its nodes.

Base case. The smallest possible synthetic tableau is an empty root. (By Definition 4, a synthetic tableau is a finite tree, and a tree must have a root. Hence, it follows that a one-element tree, containing the root only, is the smallest possible synthetic tableau.) Since there is no label,  $\mathcal{T}$  is compatible with any assignment of an arbitrary model.

Induction step. Assume the following induction hypothesis: for synthetic tableaux of size up to  $n$ , for any  $\sigma$  of an arbitrary model  $\mathcal{M}$ , there is a branch in the tableau compatible with  $\sigma$ . Suppose that the size of  $\mathcal{T}$  is  $n + 1$ . We need to consider the last rule that acted upon the tableau.

Suppose that it was the PB-rule. Let  $\mathcal{T}^*$  stand for the tableau to which the rule was applied, and assume that  $\mathcal{B}^*$  is the particular branch that the rule acted upon.  $\mathcal{T}^*$  has  $n - 1$  nodes, thus, by the induction hypothesis, there is a branch  $\mathcal{B}$  in  $\mathcal{T}^*$  compatible with  $\sigma$  in  $\mathcal{M}$ . If the branch is not  $\mathcal{B}^*$ , then it is present also in the tableau  $\mathcal{T}$ , and thus it follows that there is a branch in  $\mathcal{T}$  compatible with  $\sigma$  in  $\mathcal{M}$ , as required. If  $\mathcal{B} = \mathcal{B}^*$ , then, by Fact 1, one of the two branches created from  $\mathcal{B}^*$  by the PB-rule is compatible with  $\sigma$  in  $\mathcal{M}$ .

Now, suppose that the last rule applied was a local linear rule. Then, we consider a tableau  $\mathcal{T}^*$  to which the rule was applied, we use the induction hypothesis, and—similarly as in the above case—we conclude that, if the branch of  $\mathcal{T}^*$  compatible with  $\sigma$  is the one modified by the local rule, then, by Lemma 1, the resulting branch of  $\mathcal{T}$  is compatible with  $\sigma$  in  $\mathcal{M}$ .

Finally, the difficult case. The last rule applied was one of the general rules; suppose that it was **UG1**, the reasoning is analogous in the case of **UG2**. Let  $\mathcal{T}^*$  be the synthetic tableau to which the rule was applied. The size of  $\mathcal{T}^*$  is not greater than  $n$ , thus, by the induction hypothesis, there is a branch  $\mathcal{B}^*$  in  $\mathcal{T}^*$



compatible with  $\sigma$  in  $\mathcal{M}$ . By Fact 2, the branch must be open, so it has a formula of the form  $C \rightarrow A(x)$  in its leaf, where  $x$  does not occur freely in  $C$ .

Suppose that  $\mathcal{M} \models C \rightarrow (\forall x)A(x) [\sigma]$ . Therefore,  $\mathcal{M} \models C [\sigma]$  and  $\mathcal{M} \models (\forall x)A(x) [\sigma]$ . Hence, it follows that there is an object  $u$  such that if  $\sigma^*$  is an object assignment in  $\mathcal{M}$  such that  $\sigma^*(x) = u$  and  $\sigma^*(y) = \sigma(y)$  for each  $y \neq x$ , then (i)  $\mathcal{M} \models A(x) [\sigma^*]$ . Since  $x$  does not occur freely in  $C$ , still (ii)  $\mathcal{M} \models C [\sigma^*]$ . However, since  $\mathcal{T}^*$  satisfies the induction hypothesis, it satisfies the hypothesis also with respect to object assignment  $\sigma^*$  in  $\mathcal{M}$ . Thus, there is an open branch of  $\mathcal{T}^*$  compatible with  $\sigma^*$ . Each open branch of  $\mathcal{T}^*$  ends with  $C \rightarrow A(x)$ , thus  $\mathcal{M} \models C \rightarrow A(x) [\sigma^*]$ . However, this contradicts the previous Arrangements (i) and (ii). Thus, it follows that  $\mathcal{M} \models C \rightarrow (\forall x)A(x) [\sigma]$ . Hence, the branch resulting from  $\mathcal{B}^*$  is compatible with  $\sigma$  in  $\mathcal{M}$ .  $\square$

