Short CNF in Finitely Valued Logics

Reiner Hähnle*

Institut für Logik, Komplexität und Deduktionssysteme
Universität Karlsruhe, 7500 Karlsruhe, Germany
haehnle@ira.uka.de

Abstract. We present a transformation of formulæ from arbitrary nitely valued logics into a conjunctive normal form based on signed atomic formulæ which can be used to syntactically characterize many valued validity with a simple resolution rule very much like in classical logic. The transformation is always linear with relation to the size of the input, and we dene a generalized concept of polarity in order to remove clauses which are not needed in the proof. The transformation rules are based on the concept of 'sets assigns' developed earlier by the author in the context of tableau based deduction in many valued logics. We claim that the approach presented here is much more efcient than existing approaches to many valued resolution.

Introduction

With this paper we make a step toward the efcient mechanization of deduction in many valued logics. The need for research of that kind is motivated by the recent advent of new applications for many valued theorem proving in various subelds, for example, in formal hardware verication [5]. Other applications exist in the theory of error correcting codes or in nonmonotonic reasoning.

It is widely acknowledged that the existence of clausal normal forms for a logic can greatly improve efciency and speed of theorem proving procedures for that logic. Resolution based theorem provers usually rely on the input being in conjunctive normal form (CNF), but also most other proof procedures that claim high performance, employ CNF transformation as a preprocessing step. If these successful techniques are to be used in nonclassical theorem proving, it is likely that some variant of CNF is required for the respective nonclassical logics.

There are three main obstacles that have to be overcome when clausal normal forms are to be used in a generalized context:

1. Normal forms can become exponentially long wrt the length of the input when a naïve algorithm is used. This is not so problematic in classical logic where knowledge bases usually consist of conjunctions of relatively short formulæ. In nonclassical logics, however, even relatively short formulæ can become quite large during this process already.
2. The normalized input bears no resemblance to the orginal formula. This makes it hard to explain the machine generated proof to the user.
3. Many nonclassical logics may fail to have normal forms, or at least it is nontrivial to nd them.

The rst two problems can principally be solved by using a structure preserving clause form translation (dened in the following section) which has the double advantage of (i) producing normal forms in linear time and space wrt to the input and (ii)

* Research supported by Deutsche Forschungsgemeinschaft (DFG).

establishing a relationship between the clauses of the normal form and the subformulæ of the input formula.

As to the third problem, it is not likely that there is a uniform solution to it due to the diversity of nonclassical logics. It has been shown, however, that for certain classes of nonclassical logics structure preserving CNF transformations (and corresponding resolution rules) can be devised to give the desired results [7]. These include intuitionistic logic and various modal logics.

The purpose of this note is to dene structure preserving clause form translations (together with a suitable denition of clauses and a resolution rule) for arbitrary nitely valued logics, a domain where, to our best knowledge, no general results exist so far. The normal form computation will be linear wrt to the length of the input and quadratic wrt to the number of truth values in the worst case. The short CNF translation for many valued logics proposed in the following is centered around a technique that has been developed earlier by the author [3, 4] in connection with nonclausal theorem proving with semantic tableaux. The main advantage is a relatively simple resolution procedure for nitelyvalued logics which avoids the drawbacks of a nonclausal approach [11], while retaining the main advantages of resolution, notably, strategies for pruning the search space. We treat the propositional case thoroughly and give some hints how to handle the rst order case. Due to space restrictions we omit proofs. These may be found in the long version of this paper which is avilable from the author on request.

1 Short Normal Forms in Classical Logic

In the following we denote with $\#(M)$ the cardinality of a set M and with $|s|$ we denote the length of a string s. We use $\ulcorner x \urcorner$ to denote the ceiling function on the rationals. We assume the reader is familiar with the basic notions of computational logic. Throughout the paper we will use a standard syntax for propositional and rstorder logic, here and there enriched with some new unary and binary operator symbols. Clauses are considered as nite multisets of literals.

The central idea behind structure preserving clause form translations is to introduce additional atoms which serve as abbreviations for subformulæ of the input. Assume we have a propositional formula ϕ and we need a nite set of clauses X_ϕ such that $\models \phi$ iff $X_\phi \vdash \Box$.

Let $SF(\phi)$ denote the set of subformulas of ϕ (note that $\#(SF(\phi)) = |\phi|$) and let $m = \#(SF(\phi))$. We denote with $\bar{L}$ the complement of a literal. Now we introduce a new variable p_i for each $\phi_i \in SF(\phi)$ which is not a literal and consider for each $1 \leq i \leq m$ and $\phi_i = (\phi_j \text{ op } \phi_k)$ the formula

$$p_i \leftrightarrow (p_j \text{ op } p_k) \tag{1}$$

where op is the topmost connective of ϕ_i and p_j, p_k either correspond to ϕ_j, ϕ_k or $\phi_l = p_l$ if ϕ_l is a literal. This process is called abbreviation, denition or renaming by various authors. Let $X_{\phi,i}$ be a CNF representation of (1). The number of clauses in $X_{\phi,i}$ is bound by a constant depending on the type of connectives present and is at most 4; each clause contains at most 3 literals. Now we can dene

$$X_\phi = \left(\bigcup_i X_{\phi,i} \right) \cup \{\bar{p1}\} \tag{2}$$

where $p1$ is the denition of ϕ. It is fairly easy to see that X_ϕ has indeed the desired properties, in particular, X_ϕ contains at most $12m + 1$ literals.

