J. Symbolic Computation (1996) **22**, 121–142

On Different Structure-preserving Translations to Normal Form

UWE EGLY[†‡]

Abt. Wissensbasierte Systeme 184/3, TU Wien,
Treitlstraße 3, A–1040 Wien, Austria

(Received 28 June 1995)

In this paper, we compare different definitional transformations into normal form with respect to the Herbrand complexity of the resulting normal forms. Usually, such definitional transformations introduce labels defining subformulae. An obvious optimization is to use implications instead of equivalences, if the subformula occurs in one polarity only, in order to reduce the length of the resulting normal form. We identify a sequence of formulae $H_1, H_2, \ldots$, for which the difference of the Herbrand complexity of the different translations of H_k is bounded from below by a non-elementary function in k. If the optimized translation is applied instead of the unoptimized one, the length of any resolution or cut-free LK-proof of H_k is non-elementary in k instead of exponential in k.

1. Introduction

Most of today's calculi well suited for automating deduction on a computer require the formula being in a specific normal form. In classical first-order deduction, most calculi are based on clause form, i.e., conjunctive or disjunctive normal forms. The usual transformation of an arbitrary formula into a clause form (Bibel, 1993; Chang and Lee, 1973; Loveland, 1978) requires the application of distributivity laws as one of more subtasks. Due to the application of distributivity laws, the resulting clause set can consist of exponentially more occurrences of literals than the formula. Even more important, at least in the first-order case, the structure of the formula is lost. Definitional (or structure-preserving) transformations into clause form avoid such an exponential increase and the destruction of the formula by introducing definitions for subformulae occurring in the formula. For instance, if $A \wedge B$ is a subformula of the given formula F, then the newly introduced definition for the subformula is $L_{A\wedge B} \equiv (L_A \wedge L_B)$, where L_A is the label for A, L_B is the label for B, and $L_{A\wedge B}$ is the label for $A \wedge B$. It is often recommended to optimize the result of the transformation with respect to the number of resulting clauses by using an implication instead of an equivalence if the formula being abbreviated occurs

[†] Work was performed while I was a member of the Intellectics group, TH Darmstadt.
[‡] E-mail: uwe@kr.tuwien.ac.at

0747–7171/96/080121 + 22 $25.00/0

in one polarity only[†]. The length of the resulting formula is, in the best case, a third of the unoptimized formula because an equivalence can result in three clauses, whereas, in the best case, an implication results in one clause.

In the following, we deal with such an optimized translation (called p-definitional translation) and compare it with an unoptimized[‡] one. Our main result is as follows. For a sequence $H_1, H_2, \ldots$ of formulae obtained from a sequence presented in Orevkov (1979), the Herbrand complexity of the formula H'_k obtained by a definitional translation of H_k is exponential in k, but the Herbrand complexity of the formula H''_k obtained from H_k by the p-definitional translation is $\geq c \cdot \mathsf{s}(k-1)$ for a constant c, where $\mathsf{s}(0) = 1$ and $\mathsf{s}(n+1) = 2^{\mathsf{s}(n)}$ for all $n \geq 0$. This non-elementary relation between these Herbrand complexities has two implications. First, it implies a non-elementary relation in proof length if analytic cut-free calculi like LK, variants of analytic tableaux, or quasi-analytic calculi like resolution are applied. Second, since the size of the search space is elementarily related to the length of a shortest proof if, for instance, breadth-first search or depth-first search with iterative deepening is applied, a non-elementary decrease of the size of the search space is achieved by the definitional translation. As a consequence, the optimization is harmful in some cases and should not be mandatory.

It is rather surprising that omitting a small number of redundant parts of a formula has such tremendous impacts on the lengths of proofs and the sizes of search spaces. The non-elementary "speed-up" in proof length obtained by the unoptimized variant of the translation is enabled by the fact that any label introduced for a subformula occurs positively and negatively in the resulting formula. Roughly speaking, not only the subformula, but also the negated subformula is available in the result of the translation. If the subformula can act as a cut formula in a short derivation with cut, then the occurrence of the subformula and its negation enable the simulation of the short derivation with cut by a short derivation without cut. Such a simulation is not possible if the cut formula occurs only positively or only negatively, mainly because the cut formula occurs in both polarities in the short derivation. As a consequence, even for an analytic calculus for non-normal form, a proof of F' can be non-elementarily shorter than a proof of F, where F' denotes the result of an application of the unoptimized translation to F. As already noted, the size of the search space decreases in the same order. Although our result is based on a typical worst-case scenario, one should, however, observe that a non-elementary decrease of the size of the search space cannot be achieved by clever strategies. Even if we know a shortest analytic cut-free proof and, therefore, no alternative decisions are involved, the size of the search space remains non-elementary in k because any such proof has length at least in the order of $\mathsf{s}(k-1)$.

There are only a few investigations how the different translations into clause form influence the length of proofs in classical first-order logic. In Baaz *et al.*, (1994), a definitional translation into clause form in classical first-order logic is compared with traditional transformations into clause form recommended in almost all textbooks on automated deduction. There, the sequence of first-order formulae is translated from Statman's formulae in combinatory logic (Statman, 1979). It turns out that the lengths of shortest proofs

[†] There is an even more refined approach (Boy de la Tour, 1992) combining the traditional transformation and the definitional transformation to normal form. A subformula G is translated in such a way that the length of the resulting normal form of G is minimal.

[‡] When we use the term transformation or definitional transformation, we mean this unoptimized version with equivalences.

of clause sets obtained by a definitional transformation and a standard transformation from the same formula can differ non-elementarily.

The paper is organized as follows. In Section 2, definitions and notations are introduced. In Section 3, we introduce two different definitional translations into clausal normal form. The first translation uses equivalences in the definition of a label for a subformula occurrence, whereas the second one optimizes the length of the resulting clause form by using implications. In Section 4, practical aspects are considered. We present a formalization of the halting problem (Burkholder, 1987; Dafa, 1994) which is automatically proved by our theorem prover KoMeT (Bibel *et al.*, 1994a, b). So far, there are no automatically generated proofs of the formalization in Dafa (1994), either in non-normal form or in normal form. The key feature which enables KoMeT to find the proof is the structure-preserving translation into normal form which is integrated into KoMeT's preprocessing module. In Section 5, a class of formulae originally proposed by Orevkov (1979) is extended. These formulae have Herbrand complexities non-elementary in k. We show that the Herbrand complexities of the unoptimized translations of these formulae are exponential, whereas the Herbrand complexities are non-elementary in k if the optimized (p-definitional) translation is applied.

2. Definitions and Notations

Let $\mathcal{VARS}$ denote the set of all variables, let $\mathcal{FS}$ denote the set of all function symbols, and let $\mathcal{PS}$ denote the set of all predicate symbols. Let $\mathcal{FV}(\phi)$ denote the set of free variables of a formula ϕ or the set of variables of a term ϕ. $\mathcal{PS}$ is extended by a set of indexed predicate symbols of the form P_G, where $P \in \mathcal{PS}$ and G is a formula.

Definition 2.1. *Let $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ be first-order formulae. Then,*

$$A_1, \ldots, A_n \vdash B_1, \ldots, B_m$$

is called a sequent. *$A_1, \ldots, A_n$ form the* antecedent *of the sequent, and $B_1, \ldots, B_m$ form the* succedent *of the sequent. The informal meaning of the sequent is the same as the informal meaning of the formula $(\bigwedge_{i=1}^{n} A_i) \rightarrow (\bigvee_{i=1}^{m} B_i)$. A* non-empty *sequent is a sequent different from $\vdash$.*

Sequents and formulae are denoted by uppercase Roman letters and sequences of formulae are denoted by uppercase Greek letters. Let S be a sequent of the form $\Delta \vdash F$. Then Λ, S denotes the sequent $\Lambda, \Delta \vdash F$.

Definition 2.2. *The* length *of a formula F, denoted by $|F|$, is the number of symbol occurrences in the string representation of the formula. If $\Delta = F_1, \ldots, F_n$ or $\Delta = \{F_1, \ldots, F_n\}$, then $|\Delta| = \sum_{i=1}^{n} |F_i|$. For a sequent $S = \Delta \vdash G$, $|S| = |\Delta| + |G|$. The* logical complexity *of a formula or a sequent is the number of occurrences of $\{\neg, \wedge, \vee, \rightarrow, \forall, \exists\}$ in it.*

Definition 2.3. *Let S_1, S_2, and S be sequents. Then, $\frac{S_1}{S}$ and $\frac{S_1\ S_2}{S}$ are called* inferences. *The former one is a* unary *inference, whereas the latter one is a* binary *inference. S_1 and S_2 are called the* upper *sequents or* premises *and S is called the* lower *sequent or* conclusion *of the inference.*

Since the language has function symbols, we need the following definition which is an extension of Definition 5.2.7 in Gallier (1987).

Definition 2.4. *Let x be a variable and F be a formula. A term t is* free for x in F *if either:*

(i) *F is atomic and the predicate symbol of F is not indexed or*
(ii) *F is atomic and the predicate symbol of F is indexed with G and t is free for x in G or*
(iii) *$F = G \circ H$ and t is free for x in G and in H, for $\circ \in \{\wedge, \vee, \rightarrow\}$ or*
(iv) *$F = \neg G$ and t is free for x in G or*
(v) *$F = \forall y\ G$ or $F = \exists y\ G$ and either:*
   (a) *$x = y$ or*
   (b) *$x \neq y$, $y \notin \mathcal{FV}(t)$ and t is free for x in G.*

The term t introduced for x in A by the rules $\forall l$ and $\exists r$ below must be free for x in A, since otherwise the resulting calculus is not correct.

We use the following sequent calculus based on Baaz and Leitsch (1994) minimizing the number of weakenings in a proof. In contrast to Gentzen's original formulation in Gentzen (1935), rule applications are allowed at arbitrary places in the sequent. As a consequence, the exchange rule can be omitted. Let Γ, Δ, and Λ (possibly subscripted) denote sequences of formulae and let F denote a formula.