**Theorem 6.** *If  $A$  has a proof in the ST-system for FOL, then  $A$  is valid.*

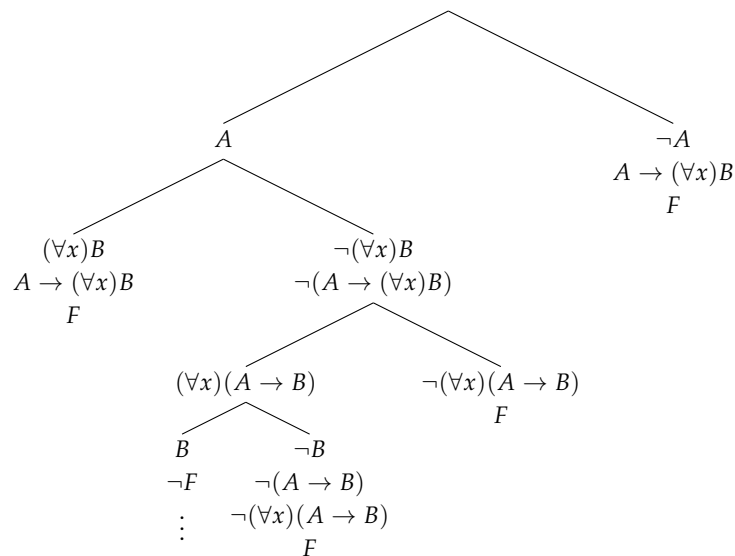
**Proof.** Suppose that  $\mathcal{T}$  is a proof of  $A$  and that  $A$  is not valid. Let  $\mathcal{M}$  be an arbitrary (but fixed) interpretation of  $\mathcal{L}_{FOL}$ , and let  $\sigma$  be an arbitrary (but fixed) object assignment in  $\mathcal{M}$  such that  $\mathcal{M} \not\models A [\sigma]$ . By the previous lemma, there is a branch  $\mathcal{B}$  of  $\mathcal{T}$  compatible with  $\sigma$ . The branch must be open, and since  $\mathcal{T}$  is a proof of  $A$ , the formula labels the leaf of  $\mathcal{B}$ . Thus,  $\mathcal{M} \models A [\sigma]$ , which is a contradiction.  $\square$

## 7. Some Further Remarks on Relations between the ST-System and the Axiomatic System

Despite finishing the completeness proof, for some time, the first author was not able to prove anything interesting in the system. Let us illustrate the difficulties with the following formula  $F$ , where  $x$  does not occur free in  $A$  (for simplicity, we omit the  $x$ s after  $A$  and  $B$ ):

$$F = (\forall x)(A \rightarrow B) \rightarrow (A \rightarrow (\forall x)B)$$

The first attempt to prove  $F$  was the following:



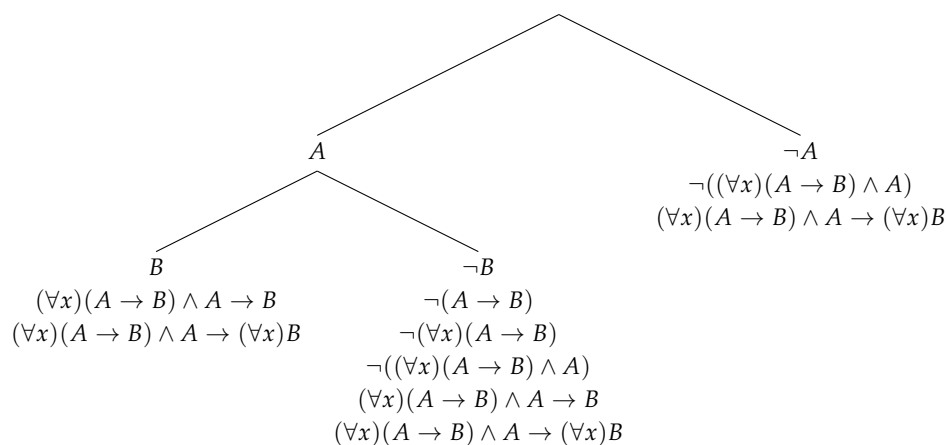
which is not satisfactory; the branch with  $\neg F$  should be closed, but it is not clear how to derive a contradiction. In a system of analytic tableaux, one would instantiate on  $\neg \forall x B$  introducing some  $\neg B(a)$ , but in this system  $\neg B(a)$  may only come by branching, and the problem then is with closing the left branch with  $B(a)$  on it. The first author overcame this difficulty after recalling a proof of the formula in axiomatic

system  $\mathcal{F}_{FO}$ :

- |    |                                                                                                                                                        |               |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| 1. | $(\forall x)(A \rightarrow B) \rightarrow (A \rightarrow B)$                                                                                           | Axiom 12      |
| 2. | $((\forall x)(A \rightarrow B) \rightarrow (A \rightarrow B)) \rightarrow ((\forall x)(A \rightarrow B) \wedge A \rightarrow B)$                       | Thesis of FOL |
| 3. | $(\forall x)(A \rightarrow B) \wedge A \rightarrow B$                                                                                                  | MP:2,1        |
| 4. | $(\forall x)(A \rightarrow B) \wedge A \rightarrow (\forall x)B$                                                                                       | GC:3          |
| 5. | $((\forall x)(A \rightarrow B) \wedge A \rightarrow (\forall x)B) \rightarrow ((\forall x)(A \rightarrow B) \rightarrow (A \rightarrow (\forall x)B))$ | Thesis of FOL |
| 6. | $(\forall x)(A \rightarrow B) \rightarrow (A \rightarrow (\forall x)B)$                                                                                | MP:5,4        |

In the second line of the proof, the converse of the exportation law is used in order to generalize on  $B$ . This is the point.

The following tree represents a proof of formula  $C = (\forall x)(A \rightarrow B) \wedge A \rightarrow (\forall x)B$ , where  $x$  does not occur free in  $A$ , in the ST-system. In the subtableau starting with  $A$ , the leaves are derived by **UG1**.



Finally, Figure 4 presents a proof of  $F$  where the problematic branch is closed by contradicting formula  $C$ . We need to use it, since without “importing”  $A$  to the antecedent we cannot generalize on  $B$ . Let us also explain that the fourth (from the left) branch contains a kind of a detour: formula  $(\forall x)(A \rightarrow B) \wedge A \rightarrow B$  is derived here to make **UG1** applicable in the subtableau starting with  $A$ . After deriving formula  $C$ , we need to extend the branch with  $F$  (obtained by  $\mathbf{r}_\rightarrow^1$ , which is a local rule), to make the tree a proof of  $F$ . The whole tableau is a good example illustrating the fact that the synthetic tableaux system is not a “tableau system” in the common sense of the term.

The interesting reflection is that about the relation between the axiomatic system and that of ST. The latter may seem bizarre, if an axiomatic proof is needed in order to get a hint on how to prove a formula. It appears that heuristics of proving theorems in axiomatic systems may serve also as heuristics of proving theorems in the ST-system. What is the benefit, if any? Does it work also the other way round, that is, is it the case that heuristics suitable for the ST-system can work for axiomatic systems as well?

At the moment, we are not able to answer this questions satisfactorily. However, the example with formula  $F$  sheds some light in this darkness. Although we need a hint to prove it, the use of the PB-rule is still restricted to subformulas (as mentioned above, the notion of subformula must take into account all substitutions) of formula  $F$ . The more complex formulas are built from the “atoms” introduced by the PB-rule. Albeit this “composition” derives us from the set of subformulas of  $F$ , it does so only via the export-import manipulation necessary to generalize on the consequent. This may be a matter of the particular form of rules for universal generalization in this ST-system.

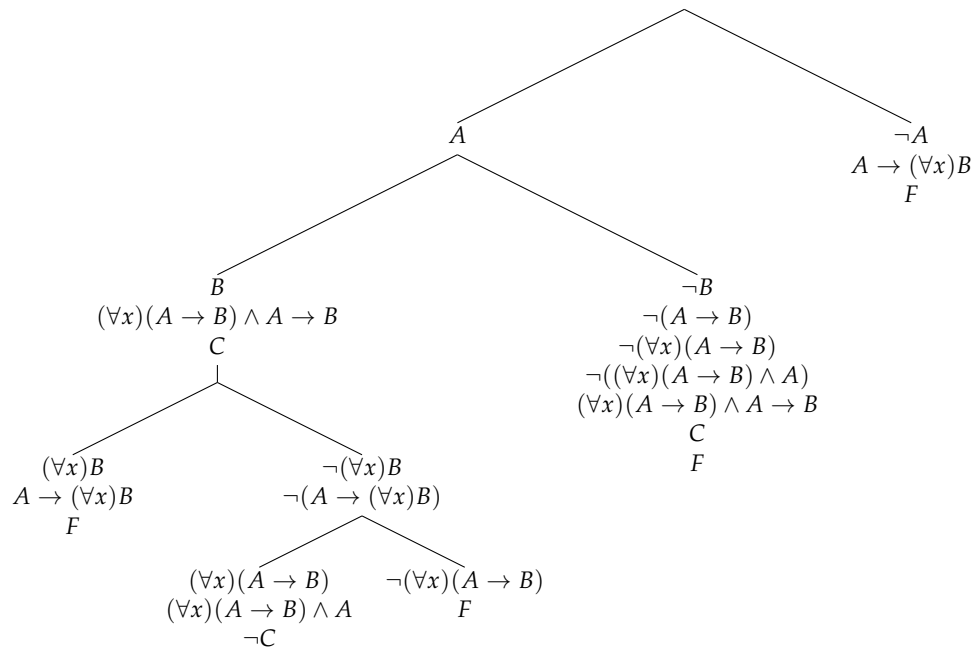


Figure 4. A proof of F.

## 8. ST-Systems for First-Order Theories

In [33], the authors presented a strategy of transforming axioms of certain forms into sequent calculus rules, in order to obtain a sequent calculus for a given first-order theory. In this section, we show that similar techniques may be applied beyond the domain of structural proof theory.

### 8.1. Universal Axioms

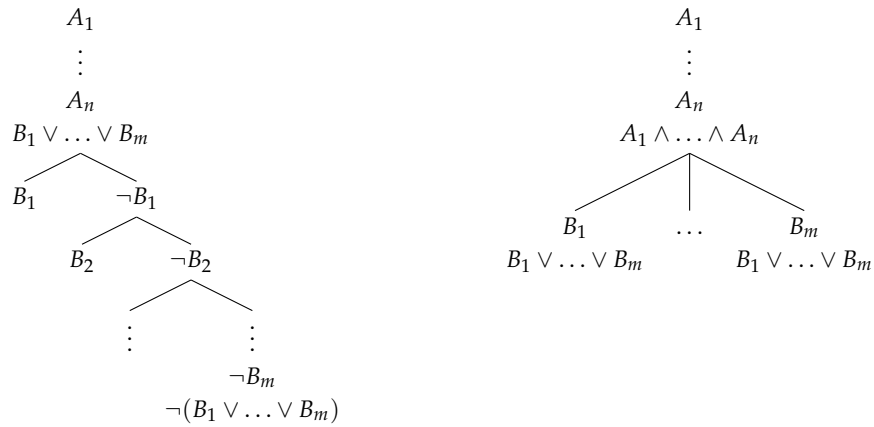
After Negri and von Plato [33], by *universal axioms* we mean sentences of  $\mathcal{L}_{\text{FOL}}$  of the following form:

$$(\forall x_1) \dots (\forall x_k)(A_1 \wedge \dots \wedge A_n \rightarrow B_1 \vee \dots \vee B_m) \quad (\text{ax})$$