The initial sequents (or the axioms) of LK are $F \vdash F$ for a formula F[†]. The inference rules for LK are the logical rules, the quantifier rules and the structural rules without cut. $\mathsf{LK}_{\mathrm{cut}}$ is the calculus extended by the cut rule.

LOGICAL RULES

$$\frac{\Delta_1, A, \Delta_2 \vdash \Gamma}{\Delta_1, (A \wedge B), \Delta_2 \vdash \Gamma}\ \wedge l_1 \qquad \frac{\Delta_1, A, \Delta_2 \vdash \Gamma}{\Delta_1, (B \wedge A), \Delta_2 \vdash \Gamma}\ \wedge l_2$$

$$\frac{\Gamma \vdash \Delta_1, A, \Delta_2 \quad \Lambda \vdash \Pi_1, B, \Pi_2}{\Gamma, \Lambda \vdash \Delta_1, \Pi_1, (A \wedge B), \Delta_2, \Pi_2}\ \wedge r$$

$$\frac{\Gamma_1, A, \Gamma_2 \vdash \Delta_1 \quad \Pi_1, B, \Pi_2 \vdash \Delta_2}{\Gamma_1, \Pi_1, (A \vee B), \Gamma_2, \Pi_2 \vdash \Delta_1, \Delta_2}\ \vee l$$

$$\frac{\Gamma \vdash \Delta_1, A, \Delta_2}{\Gamma \vdash \Delta_1, (A \vee B), \Delta_2}\ \vee r_1 \qquad \frac{\Gamma \vdash \Delta_1, A, \Delta_2}{\Gamma \vdash \Delta_1, (B \vee A), \Delta_2}\ \vee r_2$$

$$\frac{\Gamma \vdash \Delta_1, A, \Delta_2 \quad \Gamma_1, B, \Gamma_2 \vdash \Pi}{(A \rightarrow B), \Gamma, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, \Pi}\ \rightarrow l$$

$$\frac{\Gamma_1, A, \Gamma_2 \vdash \Delta_1, B, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, (A \rightarrow B), \Delta_2}\ \rightarrow r$$

† If we require an atomic formula A instead of F, then the increase of length is bounded by an exponential function in the logical complexity of F (see Lemma 2.1 below).

$$\frac{\Delta \vdash \Gamma_1, A, \Gamma_2}{\neg A, \Delta \vdash \Gamma_1, \Gamma_2}\ \neg l \qquad \frac{\Gamma_1, A, \Gamma_2 \vdash \Delta}{\Gamma_1, \Gamma_2 \vdash \Delta, \neg A}\ \neg r$$

For all logical rules, A and B are called *side* formulae (or *auxiliary* formulae) and $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $\neg A$ are called *principal* formulae of the corresponding $\wedge$, $\vee$, $\rightarrow$, and $\neg$ rules.

Quantifier Rules

$$\frac{\Delta_1, A(t), \Delta_2 \vdash \Gamma}{\Delta_1, \forall x\ A(x), \Delta_2 \vdash \Gamma}\ \forall l \qquad \frac{\Gamma \vdash \Delta_1, A(y), \Delta_2}{\Gamma \vdash \Delta_1, \forall x\ A(x), \Delta_2}\ \forall r$$

$$\frac{\Delta_1, A(y), \Delta_2 \vdash \Gamma}{\Delta_1, \exists x\ A(x), \Delta_2 \vdash \Gamma}\ \exists l \qquad \frac{\Gamma \vdash \Delta_1, A(t), \Delta_2}{\Gamma \vdash \Delta_1, \exists x\ A(x), \Delta_2}\ \exists r$$

$\forall r$ and $\exists l$ must fulfil the eigenvariable condition, i.e., the (free) variable y does not occur in $\Gamma, \Delta_1, \Delta_2$, or $A(x)$. The term t is any term free for x in A. $A(t)$ and $A(y)$ are called the side formulae and $\forall x\ A(x)$ and $\exists x\ A(x)$ are called the principal formulae.

structural rules

weakening:

$$\frac{\Gamma_1, \Gamma_2 \vdash \Delta}{\Gamma_1, A, \Gamma_2 \vdash \Delta}\ wl \qquad \frac{\Delta \vdash \Gamma_1, \Gamma_2}{\Delta \vdash \Gamma_1, A, \Gamma_2}\ wr$$

A is called the *weakening formula.*

contraction:

$$\frac{\Gamma_1, A, \Gamma_2, A, \Gamma_3 \vdash \Delta}{\Gamma_1, A, \Gamma_2, \Gamma_3 \vdash \Delta}\ cl \qquad \frac{\Delta \vdash \Gamma_1, A, \Gamma_2, A, \Gamma_3}{\Delta \vdash \Gamma_1, A, \Gamma_2, \Gamma_3}\ cr$$

cut:

$$\frac{\Gamma \vdash \Delta_1, A, \Delta_2 \quad \Lambda_1, A, \Lambda_2 \vdash \Pi}{\Gamma, \Lambda_1, \Lambda_2 \vdash \Delta_1, \Delta_2, \Pi}\ cut$$

A is called the *cut formula.*

Definition 2.5. *A* derivation α *of a sequent* S *in* LK (*or a* proof *of* S *in* LK) *is a tree of sequents satisfying the following conditions.*

(i) *The topmost sequents of* α *are initial sequents.*
(ii) *Every sequent in* α *except the lowest one is an upper sequent of an inference whose lower sequent is also in* α.
(iii) *The lowest sequent, called the* end sequent, *is* S.

S is called LK-derivable *iff there is a derivation of S in* LK. *If S is a sequent of the form $\vdash F$, where F is a formula, then F is called* LK*-derivable. The* length *of α, denoted by $|\alpha|$, is $\sum_{S \in \mathcal{T}} |S|$, where $\mathcal{T}$ is the multiset of sequences occurring in α. The* height, *h_α, of α is the number of sequents occurring on the longest branch.*

If we require that any initial sequent has the form $A \vdash A$ with A being an atomic formula, then there is an exponential increase of derivation length in the worst case. Initial sequents of this form are called atomic. The following lemma is Lemma 3.2 in Baaz and Leitsch (1994). Recall that the logical complexity of an initial sequent $F \vdash F$ is the number of occurrences of symbols $\{\neg, \wedge, \vee, \rightarrow, \forall, \exists\}$ in F.

LEMMA 2.1. *Let α be an* LK*-derivation of an end sequent S with non-atomic initial sequents. Then there exists an* LK*-derivation β of the same end sequent, β has atomic initial sequents only, and $|\beta| \leq c \cdot |\alpha| \cdot 2^m$, where m is the maximal logical complexity of an axiom in α and c is a constant.*

A special form of a cut, namely an analytic cut, is considered next.

DEFINITION 2.6. *An application of the cut rule in an* $\mathsf{LK}_{\mathrm{cut}}$*-derivation of $\vdash F$ is called* analytic *iff the cut formula is a subformula of F.*

Analytic cuts preserve the subformula property because the cut formula occurs in the formula of the end sequent.

The next definition introduces the concept of positive and negative occurrences of formulae. Roughly speaking, an occurrence of A occurs positively (negatively) in B iff the number of implicit or explicit negation signs preceding A in B is even (odd). The terms positive occurrence and negative occurrence are sometimes referred to as the *polarity* of a formula occurrence in the literature.

DEFINITION 2.7. *Let Π be an occurrence of a formula A in B. If $A = B$, then Π is a positive occurrence in B. If $B = C \circ D$, $\circ \in \{\wedge, \vee\}$ or $B = \mathsf{Q}x\ C$, $\mathsf{Q} \in \{\forall, \exists\}$ and Π is a positive (negative) occurrence of A in C or D, respectively, then the corresponding occurrence Π' of A in B is positive (negative). If $B = C \rightarrow D$ and Π is a positive (negative) occurrence of A in D, then the corresponding occurrence Π' of A in B is positive (negative). If $B = C \rightarrow D$ and Π is a positive (negative) occurrence of A in C, then the corresponding occurrence Π' of A in B is negative (positive). If $B = \neg C$ and Π is a positive (negative) occurrence of A in C, then the corresponding occurrence Π' of A in B is negative (positive).*

In a sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$, any A_i $(1 \leq i \leq n)$ occurs negatively and any B_j $(1 \leq j \leq m)$ occurs positively in the sequent.

DEFINITION 2.8. *If $\forall x\ G$ is a positive (negative) occurrence in a formula F, then this occurrence of $\forall x$ is called a* strong (weak) *quantifier; if $\exists x\ G$ is a positive (negative) occurrence in F, then this occurrence of $\exists x$ is called a weak (strong) quantifier.*

For strong quantifiers, the eigenvariable condition must be satisfied in an LK-derivation. If Skolemization is applied to arbitrary formulae (instead of formulae in negation normal form), then strong quantifiers are removed. Hence, positive occurrences of $\forall$ and negative occurrences of $\exists$ are eliminated.

We use the following Skolemization technique (Andrews, 1971, 1981) which will be implicitly used in the definitional translations defined below. This technique is an optimized variant of the classical Skolemization technique recommended in almost all textbooks, where the arity of a newly introduced Skolem function symbol is minimized. Applying the optimized technique can exponentially shorten proofs (Egly, 1994a, b).

DEFINITION 2.9. *Let F be a formula. Let $G = \mathsf{Q}x\ A(x)$ be a subformula of F, where Q is a strong quantifier and G does not occur in the scope of any other strong quantifier. Let F' be the result of first omitting this occurrence of $\mathsf{Q}x$ from F and then replacing any free occurrence of x in $A(x)$ by $f(y_1, \ldots, y_n)$, where $\{y_1, \ldots, y_n\}$ is the set of free variables occurring in G and f is a new n-ary function symbol not occurring in F. Then,*

$$\alpha(F) = F'.$$

The term $f(y_1, \ldots, y_n)$ is called a Skolem term. *Let m be the number of occurrences of strong quantifiers in F. The* Skolemization of F based on free variables*, denoted by $\mathsf{SK}_{\mathrm{F}}(F)$, is defined by $\alpha^m(F)$, where $\alpha^m(F)$ denotes the iterated application of α, i.e., $\underbrace{\alpha(\alpha(\cdots\alpha}_{m \text{ times}}(F)\cdots))$.*

Let L be a literal. The *complement* of L, denoted by $\bar{L}$, is $\neg A$ if $L = A$ for an atom A, or A otherwise.

DEFINITION 2.10. *A* clause *is a finite conjunction of zero or more literals*[†]*. The* empty clause *is denoted by $\Box$. For all clauses, $A \wedge \Box \wedge B = A \wedge B$. A* contradictory *clause is a clause consisting of complementary literals L and $\neg L$, i.e., C is of the form $C' \wedge L \wedge \neg L$.*

We adopt the positive affirmative representation (Bibel, 1993) of a formula F. Hence, F is translated into a *disjunctive* normal form. In the context of resolution, $\neg F$ is transformed into a conjunctive normal form. These two representations can be translated into each other by replacing $\wedge$ by $\vee$ and vice versa, and by consistently replacing $\neg A$ by A and A by $\neg A$ for any occurrence of an atom A in the normal form. We will use the term "classical" or "traditional" transformations into normal form for translations which first transform a closed first-order formula into a negation normal form, i.e., a formula with connectives from $\{\vee, \wedge, \neg\}$ and the negation signs occur only in front of atoms, then Skolemize the resulting formula, and then apply distributivity laws in order to generate the required normal form. Such translations are presented in most of the textbooks on automated deduction.

DEFINITION 2.11. *Let L be a literal, $C = C' \wedge L$ be a clause, and $C \in \mathcal{S}$, where $\mathcal{S}$ is a clause set. L is called* pure *in $\mathcal{S}$ if $\bar{L}$ is not weakly unifiable*[‡] *with some literal K occurring in a clause from $\mathcal{S} \setminus \{C\}$.*

The following proposition is well known.

[†] By the associativity and commutativity of $\wedge$, the order of the literals in the clause is irrelevant.
[‡] $\bar{L}$ and K are weakly unifiable if there are two renaming substitutions μ and ν such that $\bar{L}\mu = K\nu$.

Proposition 2.1. *Let $\mathcal{S}$ be a clause set, let C be a clause in $\mathcal{S}$, and let C be either a contradictory clause or a clause with a literal L which is pure in $\mathcal{S}$. Then, $\mathcal{S}$ is valid iff $\mathcal{S} \setminus \{C\}$ is valid, i.e., $\mathcal{S}$ and $\mathcal{S} \setminus \{C\}$ are* satisfiability-equivalent (*sat-equivalent*).

Definition 2.12. *Let F be a valid first-order formula without occurrences of strong quantifiers or a valid set of clauses, and let $\mathcal{GI}(F)$ be the set of all ground substitution instances of the formula or the clauses. Then*

$$\min\{|\mathcal{D}| \mid \mathcal{D} \subseteq \mathcal{GI}(F), \mathcal{D} \text{ is valid}\}$$

is called the Herbrand complexity *of F and is denoted by $\mathsf{HC}(F)^{\dagger}$.*

Definition 2.13. *Let C and D be two clauses with distinct variables. Let*

$$C = C_1 \wedge L$$
$$D = D_1 \wedge M,$$

where C_1, D_1 may be $\Box$ and L, M are two literals with complementary signs. If $L\sigma = \bar{M}\sigma$ and σ is most general, then $C_1\sigma \wedge D_1\sigma$ is called a resolvent *of C and D. C and D are called the* parent clauses *of the resolvent and L and M are the literals* resolved upon.

Definition 2.14. *Let $C = C' \wedge K \wedge L$ be a clause and let K and L be two distinct literal occurrences in C which are both positive or both negative. Let σ be a most general unifier of K and L. Then, $C'\sigma \wedge K\sigma$ is called a* binary factor *of C.*

Definition 2.15. *Let $\mathcal{S}$ be a set of clauses. The elements of this clause set are called* input *clauses. A sequence $C_0, \ldots, C_n$ is called* R-deduction (*resolution deduction*) *of a clause C from $\mathcal{S}$ if the following conditions hold.*

(i) $C_n = C$
(ii) *for all $i = 0, \ldots, n$ and $j, k \geq 0$*
   (a) *C_i is a variant of an input clause, or*
   (b) *C_i is a variant of C_j for $j < i$, or*
   (c) *C_i is a binary factor of C_j for $j < i$, or*
   (d) *C_i is a resolvent of C_j, C_k for $j, k < i$.*

A tree *resolution deduction of a clause C from a set $\mathcal{S}$ of input clauses is a resolution deduction of C from $\mathcal{S}$, where the deduction has tree form. Let α be a resolution derivation. Then, the* length $|\alpha|$ *of α is $\sum_{C \in \mathcal{T}} |C|$, where $\mathcal{T}$ is the multiset of clauses occurring in α.*

Let s be the non-elementary function with $\mathsf{s}(0) = 1$ and $\mathsf{s}(n+1) = 2^{\mathsf{s}(n)}$ for all $n \in \mathbb{N}_0$.

3. On Definitional Transformations to Normal Form

In this section, we review the structure-preserving transformation of Plaisted and Greenbaum (1986) as well as Eder's transformation (Eder, 1984, 1992) into definitional

† Observe that symbols are counted instead of disjuncts in the Herbrand disjunction.

form. Eder extended the transformation of Tseitin (1968) to the first-order case, where the introduction of equivalences is retained in order to define labels for subformulae. In a modified transformation, the result is optimized with respect to the length of the definitional form. More precisely, an equivalence is replaced by an implication if the subformula being abbreviated occurs either positively or negatively, but not in both polarities. This is essentially the translation proposed by Plaisted and Greenbaum. In the following, we restrict the translation to first-order formulae with connectives from $\{\neg, \vee, \wedge, \rightarrow\}$. Equivalences can be handled with only a little overhead but, since we do not need formulae with equivalences in the following, we do not consider them in the translation. Moreover, a unique label is introduced for any occurrence of a subformula. Since we do not consider equivalences in the translation, any subformula occurrence has a unique polarity and we can always use implications instead of equivalences in the p-definitional translation.

DEFINITION 3.1. *Let F be a first-order formula. Then $\Sigma(F)$ denotes the set of all occurrences of subformulae of F. Moreover, $\Sigma^+(F)$ denotes the set of all occurrences of subformulae with positive polarity in F and $\Sigma^-(F)$ denotes the set of all occurrences of subformulae with negative polarity in F.*

DEFINITION 3.2. *Let G be an occurrence of a first-order formula and let $\vec{x} = x_1, \ldots, x_k$ be the free variables of G. The atom $L_G(\vec{x})$ is an abbreviation (or* label*) for G. The* length *of a label $L_G(\vec{x})$ is $|G| + 1$.*

DEFINITION 3.3. *Let F be a closed first-order formula and let $G, H, I, K, M \in \Sigma(F)$. For any $G \in \Sigma(F)$ with free variables $\vec{x} = x_1, \ldots, x_k$, a label for G is introduced. Let $\vec{y} = y_1, \ldots, y_l$ be the free variables of H, $\vec{z} = z_1, \ldots, z_m$ be the free variables of I, where $\{\vec{y}\} \subseteq \{\vec{x}\}$, $\{\vec{z}\} \subseteq \{\vec{x}\}$, and $\{\vec{y}\} \cup \{\vec{z}\} = \{\vec{x}\}$. Moreover, $\vec{x}, x$ are the free variables of K and $\vec{x}$ are the free variables of M.*

(i) *If G is atomic, then*

$$C_G^+ = \exists \vec{x}\, (\neg L_G(\vec{x}) \wedge G)$$
$$C_G^- = \exists \vec{x}\, (L_G(\vec{x}) \wedge \neg G).$$

(ii) *If $G = \neg M$, then*

$$C_G^+ = \exists \vec{x}\, (\neg L_G(\vec{x}) \wedge \neg L_M(\vec{x}))$$
$$C_G^- = \exists \vec{x}\, (L_G(\vec{x}) \wedge L_M(\vec{x})).$$

(iii) *If $G = H \vee I$, then*

$$C_G^+ = \exists \vec{x}\, (\neg L_G(\vec{x}) \wedge L_H(\vec{y})) \vee \exists \vec{x}\, (\neg L_G(\vec{x}) \wedge L_I(\vec{z}))$$
$$C_G^- = \exists \vec{x}\, (L_G(\vec{x}) \wedge \neg L_H(\vec{y}) \wedge \neg L_I(\vec{z})).$$

(iv) *If $G = H \wedge I$, then*

$$C_G^+ = \exists \vec{x}\, (\neg L_G(\vec{x}) \wedge L_H(\vec{y}) \wedge L_I(\vec{z}))$$
$$C_G^- = \exists \vec{x}\, (L_G(\vec{x}) \wedge \neg L_H(\vec{y})) \vee \exists \vec{x}\, (L_G(\vec{x}) \wedge \neg L_I(\vec{z})).$$

(v) *If $G = H \rightarrow I$, then*

$$C_G^+ = \exists \vec{x}\, (\neg L_G(\vec{x}) \wedge \neg L_H(\vec{y})) \vee \exists \vec{x}\, (\neg L_G(\vec{x}) \wedge L_I(\vec{z}))$$
$$C_G^- = \exists \vec{x}\, (L_G(\vec{x}) \wedge L_H(\vec{y}) \wedge \neg L_I(\vec{z})).$$

(vi) *If* $G = \exists x\ K$*, then*

$$\begin{aligned} C^+_G &= \exists\vec{x}\,\exists x\ (\neg L_G(\vec{x}) \wedge L_K(\vec{x}, x)) \\ C^-_G &= \exists\vec{x}\ (L_G(\vec{x}) \wedge \neg L). \end{aligned}$$