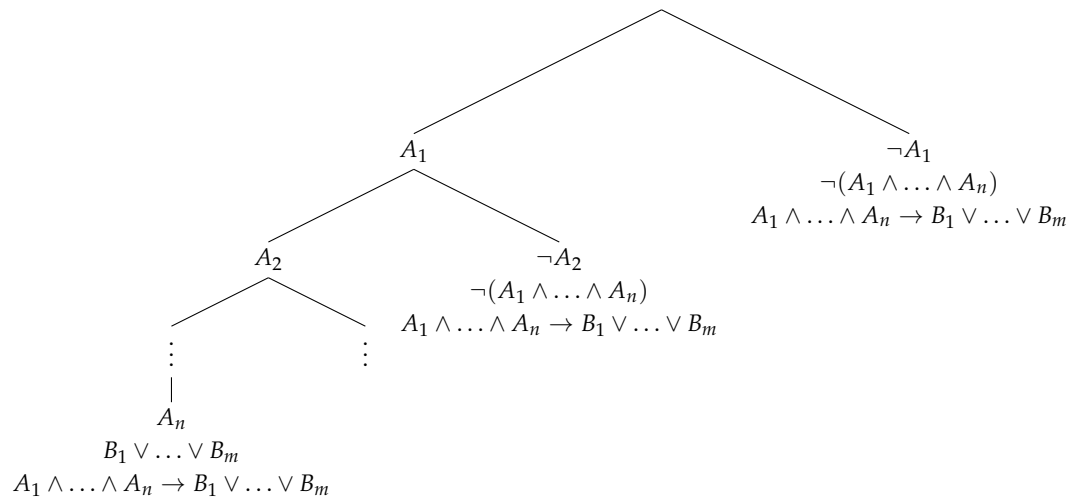
where  $A_i$  and  $B_j$  are *atomic formulas*. For simplicity, in this section, we assume that both conjunction and disjunction may have an arbitrary number of arguments. We show how the universal axioms can be converted into ST-rules. When  $n, m \geq 1$ , the rule has one of the forms indicated below.

$$\frac{A_1 \quad \vdots \quad A_n}{B_1 \vee \dots \vee B_m} (R_{ax}^1) \quad \frac{A_1 \quad \vdots \quad A_n}{B_1 \mid \dots \mid B_m} (R_{ax}^{1*}) \quad \frac{A_1 \wedge \dots \wedge A_n}{B_1 \vee \dots \vee B_m} (R_{ax}^2) \quad \frac{A_1 \wedge \dots \wedge A_n}{B_1 \mid \dots \mid B_m} (R_{ax}^{2*})$$

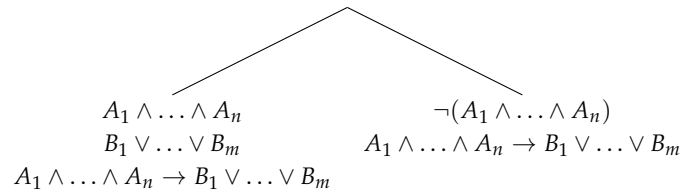
Rules  $(R_{ax}^{1*})$  and  $(R_{ax}^{2*})$  cause branching to  $m$  subtrees. If the local rules for  $\vee$  and  $\wedge$  are present in the system, then each of the four rules may be derived from any of the other three. We show two examples of such derivations below, the others are easy to find. On the left: rule  $(R_{ax}^1)$  is assumed, rule  $(R_{ax}^{1*})$  is derived. On the right: rule  $(R_{ax}^{2*})$  is assumed, rule  $(R_{ax}^1)$  is derived.



The general scheme of the proof of (ax) in an ST-system, assuming only atomic cuts and rule  $(R_{ax}^1)$ , looks as follows:



If we allow cut-formulas to be non-atomic, we can use the second rule and the proof can be simplified:



Further, for universal axioms (ax) with  $n = 0$  and  $m \geq 1$ , we obtain the following no-premises rules:

$$\frac{}{B_1 \vee \dots \vee B_m} (R_{ax}^{\mathcal{Q}1}) \quad \frac{}{B_1 \mid \dots \mid B_m} (R_{ax}^{\mathcal{Q}1*})$$

At each stage of a derivation, one can introduce the formula (in the case of  $(R_{ax}^{\mathcal{Q}1})$ ) or extend a tree by means of  $m$  subtrees (in the case of  $(R_{ax}^{\mathcal{Q}1*})$ ). As before, each of this rule is derivable from the other.