(vii) *If* $G = \forall x\ K$*, then*

$$\begin{aligned} C^+_G &= \exists\vec{x}\ (\neg L_G(\vec{x}) \wedge L) \\ C^-_G &= \exists\vec{x}\,\exists x\ (L_G(\vec{x}) \wedge \neg L_K(\vec{x}, x)). \end{aligned}$$

The atom of L *is* $L_K(\vec{x}, g(\vec{x}))$*, where* g *is a globally new function symbol neither occurring in* F *nor being introduced in the translation of any other subformula. The* definitional form *of* F *is the formula*

$$\delta(F) = \bigvee_{G\in\Sigma(F)} (C^+_G \vee C^-_G).$$

The corresponding clause set is denoted by $\gamma(F)$*. The* p-definitional form (*the definitional form obeying polarities*) *is the formula*

$$\delta^q_p(F) = \Bigl(\bigvee_{G\in\Sigma^+(F)} C^q_G\Bigr) \vee \Bigl(\bigvee_{G\in\Sigma^-(F)} C^r_G\Bigr),$$

where $q \in \{+,-\}$ *and* $\{r\} = \{+,-\} \setminus \{q\}$*.* $\delta^+_p(F)$ *is used for positive occurrences of* F*, and* $\delta^-_p(F)$ *is used for negative occurrences of* F*. The corresponding clause set is denoted by* $\gamma^q_p(F)$*.*

It is well known that $\gamma(F) \cup \{L_F\}$ as well as $\gamma^+_p(F) \cup \{L_F\}$ is valid iff F is valid†. Moreover, the time and space complexity of the translation of a first-order formula F is at most quadratic in $|F|$.

REMARK 3.1. *Each occurrence of a subformula is abbreviated by a unique label in the translations above. There are several improvements which can yield considerable impacts on proof length. We will give an example in Section* 5.

DEFINITION 3.4. *Let* F *be a closed first-order formula. A* definitional (*tree*) *resolution deduction of* F *is a* (*tree*) *resolution deduction of* $\neg L_F$ *from* $\gamma(F)$*. A* p-definitional (*tree*) *resolution deduction of* F *is a* (*tree*) *resolution deduction of* $\neg L_F$ *from* $\gamma^+_p(F)$*.*

EXAMPLE 3.1. *Let* $F = (\forall x\ p(x)) \rightarrow \exists y\ p(y)$*. Five labels are introduced by the p-definitional translation, namely* $L_{p(x)}(x)$*,* $L_{p(y)}(y)$*,* $L_{\forall xp(x)}$*,* $L_{\exists yp(y)}$*, and* L_F*. Observe that the predicate symbols* $L_{p(x)}$ *and* $L_{p(y)}$ *are different. Then* $\gamma^+_p(F)$ *consists of the following clauses.*

$$\begin{aligned} C_1 &= L_{p(x)}(x) \wedge \neg p(x) \\ C_2 &= \neg L_{p(y)}(y) \wedge p(y) \end{aligned}$$

† If our p-definitional form is compared with the definitional translation in Plaisted and Greenbaum (1986), one may observe that the signs "+" and "−" are exchanged. The reason is that we transform F into p-definitional form resulting in a disjunctive normal form, whereas Plaisted and Greenbaum transform $\neg F$ into p-definitional form resulting in a conjunctive normal form.

$$C_3 = L_{\forall x\ p(x)} \wedge \neg L_{p(x)}(x)$$
$$C_4 = \neg L_{\exists y\ p(y)} \wedge L_{p(y)}(y)$$
$$C_5 = \neg L_F \wedge \neg L_{\forall x\ p(x)}$$
$$C_6 = \neg L_F \wedge L_{\exists y\ p(y)}.$$

It is possible to derive the clauses which occur in a traditional translation into clause form from $\{C_1, \ldots, C_6, L_F\}$ *by resolution. First, derive* $\neg L_{\forall x\ p(x)}$ *and* $L_{\exists y\ p(y)}$ *from* C_5, C_6, *and* L_F. *Then the clauses* $\neg L_{p(x)}(x)$ *and* $L_{p(y)}(y)$ *are derived. Two further resolutions yield two clauses* $\neg p(x)$ *and* $p(y)$. *These clauses occur if the traditional translation into clause form is applied. If the first resolutions with parent clause* L_F *are omitted and an additional binary factorization with literal occurrences of* L_F *is introduced, then we have a p-definitional resolution deduction of* F.

The p-definitional form of a formula can be considered as a "reduced" variant of Eder's translation in the sense that only one direction of the equivalence is generated. One may ask whether it makes sense to use Eder's approach and to generate more clauses than necessary. The hidden assumption behind this question is the supposition that one can find proofs for short clause sets much easier as for clause sets with more elements. The number of clauses, however, is not the only significant measure for the intractability of a clause set. For instance, Quaife (1990) as well as Guha and Zhang (1989) observed that rather long clause sets obtained from Andrews' challenge problem [problem 34 in Pelletier (1986)] by different variants of definitional translations can be refuted faster than shorter clause sets. Moreover, additional information can enable much shorter proofs and even much smaller search spaces. We will see in Section 5 that there exists a sequence of formulae $H_1, H_2, \ldots$ such that any p-definitional resolution derivation of H_k has length non-elementary in k, but there exists a definitional resolution derivation of H_k of length exponential in k. The search space is decreased in the same order, because its size is elementarily related to the length of a shortest proof. In the next section, we consider a practical application of definitional translations in order to show their practical value which is widely neglected.

4. A Practical Application

In this section, the practical value of definitional translations into normal form is demonstrated. The comparison made here is between a variant of the p-definitional normal form and the traditional clause form[†]. We use the famous halting problem (Burkholder, 1987; Bruschi, 1991; Dafa, 1993, 1994), where we consider Dafa's formalization in Dafa (1994). There are several unsuccessful attempts to prove this problem automatically, either by high-performance theorem provers based on normal forms, or by theorem provers based on non-normal forms. The key feature which enables our theorem prover KoMeT (Bibel *et al.*, 1994a, b) to find a proof is the p-definitional translation into normal form. This claim is immediately apparent, since we get, after some simple syntax transformations of the input clauses, a proof of the halting problem with the OTTER system (McCune, 1994). Hence, we have a practically interesting and successful applica-

[†] A detailed comparison between traditional, p-definitional and definitional translations is given in Egly and Rath (1996), where approximately 200 test problems from an α-version of the non-normal form TPTP library are used.

Table 1. Intuitive meaning of the predicates.

Predicate	Meaning
$a(x)$	x is an algorithm
$c(x)$	x is a computer program in some programming language
$d(x, y, z)$	x is able to decide whether y halts on a given input z
$h_2(x, y)$	x halts on a given input y
$h_3(x, y, z)$	x halts on given input pair $\langle y, z\rangle$
$o(x, y)$	x outputs y

tion of the definitional transformations into normal form, which are often considered as a theoretically interesting but practically useless technique.

The formalization is taken from Dafa (1994). Table 1 presents the intuitive meaning of predicates used in the formalization. The formulae are as follows.

$$\begin{aligned} (\exists x\ (a(x) \wedge \forall y\ (c(y) \rightarrow \forall z\ d(x, y, z)))) \rightarrow \exists w\ (c(w) \wedge \forall y\ (c(y) \\ \rightarrow \forall z\ d(w, y, z))) \end{aligned} \tag{4.1}$$

$$\begin{aligned} \forall w\ ((c(w) \wedge \forall u\ (c(u) \rightarrow \forall v\ d(w, u, v))) \rightarrow \forall y\, z\ ((c(y) \wedge h_2(y, z) \\ \rightarrow (h_3(w, y, z) \wedge o(w, g)) \wedge (c(y) \wedge \neg h_2(y, z) \\ \rightarrow (h_3(w, y, z) \wedge o(w, b))))) \end{aligned} \tag{4.2}$$

$$\begin{aligned} \forall w\ ((c(w) \wedge \forall y\, z\ ((c(y) \wedge h_2(y, z) \rightarrow (h_3(w, y, z) \wedge o(w, g)) \wedge (c(y) \wedge \neg h_2(y, z) \\ \rightarrow (h_3(w, y, z) \wedge o(w, b)))) \\ \rightarrow \exists v\ (c(v) \wedge \forall y\ (((c(y) \wedge h_3(w, y, y) \wedge o(w, g)) \\ \rightarrow \neg h_2(v, y)) \wedge ((c(y) \wedge h_3(w, y, y) \wedge o(w, b)) \\ \rightarrow (h_2(v, y) \wedge o(v, b))))))) \end{aligned} \tag{4.3}$$

$$\neg(\exists x\ (a(x) \wedge \forall y\ (c(y) \rightarrow \forall z\ d(x, y, z)))). \tag{4.4}$$

An explanation of the different formulae (4.1) to (4.4) can be found in Dafa (1994). The problem is to prove

$$(4.1) \wedge (4.2) \wedge (4.3) \rightarrow (4.4). \tag{4.5}$$

Formula (4.5) is transformed into p-definitional form by introducing labels for subformulae. Table 2 shows which labels are introduced, where the same label is introduced for different syntactically identical copies of the same subformula. Applied to the resulting clause set, KoMeT as well as OTTER finds a proof† in less than 30 sec. This example shows that, even if the number of clauses in the normal form is slightly increased by using a p-definitional translation instead of a traditional translation, the proof is found much easier for the larger clause set.

5. A Non-elementary Speed-up Result

After considering the practical value of definitional translations into normal form, we are interested in a comparison of the p-definitional translation with the definitional translation with respect to proof length. Our discussion is based on a sequence $H_1, H_2, \ldots$

† The KoMeT proof is documented in Egly and Rath (1995).

Table 2. The labels and subformulae.

Label	Subformula	Polarity
$d_1(x, y)$	$\forall z\ d(x, y, z)$	p/n
$d_2(y, x)$	$c(y) \rightarrow d_1(x, y)$	p/n
$d_3(x)$	$\forall y\ d_2(y, x)$	p/n
$d_4(x)$	$a(x) \wedge d_3(x)$	p/n
d_5	$\exists x\ d_4(x)$	p/n
$d_6(w, y)$	$\forall z\ d(w, y, z)$	n
$d_7(y, w)$	$c(y) \rightarrow d_6(w, y)$	n
$d_8(w)$	$\forall y\ d_7(y, w)$	n
$d_9(w)$	$c(w) \wedge d_8(w)$	n
d_{10}	$\exists w\ d_9(w)$	n
d_{11}	$d_5 \rightarrow d_{10}$	n
$d_{12}(w, u)$	$\forall v\ d(w, u, v)$	p
$d_{13}(u, w)$	$c(u) \rightarrow d_{12}(w, u)$	p
$d_{14}(w)$	$\forall u\ d_{13}(u, w)$	p
$d_{15}(w)$	$c(w) \wedge d_{14}(w)$	p
$d_{16}(y, z)$	$c(y) \wedge h_2(y, z)$	p/n
$d_{17}(w, y, z)$	$h_3(w, y, z) \wedge o(w, g)$	p/n
$d_{18}(y, z, w)$	$d_{16}(y, z) \rightarrow d_{17}(w, y, z)$	p/n
$d_{19}(y, z)$	$c(y) \wedge \neg h_2(y, z)$	p/n
$d_{20}(w, y, z)$	$h_3(w, y, z) \wedge o(w, b)$	p/n
$d_{21}(y, z, w)$	$d_{19}(y, z) \rightarrow d_{20}(w, y, z)$	p/n
$d_{22}(y, z, w)$	$d_{18}(y, z, w) \wedge d_{21}(y, z, w)$	p/n
$d_{23}(w)$	$\forall yz\ d_{22}(y, z, w)$	p/n
$d_{24}(w)$	$d_{15}(w) \rightarrow d_{23}(w)$	n
d_{25}	$\forall w\ d_{24}(w)$	n
$d_{26}(w)$	$c(w) \wedge d_{23}(w)$	p
$d_{27}(y, w)$	$c(y) \wedge h_3(w, y, y) \wedge o(w, g)$	p
$d_{28}(y, w, v)$	$d_{27}(y, w) \rightarrow \neg h_2(v, y)$	n
$d_{29}(y, w)$	$c(y) \wedge h_3(w, y, y) \wedge o(w, b)$	p
$d_{30}(y, v)$	$h_2(v, y) \wedge o(v, b)$	n
$d_{31}(y, w, v)$	$d_{29}(y, w) \rightarrow d_{30}(y, v)$	n
$d_{32}(y, w, v)$	$d_{28}(y, w, v) \wedge d_{31}(y, w, v)$	n
$d_{33}(w, v)$	$\forall y\ d_{32}(y, w, v)$	n
$d_{34}(v, w)$	$c(v) \wedge d_{33}(w, v)$	n
$d_{35}(w)$	$\exists v\ d_{34}(v, w)$	n
$d_{36}(w)$	$d_{26}(w) \rightarrow d_{35}(w)$	n
d_{37}	$\forall w\ d_{36}(w)$	n
d_{38}	$d_{25} \wedge d_{37}$	n
d_{39}	$d_{11} \wedge d_{38}$	n
d_{40}	$d_{39} \rightarrow \neg d_5$	p

of formulae which are modified and extended variants of formulae $F_1, F_2, \ldots$ presented in Orevkov (1979).

Definition 5.1. *Let F_k occur in the infinite sequence of formulae $(F_k)_{k\in\mathbb{N}}$ where*

$$F_k = \forall b_0\ ((\forall w_0 \exists v_0\ p(w_0, b_0, v_0) \wedge \forall uvw\ (\exists y\ (p(y, b_0, u) \wedge \exists z\ (p(v, y, z) \wedge p(z, y, w))) \\ \rightarrow p(v, u, w))) \\ \rightarrow \exists v_k\ (p(b_0, b_0, v_k) \wedge \exists v_{k-1}\ (p(b_0, v_k, v_{k-1}) \wedge \ldots \wedge \exists v_0\ p(b_0, v_1, v_0)) \cdots).$$

Using this sequence $(F_k)_{k\in\mathbb{N}}$, Orevkov showed that cut elimination can tremendously affect proof length. More precisely, he proved that there exists an LK_{cut}-derivation of

$\vdash F_k$ with a single occurrence of the cut rule and the number of sequents in this derivation is linear in k, but any cut-free LK-derivation of $\vdash F_k$ has height $\geq 2 \cdot \mathsf{s}(k) + 1$.

Considering the modified formulae $H_1, H_2, \ldots$, there are two open questions in our context.

(i) Does $\delta_p^+(H_k)$ have Herbrand complexity non-elementary in k?
(ii) Does $\delta(H_k)$ have Herbrand complexity elementary in k?

We answer both questions positively in the sequel of this section. It is shown in the following that using the definitional form instead of the p-definitional form may yield a non-elementary decrease of Herbrand complexity (and proof length) compared to the length of any (shortest) cut-free proof. We consider tree derivations in LK. The difference between this tree-oriented point of view and a dag†-oriented or sequence-oriented point of view with respect to proof length is at most exponential. We first present a slightly modified version of Orevkov's $\mathsf{LK}_{\mathrm{cut}}$-derivation of $\vdash F_k$ with one application of the cut rule. Let ψ_k be this $\mathsf{LK}_{\mathrm{cut}}$-derivation. The cut is then changed to an analytic cut by extending F_k by $A \vee q$, where A is the cut formula and q is an atom neither occurring in F_k nor in A. The remaining derivation is adjusted accordingly. Instead of F_k, we get an extended formula H_k. The Herbrand complexity is estimated for $\mathsf{SK}_{\mathrm{F}}(H_k)$ as well as for $\gamma_p^+(H_k) \cup \{L_{H_k}\}$ which are both in the order of $\mathsf{s}(k-1)$. Finally, we show that there exists a definitional tree resolution derivation of H_k such that the length of this derivation as well as the Herbrand complexity for $\gamma(H_k) \cup \{L_{H_k}\}$ is exponential in k. As a consequence, proof length as well as Herbrand complexity is decreased non-elementarily by the use of the definitional form.

By such an extreme decrease of proof length, the potential search space is decreased in the same order if a suitable search strategy like breadth-first search or depth-first search with iterative deepening is assumed. The reason is that the search space which has to be considered in order to find a proof of F is elementarily related to the length m of a shortest proof of F. Therefore, if m is decreased non-elementarily, the size of the search space is also decreased non-elementarily. This result might surprise because the branching degree of each node in the search space is increased by introducing additional clauses by the definitional translation. However, this increase of the branching degree yields an elementary increase of the size of the search space which is more than compensated by the much shorter minimal proof length.

First, a short derivation of $\vdash F_k$ in $\mathsf{LK}_{\mathrm{cut}}$ is presented. Abbreviations shown in Figure 1 are used in order to simplify the notation. The first step is the derivation of the sequent $A_0(b_0), C \vdash A_1(b_0)$. This derivation is called β_1 and uses α_1. We first introduce α_i ($i \geq 1$) and then β_0 and β_1.

$$\alpha_i \ (i \geq 1)$$

$$\dfrac{Ax(w_i, b_0, v_i) \qquad \dfrac{Ax(w_{i-1}, w_i, v_i') \quad Ax(v_i', w_i, v_{i-1})}{p(w_{i-1}, w_i, v_i'), p(v_i', w_i, v_{i-1}) \vdash C_1(w_{i-1}, w_i, v_{i-1})}\ \wedge r, \exists r}{p(w_i, b_0, v_i), p(w_{i-1}, w_i, v_i'), p(v_i', w_i, v_{i-1}) \vdash C_2(v_i, w_{i-1}, v_{i-1})}\ \wedge r, \exists r$$

† directed acyclic graph.

$$
\begin{aligned}
C_1(\alpha,\beta,\gamma) &= \exists z\ (p(\alpha,\beta,z) \wedge p(z,\beta,\gamma))\\
C_2(\alpha,\beta,\gamma) &= \exists y\ (p(y,b_0,\alpha) \wedge C_1(\beta,y,\gamma))\\
C &= \forall u\, \forall v\, \forall w\ (C_2(u,v,w) \rightarrow p(v,u,w))\\
B_0(\alpha) &= \exists v_0\ p(b_0,\alpha,v_0)\\
B_{i+1}(\alpha) &= \exists v_{i+1}\ (p(b_0,\alpha,v_{i+1}) \wedge B_i(v_{i+1}))\\
A_0(\alpha) &= \forall w_0\, \exists v_0\ p(w_0,\alpha,v_0)\\
A_{i+1}(\alpha) &= \forall w_{i+1}\ (A_i(w_{i+1}) \rightarrow \bar{A}_{i+1}(w_{i+1},\alpha))\\
\bar{A}_0(\alpha,\delta) &= \exists v_0\ p(\alpha,\delta,v_0)\\
\bar{A}_{i+1}(\alpha,\delta) &= \exists v_{i+1}\ (A_i(v_{i+1}) \wedge p(\alpha,\delta,v_{i+1}))\\
Ax(x,y,z) &= p(x,y,z) \vdash p(x,y,z)\\
Bx(v) &= A_{i-2}(v) \vdash A_{i-2}(v).
\end{aligned}
$$

Figure 1. Abbreviations used in the following LK-derivations.