If  $n \geq 1$  and  $m = 0$ , the rules can be formulated as follows:

$$\frac{\begin{array}{c} A_1 \\ \vdots \\ A_n \\ \hline C \end{array} (R_{ax}^{\mathcal{O}2})}{\frac{A_1 \wedge \dots \wedge A_n}{C} (R_{ax}^{\mathcal{O}2*})}$$

$C$  is an arbitrary formula. If we had *Falsum* ( $\perp$ ) in the language, we could put  $C = \perp$ . Again, these rules are inter-derivable in the system. Below to the left, we assume  $(R_{ax}^{\mathcal{O}2*})$  and derive  $(R_{ax}^{\mathcal{O}2})$ ; and vice versa to the right, where  $F$  is for  $\neg(A_1 \wedge \dots \wedge A_n)$ :

$$\begin{array}{c} A_1 \\ \vdots \\ A_n \\ | \\ A_1 \wedge \dots \wedge A_n (r_{\wedge}^3) \\ \hline C (R_{ax}^{\mathcal{O}2*}) \end{array} \qquad \begin{array}{c} A_1 \wedge \dots \wedge A_n \\ \swarrow \quad \searrow \\ A_1 \quad \neg A_1 \\ \swarrow \quad \searrow \quad \quad F \\ A_2 \quad \neg A_2 \\ \swarrow \quad \searrow \quad \quad F \\ \vdots \quad \quad \quad \vdots \\ A_n \quad \quad \quad F \\ \hline C (R_{ax}^{\mathcal{O}2}) \end{array}$$

## 8.2. First Example: Identity

We show how to use the general approach in order to tackle the First-Order Logic with identity. First, we have to extend the language  $\mathcal{L}_{\text{FOL}}$  by means of a special predicate symbol,  $=$ , representing identity. Axiomatic system for FOL with identity,  $\mathcal{F}_{\text{FO}}^=$ , can be obtained from  $\mathcal{F}_{\text{FO}}$  by the addition of the following two axioms:

$$(\forall x)(x = x) \qquad (ref_=)$$

$$t_i = t_j \wedge A(t_i) \rightarrow A(t_j / t_i) \qquad (rep_=)$$

where the notation " $A(t_j / t_i)$ " indicates that some occurrences (possibly all of them) of a term  $t_i$  in a formula  $A$  has been replaced by a term  $t_j$ . Similarly as in the case of substitutions, if a formula of the form  $A(t_j / t_i)$  is considered, it is assumed that the replacement operation was performed "correctly", that is, the variables of  $t_i$  are not bounded in  $A$ , and  $t_j$  is free for each variable of  $t_i$  in  $A$ .

According to the introduced strategy, the axioms correspond to the following two rules:

$$\frac{}{t = t} (R_{ref=}^{\mathcal{O}1}) \qquad \frac{t_i = t_j \quad A(t_i)}{A(t_j / t_i)} (R_{rep=}^1)$$

The assumed semantics is model-theoretic, with the identity predicate interpreted as identity in the domain in every interpretation. The rules have a local character, therefore, for the soundness of the whole ST-system, it is enough to check that they have a property expressed in Lemma 1. We skip the details.

ST-system for FOL with identity results from the system designed for FOL in Section 5 by means of extending the latter with the two rules for identity. Let us call the resulting calculus  $\text{ST}_{\text{FOL}}^=$ .

**Theorem 7** (Completeness). *If a formula is provable in  $\mathcal{F}_{\text{FO}}^=$ , then it is also provable in  $\text{ST}_{\text{FOL}}^=$ .*

**Proof.** We know that all the rules of inference of  $\mathcal{F}_{\text{FO}}^=$  can be simulated in  $\text{ST}_{\text{FOL}}^=$  (Theorem 3). What we have to show is that both axioms are derivable in our system. This is in fact the case as the following derivations show. The first axiom is derivable by:

$$\begin{array}{cc}
 & \swarrow \quad \searrow \\
 x = x & \neg(x = x) \\
 (\forall x)(x = x) & x = x
 \end{array}$$

The derivation starts with branching on  $x = x$ . Then, on the right, we use rule  $(R_{ref=}^{\emptyset 1})$ , to introduce  $x = x$ , which makes the branch closed. Now, the only open branch of the tree is the left one, where the rule **UG** is applied (let us recall that it is derivable in the ST-system for FOL).

The second axiom,  $(rep=)$ , is derivable by means of the following tree:

$$\begin{array}{cc}
 & \swarrow \quad \searrow \\
 t_i = t_j & \neg(t_i = t_j) \\
 \begin{array}{cc}
 \swarrow \quad \searrow & \\
 A(t_i) & \neg A(t_i) \\
 A(t_j // t_i) & \neg(t_i = t_j \wedge A(t_i)) \\
 (rep=) & (rep=)
 \end{array} & \neg(t_i = t_j \wedge A(t_i)) \\
 & (rep=)
 \end{array}$$

This finishes the proof.  $\square$

In some systems for FOL with identity, such as natural deduction system (see [33]), it is possible to use a version of the replacement rule restricted solely to atomic formulas. In our setting, such a rule would have the following form:

$$\frac{t_i = t_j \quad P(t_i)}{P(t_j // t_i)} (repAt)$$

Naturally, the formulas expressing symmetry and transitivity of identity:

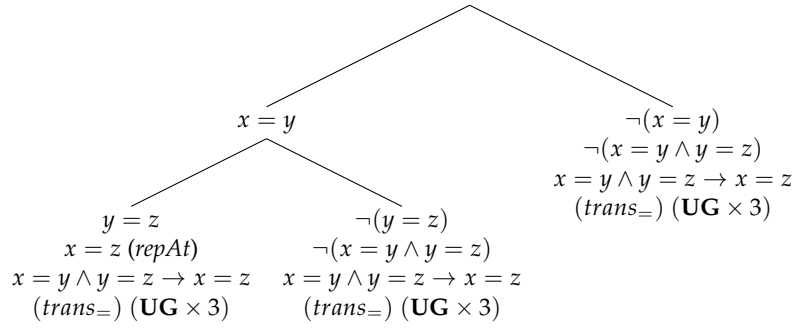
$$(\forall x)(\forall y)(x = y \rightarrow y = x) \quad (sym=)$$

$$(\forall x)(\forall y)(\forall z)(x = y \wedge y = z \rightarrow x = z) \quad (trans=)$$

are derivable in  $ST_{FOL}^=$ , and both are derivable by the use of  $(repAt)$ . Figure 5 presents a derivation of  $(sym=)$ . In Figure 6,  $(trans=)$  is proved, where on the leftmost branch rule  $(repAt)$  is used with  $x = y$  for  $A(t_i)$ ,  $y = z$  for  $t_i = t_j$ , and  $x = z$  for  $A(t_j)$ .

$$\begin{array}{cc}
 & \swarrow \quad \searrow \\
 x = y & \neg(x = y) \\
 \begin{array}{cc}
 \swarrow \quad \searrow & \\
 y = x & \neg(y = x) \\
 x = y \rightarrow y = x & x = x (R_{ref=}^{\emptyset 1}) \\
 (sym=) (UG \times 2) & y = x (repAt)
 \end{array} & x = y \rightarrow y = x \\
 & (sym=) (UG \times 2)
 \end{array}$$

Figure 5. Symmetry.



**Figure 6.** Transitivity.

### 8.3. Second Example: Partial Order

The second example of a first-order theory is that of a partial order. We add to the language  $\mathcal{L}_{\text{FOL}}$  a new predicate symbol,  $\leq$ , and we build  $\mathcal{F}_{\text{FO}}^{\leq}$  by adding two specific axioms to  $\mathcal{F}_{\text{FO}}$ :

$$(\forall x)(x \leq x) \quad (\text{ref}_{\leq})$$

$$(\forall x)(\forall y)(\forall z)(x \leq y \wedge y \leq z \rightarrow x \leq z) \quad (\text{trans}_{\leq})$$

Following the presented strategy, we transform the first axiom into the following rule:

$$\frac{}{x \leq x} (R_{\text{ref}_{\leq}}^{\emptyset 1})$$

which states that at each branch  $x \leq x$  can be synthesized without any additional information present.

The second axiom corresponds to the rule:

$$\frac{x \leq y \quad y \leq z}{x \leq z} (R_{\text{trans}_{\leq}}^1)$$

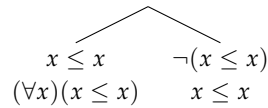
We can synthesize  $x \leq z$  on the condition that we have already synthesized  $x \leq y$  and  $y \leq z$ .

If we add these two rules to  $\text{ST}_{\text{FOL}}$ , we obtain a system which we call:  $\text{ST}_{\text{FOL}}^{\leq}$ .

**Theorem 8** (Completeness). *If a formula is provable in  $\mathcal{F}_{\text{FO}}^{\leq}$ , then it is also provable in  $\text{ST}_{\text{FOL}}^{\leq}$ .*

**Proof.** Again, we can derive the rules of inference of  $\mathcal{F}_{\text{FO}}^{\leq}$  as in Theorem 3. We show how the axioms can be derived.

The first axiom is derivable in a manner similar to reflexivity axiom in FOL with identity:



The second axiom is derived as displayed on Figure 7.  $\square$



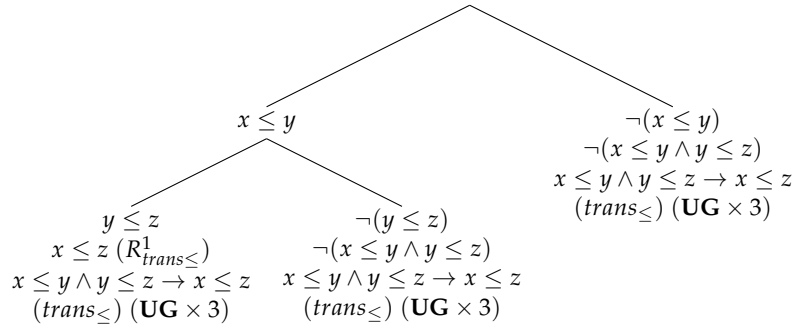


Figure 7. A proof of  $(trans_{\leq})$ .

The tree displayed on Figure 8 represents a proof of formula  $F = \neg(x \leq z) \rightarrow (\neg(x \leq y) \vee \neg(y \leq z))$ . The formula has been synthesized on every open branch. There is one closed branch containing both  $x \leq z$  and its negation.