β_0

$$
\dfrac{A_0(b_0) \vdash A_0(b_0)}{A_0(b_0), C \vdash A_0(b_0)}\ wl.
$$

β_1

$$
\dfrac{Ax(w_1,b_0,v_1) \qquad \dfrac{\dfrac{\dfrac{\alpha_1 \qquad \dfrac{Ax(w_0,v_1,v_0)}{p(w_0,v_1,v_0) \vdash \bar{A}_0(w_0,v_1)}\ \exists r}{p(w_1,b_0,v_1), C, p(w_0,w_1,v_1'), p(v_1',w_1,v_0) \vdash \bar{A}_0(w_0,v_1)}\ \rightarrow l, \forall l, \forall l, \forall l}{p(w_1,b_0,v_1), C, p(w_0,w_1,v_1'), A_0(w_1) \vdash \bar{A}_0(w_0,v_1)}\ \exists l, \forall l}{p(w_1,b_0,v_1), C, A_0(w_1) \vdash A_0(v_1)}\ \exists l, \forall l, \forall r, cl}{\dfrac{p(w_1,b_0,v_1), C \vdash A_0(w_1) \rightarrow \bar{A}_1(w_1,b_0)}{A_0(b_0), C \vdash A_1(b_0)}\ \exists l, \forall l, \forall r}\ \wedge r, \exists r, \rightarrow r, cl
$$

The next step is the derivation of a sequent $A_0(b_0), C \vdash A_i(b_0)$ $(i > 1)$. For each i, we decompose this derivation into two derivations γ_i and β_i.

$\gamma_i\ (i > 1)$

$$
\dfrac{\dfrac{\alpha_i \qquad \dfrac{Bx(v_{i-1}) \quad Ax(w_{i-1},v_i,v_{i-1})}{p(w_{i-1},v_i,v_{i-1}), A_{i-2}(v_{i-1}) \vdash \bar{A}_{i-1}(w_{i-1},v_i)}\ \wedge r, \exists r}{p(w_i,b_0,v_i), C, p(w_{i-1},w_i,v_i'), A_{i-2}(v_{i-1}), p(v_i',w_i,v_{i-1}) \vdash \bar{A}_{i-1}(w_{i-1},v_i)}\ \rightarrow l, \forall l, \forall l, \forall l}{p(w_i,b_0,v_i), C, p(w_{i-1},w_i,v_i'), \bar{A}_{i-1}(v_i',w_i) \vdash \bar{A}_{i-1}(w_{i-1},v_i)}\ \wedge l_1, \wedge l_2, cl, \exists l
$$

$\beta_i \ (i > 1)$

$$
\dfrac{\dfrac{Ax(w_i, b_0, v_i) \qquad \dfrac{\dfrac{Bx(w_{i-1}) \qquad \dfrac{\dfrac{Bx(v_i') \quad \gamma_i}{p(w_i, b_0, v_i), C, A_{i-2}(v_i'), p(w_{i-1}, w_i, v_i'), A_{i-1}(w_i) \vdash \bar{A}_{i-1}(w_{i-1}, v_i)} \; \to l, \forall l}{p(w_i, b_0, v_i), C, \bar{A}_{i-1}(w_{i-1}, w_i), A_{i-1}(w_i) \vdash \bar{A}_{i-1}(w_{i-1}, v_i)} \; \wedge l_1, \wedge l_2, cl, \exists l}{p(w_i, b_0, v_i), C, A_{i-1}(w_i), A_{i-2}(w_{i-1}) \vdash \bar{A}_{i-1}(w_{i-1}, v_i)} \; \to l, \forall l, cl}{p(w_i, b_0, v_i), C, A_{i-1}(w_i) \vdash A_{i-1}(v_i)} \; \to r, \forall r}{p(w_i, b_0, v_i), C \vdash A_{i-1}(w_i) \to \bar{A}_i(w_i, b_0)} \; \wedge r, \exists r, \to r, cl}{A_0(b_0), C \vdash A_i(b_0)} \; \exists l, \forall l, \forall r
$$

$\delta_0(t)$

$$
\dfrac{\dfrac{\dfrac{Ax(b_0, t, v_0')}{\exists v_0 \ p(b_0, t, v_0) \vdash \exists v_0 \ p(b_0, t, v_0)} \; \exists r, \exists l}{A_0(t) \vdash B_0(t)} \; \forall l}{A_0(b_0), C, A_0(t) \vdash B_0(t)} \; wl, wl
$$

$\delta_{i+1}(t) \ (i \geq 0)$

$$
\dfrac{\beta_i \qquad \dfrac{\dfrac{Ax(b_0, t, v_{i+1}) \quad \delta_i(v_{i+1})}{A_0(b_0), C, A_i(v_{i+1}), p(b_0, t, v_{i+1}) \vdash B_{i+1}(t)} \; \wedge r, \exists r}{A_0(b_0), C, \bar{A}_{i+1}(b_0, t) \vdash B_{i+1}(t)} \; \wedge l_1, \wedge l_2, cl, \exists l}{A_0(b_0), C, A_{i+1}(t) \vdash B_{i+1}(t)} \; \to l, \forall l, cl, cl
$$

Now we can finish the derivation of F_k by ϵ_k $(k \geq 0)$.

ϵ_k

$$
\dfrac{\dfrac{\beta_k \quad \delta_k(b_0)}{A_0(b_0), C \vdash B_k(b_0)} \; cut, cl, cl}{\vdash \forall b \ ((A_0(b) \wedge C) \to B_k(b))} \; \wedge l_1, \wedge l_2, cl, \to r, \forall r
$$

The derivation ψ_k of $\vdash F_k$ in LK_{cut} presented so far has one application of the cut rule where the cut formula has a free variable. We transform ψ_k into ϕ_k by making the application of the cut rule analytic.

DEFINITION 5.2. *Let H_k $(k \in \mathbb{N})$ be a formula of the form*

$$\forall b \ (((A_k(b) \vee q) \wedge A_0(b) \wedge C) \to B_k(b))$$

where q is a predicate with a predicate symbol not occurring elsewhere in $A_i(b)$, $B_i(b)$ $(i = 0, \ldots, k)$, and C.

An $\mathsf{LK}_{\mathrm{cut}}$-derivation of $\vdash H_k$ is obtained from the $\mathsf{LK}_{\mathrm{cut}}$-derivation of $\vdash F_k$ presented above by simply adding a wl inference with weakening formula $A_k(b) \vee q$ directly below the cut. Let ϵ'_k $(k \geq 0)$ be the following new derivations.

$$\epsilon'_k$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\beta_k \quad \delta_k(b_0)}{A_k(b_0) \vee q, A_0(b_0), C \vdash B_k(b_0)}\, cut, wl, cl, cl}{(A_k(b_0) \vee q) \wedge A_0(b_0) \wedge C, A_0(b_0), C \vdash B_k(b_0)}\, \wedge l_1}{(A_k(b_0) \vee q) \wedge A_0(b_0) \wedge C \vdash B_k(b_0)}\, \wedge l_1, \wedge l_2, cl, \wedge l_2, cl}{\vdash \forall b\, (((A_k(b) \vee q) \wedge A_0(b) \wedge C) \to B_k(b))}\, \to r, \forall r$$

By replacing ϵ_k by ϵ'_k, we transform ψ_k into ϕ_k. The latter $\mathsf{LK}_{\mathrm{cut}}$-derivation has only one application of the analytic cut rule.

LEMMA 5.1. *Let ϕ_k be the $\mathsf{LK}_{\mathrm{cut}}$-derivation of $\vdash H_k$ with one application of an analytic cut described above. Then, $|\phi_k| \leq c \cdot 2^{d \cdot k}$ for constants c and d.*

PROOF. By Theorem 2 in Orevkov (1979), the number of sequents in the $\mathsf{LK}_{\mathrm{cut}}$-derivation of $\vdash F_k$ is $e \cdot k$, where e is a constant. Since $|A_k(b) \vee q| \leq c_1 \cdot 2^{d_1 \cdot k}$ for constants c_1, d_1, there exist constants c, d such that the length of ϕ_k is less than $c \cdot 2^{d \cdot k}$. □

In contrast to the short $\mathsf{LK}_{\mathrm{cut}}$-derivation of $\vdash F_k$, any derivation of $\vdash F_k$ in LK without cut has length non-elementary in k. The following lemma is a corollary of Theorem 1 in Orevkov (1979).

LEMMA 5.2. *Let ψ_k be a cut-free LK-derivation of $\vdash F_k$. Then, $h_{\psi_k} \geq 2 \cdot \mathsf{s}(k) + 1$.*

For $\mathsf{SK}_{\mathrm{F}}(F_k)$, the Herbrand complexity can be estimated with the following result from Theorem 4 in Orevkov (1979).

LEMMA 5.3. *Let $\mathcal{S}_{F_k}$ be a clause set obtained from F_k by a classical translation† into normal form, i.e., without introducing new predicate symbols. Any resolution refutation of $\mathcal{S}_{F_k}$ has length greater than $\mathsf{s}(k-1) - \log_2(k+5)$.*

Let $\mathcal{T}$ be a valid set of Horn clauses‡, let $\mathcal{T}'$ be a valid set of ground instances of clauses from $\mathcal{T}$, and let l be the length of $\mathcal{T}'$. Then there exists a resolution proof of $\mathcal{T}'$ of length polynomial in l. This resolution proof can be lifted to a resolution proof of $\mathcal{T}$. Therefore, $\mathsf{HC}(\mathcal{S}_{F_k}) > c \cdot \mathsf{s}(k-1)$, because any resolution proof of the Horn clause set $\mathcal{S}_{F_k}$ has length greater than $\mathsf{s}(k-1) - \log_2(k+5)$.

LEMMA 5.4. *Let F be a valid closed first-order formula and let $\mathcal{S}_F$ be a clause set obtained from F by a classical translation into normal form. Then,*

$$\mathsf{HC}(\mathsf{SK}_{\mathrm{F}}(F)) \leq |\mathsf{SK}_{\mathrm{F}}(F)| \cdot \mathsf{HC}(\mathcal{S}_F) \qquad \textit{and} \qquad \mathsf{HC}(\mathcal{S}_F) \leq |\mathcal{S}_F| \cdot \mathsf{HC}(\mathsf{SK}_{\mathrm{F}}(F)).$$

† In contrast to Orevkov, we use another variant of Skolemization but the results are identical for F_k.

‡ Such clauses have at most one negative literal because a disjunctive normal form is used.

PROOF. Let $\mathcal{T} = \{C_1\sigma_1, \ldots, C_n\sigma_n\}$ be a valid set of minimal length of ground instances of clauses occurring in $\mathcal{S}_F$. Note that $\mathcal{S}_F$ is sat-equivalent to F.

$\mathsf{HC}(\mathsf{SK}_\mathrm{F}(F)) \leq |\mathsf{SK}_\mathrm{F}(F)| \cdot \mathsf{HC}(\mathcal{S}_F)$: Replace any $C_i\sigma_i \in \mathcal{T}$ $(1 \leq i \leq n)$ by a ground instance of $\mathsf{SK}_\mathrm{F}(F)$, namely $G = \mathsf{SK}_\mathrm{F}(F)\sigma_i\mu_i$ such that $C_i\sigma_i$ occurs in $\mathcal{S}_G$. Note that $\mathsf{SK}_\mathrm{F}(F)$ has weak quantifiers only and that all weakly quantified variables are replaced by ground terms. The second ground substitution is needed to ground the remaining part of $\mathsf{SK}_\mathrm{F}(F)$, i.e., any remaining variable is replaced by a common constant. Therefore, $\{\mathsf{SK}_\mathrm{F}(F)\sigma_1\mu_1, \ldots, \mathsf{SK}_\mathrm{F}(F)\sigma_n\mu_n\}$ is valid and $\mathsf{HC}(\mathsf{SK}_\mathrm{F}(F)) \leq |\mathsf{SK}_\mathrm{F}(F)| \cdot \mathsf{HC}(\mathcal{S}_F)$.

$\mathsf{HC}(\mathcal{S}_F) \leq |\mathcal{S}_F| \cdot \mathsf{HC}(\mathsf{SK}_\mathrm{F}(F))$: Transform any ground instance occurring in the valid set of ground instances of $\mathsf{SK}_\mathrm{F}(F)$ into clause form. The resulting clause set is valid and the Herbrand complexity of $\mathcal{S}_F$ is $\leq |\mathcal{S}_F| \cdot \mathsf{HC}(\mathsf{SK}_\mathrm{F}(F))$. □

The last lemma together with the polynomial lengths (in k) of $\mathsf{SK}_\mathrm{F}(F_k)$ and $\mathcal{S}_F$ imply

LEMMA 5.5. $\mathsf{HC}(\mathsf{SK}_\mathrm{F}(F_k)) \geq c \cdot \mathsf{s}(k-1)$ *for a constant* c.

Up to now, we know that the Herbrand complexities of F_k and $\mathcal{S}_{F_k}$ are non-elementary in k. Next, we consider the p-definitional translations of H_k and F_k.

LEMMA 5.6. *Let $\mathcal{S}$ be a valid set of clauses, $C \in \mathcal{S}$, $\mathcal{S}' = \mathcal{S} \setminus \{C\}$, and C is either of the form $C' \wedge K$, where K is pure in $\mathcal{S}$, or of the form $C' \wedge \neg L \wedge L$ with an atom L. Then,* $\mathsf{HC}(\mathcal{S}') = \mathsf{HC}(\mathcal{S})$.

PROOF. Let $\mathcal{T}$ be a valid set of minimal length of ground instances from $\mathcal{S}$ and let $\mathcal{U} \subset \mathcal{T}$ be the set of all ground instances of C in $\mathcal{T}$. Assume that $\mathcal{U} \neq \{\}$. Let $\mathcal{T}' = \mathcal{T} \setminus \mathcal{U}$ be the result of the deletion of contradictory clauses and clauses with pure literals from $\mathcal{T}$. Since such deletions preserve validity, $\mathcal{T}'$ is valid iff $\mathcal{T}$ is valid. By the deletion of clauses from $\mathcal{T}$, $|\mathcal{T}'| \leq |\mathcal{T}|$. Since all ground instances of C in $\mathcal{T}$ occur in $\mathcal{U}$, $\mathcal{T}'$ is a valid set of ground instances of clauses from $\mathcal{S}'$. But then, $\mathcal{T}$ cannot be of minimal length. Therefore, $\mathcal{U} = \{\}$ and $\mathsf{HC}(\mathcal{S}') = \mathsf{HC}(\mathcal{S})$. □

LEMMA 5.7. *Let $\mathcal{S}$ be the valid clause set $\gamma_p^+(H_k) \cup \{L_{H_k}\}$ and let $\mathcal{S}' = \gamma_p^+(F_k) \cup \{L_{F_k}\}$. Then,* $\mathsf{HC}(\mathcal{S}') = \mathsf{HC}(\mathcal{S})$.

PROOF. Recall Definition 5.2 and note that any occurrence of a subformula is abbreviated by a unique label. The following clauses

$$C_1 = L_{A_k(x) \vee q}(x) \wedge \neg L_{A_k(x)}(x) \wedge \neg L_q$$
$$C_2 = L_q \wedge \neg q$$

occur in $\mathcal{S}$ where $L_{A_k(x) \vee q}(x)$ is the label for $A_k(x) \vee q$, $L_{A_k(x)}(x)$ is the label for $A_k(x)$, and L_q is the label for q. The predicate symbol q occurs in C_2 only. Let us consider an arbitrary valid set $\mathcal{T}$ of ground instances of clauses from $\mathcal{S}$. Note that q is a pure literal in $\mathcal{T}$. As a consequence, C_2 can be removed from $\mathcal{T}$ without affecting validity. By this removal of C_2, the literal $\neg L_q$ becomes pure in the resulting clause set and any ground instance of C_1 is removed. By these removals, all clauses introduced for subformula occurrences of $A_k(x)$ can be stepwise deleted because they have pure literals. Moreover, clauses containing the literal $\neg L_{A_k(x) \vee q}(x)$ are removed. Eventually, we get $\mathcal{T}'$

representing ground instances of $\gamma_p^+(F_k) \cup \{L_{F_k}\}$ and $\mathcal{T}'$ is valid. Hence, $\mathsf{HC}(\mathcal{S}') = \mathsf{HC}(\mathcal{S})$. $\square$

LEMMA 5.8. *$\mathsf{HC}(\gamma_p^+(F_k) \cup \{L_{F_k}\}) \geq c \cdot \mathsf{s}(k-1)$ for a constant c.*

PROOF. Let $\mathcal{S} = \gamma_p^+(F_k) \cup \{L_{F_k}\}$ and let $\mathcal{T} = \{C_1\sigma_1, \ldots, C_n\sigma_n\}$ be a valid set of ground instances of clauses occurring in $\mathcal{S}$. Let $C_i = L_i \wedge C_i'$ $(i = 1, \ldots, n)$ with $L_i \in \{L_H(x_1, \ldots, x_m), \neg L_H(x_1, \ldots, x_m)\}$ for a subformula occurrence H of F_k and $C_i \in \gamma_p^+(H)$ or $C_i \in \gamma_p^-(H)$. Let H' be the existential closure of the Skolemized subformula H, i.e., $H' = \exists \mathsf{SK}_{\mathrm{F}}(H)$. Note that $\mathsf{SK}_{\mathrm{F}}(F_k)$ and H' have weak quantifiers only and that all weakly quantified variables are replaced by ground terms. Replace any $C_i\sigma_i \in \mathcal{T}$ $(i = 1, \ldots, n)$ by a ground instance of $\mathsf{SK}_{\mathrm{F}}(F_k)$, namely $G = \mathsf{SK}_{\mathrm{F}}(F_k)\sigma_i\mu_i$ such that the ground subformula $H'\sigma_i\lambda_i$ occurs in $\mathsf{SK}_{\mathrm{F}}(F_k)\sigma_i\mu_i$. The second ground substitutions μ_i and λ_i are needed to ground the remaining part of $\mathsf{SK}_{\mathrm{F}}(F_k)$ and H', respectively, i.e., any remaining variable is replaced by a common constant. Therefore, $\{\mathsf{SK}_{\mathrm{F}}(F_k)\sigma_1\mu_1, \ldots, \mathsf{SK}_{\mathrm{F}}(F_k)\sigma_n\mu_n\}$ is valid and $\mathsf{HC}(\mathsf{SK}_{\mathrm{F}}(F_k)) \leq |\mathsf{SK}_{\mathrm{F}}(F_k)| \cdot \mathsf{HC}(\mathcal{S})$. As a consequence, $\mathsf{HC}(\gamma_p^+(F_k) \cup \{L_{F_k}\}) \geq c \cdot \mathsf{s}(k-1)$ $\square$

The following proposition is (partially) Proposition 3.6.2 in Eder (1992).

PROPOSITION 5.1. *Let ϕ be a tree derivation of $\vdash F$ in LK for some closed first-order formula F. Then there is a derivation Φ of $\neg L_F$ from $\gamma(F)$ in tree resolution whose length is less than $45 \cdot |\phi|^4$.*

If we would like to obtain tree resolution derivations from derivations in $\mathsf{LK}_{\mathrm{cut}}$ such that the former derivation is only polynomially longer than the latter derivation, we need the concept of extensions (Tseitin, 1968; Eder, 1992). Roughly speaking, an extension allows for the introduction of a new formula G, which is introduced into the derivation as $\gamma(G)$. Hence, a cut formula A is translated into $\gamma(A)$. Since in cut-free LK-derivations, there are no cut formulae, extensions are not necessary in a corresponding resolution derivation of $\neg L_F$ from $\gamma(F)$ of approximately the same length. If a cut formula is a subformula of the formula in the end sequent, then the result of an extension is already in the definitional normal form. More precisely, $\gamma(A) \subset \gamma(F)$ and, as a consequence, a derivation of $\vdash F$ in $\mathsf{LK}_{\mathrm{cut}}$ with the restriction that all cut formulae occur as subformulae in F can be translated into a tree resolution derivation of $\neg L_F$ from $\gamma(F)$ with only a moderate increase of proof length. Hence, we get the following lemma.