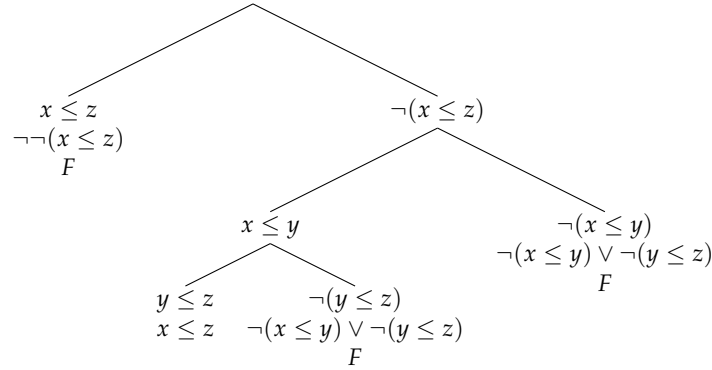


Figure 8. A proof of  $F = \neg(x \leq z) \rightarrow (\neg(x \leq y) \vee \neg(y \leq z))$ .

As in the case of identity, the issue of soundness is settled by verifying that the additional local rules satisfy Lemma 1.

Let us note that the theory of liner order can be obtained from the theory of partial order by means of the addition of the linearity axiom:

$$x \leq y \vee y \leq x \quad (lin_{\leq})$$

which corresponds to one of the following rules:

$$\frac{}{x \leq y \vee y \leq x} (R^{01}_{lin_{\leq}}) \quad \frac{}{x \leq y \mid y \leq x} (R^{01*}_{lin_{\leq}})$$

One can easily show that the ST-system for liner order is capable of simulating corresponding axiomatic system, which implies completeness.

## 9. Conclusions

In this paper, we describe a synthetic tableaux system for First-Order Logic and a general strategy for providing proof systems for some of its axiomatic extensions. We show that the resulting systems are complete with respect to the corresponding axiomatic systems. Synthetic tableaux method was developed to formalize direct validity checking and was inspired by Kalmár's completeness proof (see [28]). ST

system for First-Order Logic is different from the system for propositional logic, due to the fact that one has to introduce *global rules*, which act on the whole proof tree, while on the propositional level *local rules* are sufficient.

The rule of cut is, in a sense, *essential* to this method—it is the rule which starts every derivation introducing the information needed to synthesize more complex formulas. The presence of the cut rule in a system may be an advantage and a disadvantage at the same time. From the *proof search oriented perspective*, an application of the cut rule is problematic: the rule is unrestricted, so finding the right formula to cut on is hard, even if cut is restricted to subformulas of a goal-formula only. On the other hand, once the right cut formula has been found, the resulting proof may have significantly smaller size than that of proofs constructed by more mechanical methods; we indicate this issue in Section 4.

A step in the same direction has been made before in [34]. The authors proposed a proof procedure for First-Order Logic based on tableaux method with the cut rule. Our work is similar to this proposal in the sense that the cut rule is used while proving theorems and not as the post-processing tool employed to transform already existing derivations to reduce their size. The main difference between the approach in [34] and our approach concerns the proof method used and the form of syntactic structures being processed—in the cited work, the rules act on clauses, whereas our solution does not assume deriving a clausal form.

Finally, we wish to point out some open problems related to the introduced systems. Clearly, there is a connection between Frege systems and ST method, which we think should be further investigated. In particular, a question arises whether proof heuristics working well in one system can be transferred to the other. Another problem pertains to the lack of a subformula property being an effect of unrestricted cut applications. Is there a way to restrict the class of formulas appearing in a proof tree for a given formula?

**Author Contributions:** The first author is mainly responsible for Sections 1–7, and the second author is mainly responsible for Sections 8 and 9; however, the general issues as well as many details in the whole paper were discussed and corrected jointly. Conceptualization, D.L.-J. and S.C.; Formal analysis, D.L.-J. and S.C.; Investigation, D.L.-J. and S.C.; Methodology, D.L.-J. and S.C.; Supervision, D.L.-J.; Writing—original draft, D.L.-J. and S.C.; and Writing—review and editing, D.L.-J. and S.C.

**Funding:** The first version of this paper, prepared by Dorota Leszczyńska-Jasion, was part of the project supported by funds of the National Science Centre, Poland, grant No. 2012/04/A/HS1/00715. The work of Szymon Chlebowski was supported financially by the National Science Centre, Poland, grant No. 2017/26/E/HS1/00127.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References

1. Boolos, G. Don't eliminate cut. *J. Philos. Log.* **1984**, *13*, 373–378. [\[CrossRef\]](#)
2. Orevkov, V.P. Lower bounds for increasing complexity of derivations after cut elimination. *J. Sov. Math.* **1982**, *20*, 2337–2350. [\[CrossRef\]](#)
3. Statman, R. Lower Bounds on Herbrand's Theorem. *Proc. Am. Math. Soc.* **1979**, *75*, 104–107.
4. Woltzenlogel Paleo, B. Atomic cut introduction by resolution: proof structuring and compression. In *Logic for Programming, Artificial Intelligence and Reasoning (LPAR-16)*; Lecture Notes in Computer Science; Clark, E., Voronkov, A., Eds.; Springer: Berlin, Germany, 2010; Volume 6355, pp. 463–480.
5. Ebner, G.; Hetzl, S.; Leitsch, A.; Reis, G.; Weller, D. On the Generation of Quantified Lemmas. *J. Autom. Reason.* **2019**, *63*, 95–126. [\[CrossRef\]](#)
6. Finger, M.; Gabbay, D. Cut and pay. *J. Log. Lang. Inf.* **2006**, *15*, 195–218. [\[CrossRef\]](#)
7. Finger, M.; Gabbay, D. Equal Rights for the Cut: Computable Non-analytic Cuts in Cut-based Proofs. *Log. J. IGPL* **2007**, *15*, 553–575. [\[CrossRef\]](#)
8. Urbański, M. Remarks on Synthetic Tableaux for Classical Propositional Calculus. *Bull. Sect. Log.* **2001**, *30*, 194–204.