LEMMA 5.9. *Let ϕ be a tree derivation of $\vdash F$ in $\mathsf{LK}_{\mathrm{cut}}$ for some closed first-order formula F, where all cuts are analytic. Then, there is a tree resolution derivation Φ of $\neg L_F$ from $\gamma(F)$ and $|\Phi| < 45 \cdot |\phi|^4$.*

If we would use either $\gamma_p^+(A)$ or $\gamma_p^-(A)$ instead of $\gamma(A)$ then the analytic cut cannot be simulated and a result similar to Lemma 5.9 would not hold in this case. The reason for the non-existence of a short simulation of an analytic cut by p-definitional resolution is the non-availability of the cut formula in both polarities. Lemma 5.9 and Lemma 5.1, together with the fact that Herbrand complexity is a lower bound on the length of tree resolution proofs[†] (for valid clause sets) yield the following corollary.

COROLLARY 5.1. *There exists a tree resolution derivation* Φ *of* $\neg L_{H_k}$ *from* $\gamma(H_k)$ *and* $|\Phi| \leq c \cdot 2^{d \cdot k}$ *for constants* c, d. *Moreover,* $\mathsf{HC}(\gamma(H_k) \cup \{L_{H_k}\}) \leq c \cdot 2^{d \cdot k}$.

This corollary, together with Lemma 5.8 yields the following theorem.

THEOREM 5.1. *There exists an infinite sequence of valid formulae* $(H_k)_{k \in \mathbb{N}}$ *such that, for* $k > 1$, $\mathsf{HC}(\gamma_p^+(H_k) \cup \{L_{H_k}\})$ *as well as the length of any tree resolution derivation of* $\neg L_{H_k}$ *from* $\gamma_p^+(H_k)$ *is not less than* $c \cdot \mathsf{s}(k-1)$ *for a constant* c, *but there exists a resolution derivation* Φ *of* $\neg L_{H_k}$ *from* $\gamma(H_k)$ *with length exponential in* k. *Moreover,* $\mathsf{HC}(\gamma(H_k) \cup \{L_{H_k}\})$ *is exponential in* k.

We mentioned in Remark 3.1 that there are some improvements for structure-preserving translations which can have strong impacts on proof length and proof search. We examine the power of the following improvement: introduce the same label for all occurrences of the same subformula. In other words, we consider the formula represented as a dag instead of a tree. By using a formula dag instead of a formula tree, the representation of a formula can be exponentially shorter.

Consider the formula G_k

$$\forall b\ ((((A_k(b) \to A_k(b)) \vee q) \wedge A_0(b) \wedge C) \to B_k(b))$$

where q is a predicate with a predicate symbol not occurring elsewhere in $A_i(b)$, $B_i(b)$ $(i = 0, \ldots, k)$, and C. If we introduce labels with different predicate symbols for any occurrence of the same formula (and therefore, the two identical copies of A_k get labels with different predicate symbols) then the Herbrand complexity of the p-definitional form is non-elementary in k. However, if identical labels are introduced, a simulation of an analytic cut is possible which results in a low Herbrand complexity of the improved p-definitional form.

6. Conclusion

In this paper, we showed that definitional translations are not only of theoretical interest but also have some practical value which is often underestimated. From a theoretical point of view, two variants of the same translation behave very differently with respect to the Herbrand complexities of the resulting clause sets. The optimized translation, which introduces implications instead of equivalences reduces the number of clauses occurring in the resulting normal form. This optimization has the severe drawback that simulations of analytic cuts are not possible in some circumstances in analytic calculi, but this simulation is possible if equivalences are used. This result shows that information which seems to be worthless can be utilized in order to get drastically shorter and more readable proofs, together with a drastically reduced size of the search space.

[†] This can be shown to be similar to the case of linear input resolution in Baaz and Leitsch (1992).

Acknowledegments

The author would like to thank Hans Tompits and the anonymous referees for their constructive criticism and for their useful comments on an earlier version of this paper.

References

Andrews, P.B. (1971). Resolution in type theory. *J. Symbolic Logic*, **36**:414–432.

Andrews, P.B. (1981). Theorem proving via general matings. *J. ACM*, **28**(2):193–214.

Baaz, M., Fermüller, C., Leitsch, A. (1994). A non-elementary speed up in proof length by structural clause form transformation. *Proc. LICS'94*, pp. 213–219. IEEE Computer Society Press Los Alamitos, California.

Baaz, M., Leitsch, A. (1992). Complexity of resolution proofs and function introduction. *Annals of Pure and Applied Logic*, **57**:181–215.

Baaz, M., Leitsch, A. (1994). On skolemization and proof complexity. *Fundamenta Informaticae*, **20**:353–379.

Bibel, W. (1993). *Deduction: Automated Logic*. London: Academic Press.

Bibel, W., Brüning, S., Egly, U., Rath, T. (1994a). Komet. In A. Bundy, (ed), *Proc. Conference on Automated Deduction*, pp. 783–787. Heidelberg: Springer-Verlag.

Bibel, W., Brüning, S., Egly, U., Rath, T. (1994b). Towards an Adequate theorem prover based on the connection method. In *Proc. Sixth International Conference on Artificial Intelligence and Information-Control of Robots*, pp. 137–148. World Scientific Publishing Company.

Boy de la Tour, T. (1992). An optimality result for clause form translation. *J. Symbolic Computation*, **14**:283–301.

Bruschi, M. (1991). The halting problem. *AAR Newsletter*, pp. 7–12, March.

Burkholder, L. (1987). The halting problem. *SIGACT News*, **18**(3):48–60.

Chang, C.L., Lee., R.C. (1973). *Symbolic Logic and Mechanical Theorem Proving*. New York: Academic Press.

Dafa, L. (1993). A mechanical proof of the halting problem in natural deduction style. *AAR Newsletter*, pp. 4–9, June.

Dafa, L. (1994). The formulation of the halting problem is not suitable for describing the halting problem. *AAR Newsletter*, pp. 1–7, October.

Eder, E. (1984). An implementation of a theorem prover based on the connection method. In W. Bibel and B. Petkoff, (eds), *AIMSA 84, Artificial Intelligence—Methodology, Systems, Applications, Varna, Bulgaria*. North-Holland Publishing Company.

Eder, E. (1992). *Relative Complexities of First Order Calculi*. Braunschweig: Vieweg.

Egly, U. (1994a). *On Methods of Function Introduction and Related Concepts*. PhD thesis, TH Darmstadt, Alexanderstr. 10, D–64283 Darmstadt.

Egly, U. (1994b). On the value of antiprenexing. In F. Pfenning, (ed), *Proceedings of the International Conference on Logic Programming and Automated Reasoning*, pp. 69–83. Berlin: Springer-Verlag.

Egly, U. (1996). On definitional translations to normal form for intuitionistic logic. *Fundamenta Informaticae*. To appear.

Egly, U., Rath, T. (1995). The halting problem: an automatically generated proof. *AAR Newsletter*, **30**:10–16.

Egly, U., Rath, T. (1996). On the practical value of different definitional translations to normal form. In M.A. McRobbie, J.K. Slaney, (eds), *Proc. Conference on Automated Deduction* pp. 403–417. Berlin: Springer-Verlag.

Gallier, J.H. (1987). *Logic for Computer Science: Foundations of Automatic Theorem Proving*. New York: John Wiley & Sons.

Gentzen, G. (1935). Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, **39**:176–210, 405–431. English translation in Szabo (1969).

Greenbaum, S., Nagasaka, A., O'Rorke, P., Plaisted, D. (1982). Comparison of natural deduction and locking resolution implementations. In D. W. Loveland, (ed), *Proc. Conference on Automated Deduction*, volume 138 of *Lecture Notes in Computer Science*, pp. 159–171. Berlin: Springer-Verlag.

Guha, A., Zhang, H. (1989). Andrews' challenge problem: clause conversion and solutions. *AAR Newsletter*, pp. 5–8, December.

Loveland, D.W. (1978). *Automated Theorem Proving: A Logical Basis*, volume 6 of *Fundamental Studies in Computer Science*. North-Holland, Amsterdam, New York, Oxford.

McCune, W. (1994). OTTER 3.0 Reference Manual and Guide. Technical Report ANL-94/6, Argonne National Laboratory, Mathematics and Computer Science Division.

Orevkov, V.P. (1979). Lower bounds for increasing complexity of derivations after cut elimination. *Zapiski Nauchnykh Seminarov Leningradskogo Otdeleniya Matematicheskogo Instituta im V.A. Steklova AN SSSR*, **88**:137–161. English translation in *J. Soviet Mathematics*, (1982) 2337–2350.

Pelletier, F.J. (1986). Seventy-five problems for testing automatic theorem provers. *J. Automated Reasoning*, 2:191–216.

Plaisted, D.A., Greenbaum, S. (1986). A structure-preserving clause form translation. *J. Symbolic Computation*, **2**:293–304.

Quaife, A. (1990). Andrews' challenge problem revisited. *AAR Newsletter*, pp. 3–7, May.

Statman, R. (1979). Lower Bounds on Herbrand's Theorem. In *Proc. AMS 75*, pp. 104–107.

Szabo, M.E. (ed) (1969). *The Collected Papers of Gerhard Gentzen*. Studies in Logic and the Foundations of Mathematics. Amsterdam: North-Holland.

Takeuti, G. (1975). *Proof Theory*, volume 81 of *Studies in Logic and the Foundations of Mathematics*. Amsterdam: North-Holland.

Tseitin, G.S. (1968). On the complexity of derivation in propositional calculus. In A. O. Slisenko, (ed), *Studies in Constructive Mathematics and Mathematical Logic, Part II*, pp. 234–259. Seminars in Mathematics, V.A. Steklov Mathematical Institute, vol. 8, Leningrad. English translation: Consultants Bureau, New York, 1970, pp. 115–125.