9. Urbański, M. *Tabele Syntetyczne a Logika pytań (Synthetic Tableaux and the Logic of Questions)*; Wydawnictwo UMCS: Lublin, Poland, 2002.
10. D'Agostino, M. *Investigations into the Complexity of Some Propositional Calculi*; Technical Monograph; Oxford University Computing Laboratory, Programming Research Group: Oxford, UK, 1990.
11. Mondadori, M. Efficient Inverse Tableaux. *J. IGPL* **1995**, *3*, 939–953. [[CrossRef](#)]
12. Urbański, M. Synthetic Tableaux for Łukasiewicz's Calculus Ł3. *Log. Anal.* **2002**, *177–178*, 155–173.
13. Urbański, M. How to Synthesize a Paraconsistent Negation. The Case of CLuN. *Log. Anal.* **2004**, *185–188*, 319–333.
14. D'Agostino, M.; Mondadori, M. The Taming of the Cut. Classical Refutations with Analytic Cut. *J. Log. Comput.* **1994**, *4*, 285–319. [[CrossRef](#)]
15. D'Agostino, M. Are Tableaux an Improvement on Truth-Tables? Cut-Free proofs and Bivalence. *J. Log. Lang. Comput.* **1992**, *1*, 235–252. [[CrossRef](#)]
16. Urbański, M. Synthetic Tableaux and Erotetic Search Scenarios: Extension and Extraction. *Log. Anal.* **2001**, *173–175*, 69–91.
17. Smullyan, R.M. Analytic cut. *J. Symb. Log.* **1968**, *33*, 560–564. [[CrossRef](#)]
18. D'Agostino, M. Tableau Methods for Classical Propositional Logic. In *Handbook of Tableau Methods*; D'Agostino, M., Gabbay, D.M., Hähnle, R., Posegga, J., Eds.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1999; pp. 45–123.
19. Indrzejczak, A. *Natural Deduction, Hybrid Systems and Modal Logics*; Springer: Berlin, Germany, 2010.
20. Urbański, M. *Rozumowania abdukcyjne. Modele i Procedury (Abductive Reasoning. Models and Procedures)*; Wydawnictwo Naukowe UAM: Poznań, Poland, 2009.
21. Wiśniewski, A. Erotetic Search Scenarios. *Synthese* **2003**, *134*, 389–427. [[CrossRef](#)]
22. Wiśniewski, A. *Questions, Inferences, and Scenarios*; Studies in Logic, Logic and Cognitive Systems; College Publications: London, UK, 2013; Volume 46.
23. Leszczyńska-Jasion, D. *Erotetic Search Scenarios as Families of Sequences and Erotetic Search Scenarios as Trees: Two Different, Yet Equal Accounts*; Technical Report 1(1); Institute of Psychology, Adam Mickiewicz University: Poznań, Poland, 2013.
24. Troelstra, A.S.; Schwichtenberg, H. *Basic Proof Theory*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2000.
25. Negri, S.; von Plato, J. *Structural Proof Theory*; Cambridge University Press: Cambridge, UK, 2001.
26. Fitting, M. *First-Order Logic and Automated Theorem Proving*, 2nd ed.; Springer: Berlin, Germany, 1996.
27. Urbański, M. *First-order Synthetic Tableaux*; Technical Report; Institute of Psychology, Adam Mickiewicz University: Poznań, Poland, 2005.
28. Kalish, D.; Montague, R. *Logic. Techniques of Formal Reasoning*, 2nd ed.; Harcourt Brace Jovanovich College Publishers: Fort Worth, TX, USA, 1980.
29. Buss, S.R. Chapter I. "An Introduction to Proof Theory". In *Handbook of Proof Theory*; Buss, S.R., Ed.; Elsevier: Amsterdam, The Netherlands, 1998; pp. 1–78.
30. Cook, S.A.; Reckhow, R.A. On the length of proofs in the propositional calculus. In *Proceedings of the 6th Annual Symposium on the Theory of Computing*, Seattle, WA, USA, 30 April–2 May 1974; pp. 135–148.
31. Cook, S.A.; Reckhow, R.A. The Relative Efficiency of Propositional Proof Systems. *J. Symb. Log.* **1979**, *44*, 36–50. [[CrossRef](#)]
32. Smullyan, R.M. *First-Order Logic*; Springer: Berlin/Heidelberg, Germany; New York, NY, USA, 1968.
33. Negri, S.; von Plato, J. *Proof Analysis: A Contribution to Hilbert's Last Problem*; Cambridge University Press: Cambridge, UK, 2011.
34. Lettmann, M.P.; Peltier, N. A Tableaux Calculus for Reducing Proof Size. In *Proceedings of the 9th International Joint Conference on Automated Reasoning, IJCAR 2018*, Oxford, UK, 14–17 July 2018; pp. 64–80.