In the rstorder case the p_i are atomic formulæ with an appropriate arity.

Example 1. Consider the propositional tautology $p \supset (q \supset p)$. We introduce the following renamings: $p1 \leftrightarrow p \supset p2$, $p2 \leftrightarrow q \supset p$. So the formula is a tautology iff the following set of clauses is unsatisable:

$$\begin{array}{lll} 1. & \sim p1 & \\ 2. & \sim p1 \vee p2 \vee \sim p & \\ 3. & p1 \vee p & \\ 4. & p1 \vee \sim p2 & \end{array} \qquad \begin{array}{lll} 5. & \sim p2 \vee p \vee \sim q & \\ 6. & p2 \vee q & \\ 7. & p2 \vee \sim p & \end{array}$$

A refutation of this clause set is as follows:

$$\begin{array}{lll} 8. & [1,3] & p \\ 9. & [1,4] & \sim p2 \end{array} \qquad \begin{array}{lll} 10. & [7,8] & p2 \\ 11. & [9,10] & \Box \end{array}$$

Note that 2., 5. (each corresponding to one half of an equivalence) and 6. were not needed. Obviously, our CNF contains redundant clauses.

There is a rather obvious improvement of the procedure, if one observes that instead of logical equivalence in (1), depending on the polarity (cf. [10]) of p_i in the original formula (ie in $\sim \phi$), only one direction of the implication is needed in order to characterize satisability. Therefore, instead of (1) we write

$$\begin{array}{l} p_i \supset (p_j \text{ op } p_k) \text{ if } p_i \text{ occurs positively in } \sim \phi \\ (p_j \text{ op } p_k) \supset p_i \text{ if } p_i \text{ occurs negatively in } \sim \phi \\ p_i \leftrightarrow (p_j \text{ op } p_k) \text{ if } p_i \text{ occurs positively and negatively in } \sim \phi \end{array} \tag{3}$$

If we apply this optimization to the previous example, clauses 2. and 5. are not generated.

Further improvements are possible if not all subformulas are being renamed, for instance, conjunctions need not to be renamed. An optimal result (the clause set $\{p, \sim p, q\}$) would have been obtained using a topdown renaming algorithm [2] which takes this into account. In [2] one may also nd additional references regarding structure preserving clause form for classical logic.

2 ManyValued Logic

Denition 1 Syntax, Truth Values. Let L be a propositional language with propositional variables L_0 and connectives F. Let $N = \left\{0, \frac{1}{n-1}, \ldots, \frac{n-2}{n-1}, 1\right\}$ be the set of truth values and let $n = \#(N)$.

Denition 2 Semantics, ManyValued Logic. Connectives $F \in \mathbf{F}$ are interpreted as functions with nite range and domain, in other words, if k is the arity of F we associate a function $f : N^k \to N$ with F which we call the interpretation of F. Let f be the family of functions over N associated with connectives in F. Then we call f $n$valued matrix for L and the triple $\langle \mathbf{L}, \mathbf{f}, N \rangle$ $n$valued propositional logic.

Denition 3 Valuation. Let $\mathcal{L} = \langle \mathbf{L}, \mathbf{f}, N \rangle$ be a $n$valued propositional logic. A valuation for $\mathcal{L}$ is a function $v : L_0 \to N$. As usual, v can be uniquely extended to a homomorphism from L to N via

$$v(F(\phi 1, \ldots, \phi_k)) = f(v(\phi 1), \ldots, v(\phi_k))$$

where f is the interpretation of F.

Denition 4 *S*Satisable, *S*Tautology. For $S \subseteq N$ and a $n$valued propositional logic $\mathcal{L}$ call a formula $\phi \in$ L *S*satisable iff there is a valuation such that $v(\phi) \in S$. Call ϕ a *S*tautology iff $v(\phi) \in S$ for all valuations.

For some examples we refer the reader to the following section. Our task is now to nd

1. a language of clauses C;
2. a structure preserving linear translation tr from $\mathbf{L} \times 2^N$ into 2^C;
3. a resolution rule on C, i.e. a decidable relation $R \subseteq C^{k+1}$ for some k.

such that $\mathsf{tr}(\phi, S) \vdash \Box$ iff ϕ is a *S*tautology (where $\vdash$ is the reexive and transitive closure of R).

3 A Structure Preserving Normal Form Translation for ManyValued Logics

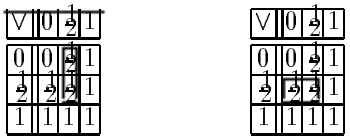
In [3, 4] the author introduced semantic tableau systems that can be used to implement a generic theorem prover which performs efciently in a variety of nitelyvalued logics. The key idea was to enhance the formula language in such a way that the still to be considered valuations at each step of the proof can efciently be kept track of. The technical device was the use of truth value sets as signs or prexes in front of the formulæ. We dene the set of signed formulæ $\mathbf{L}^* = \{S : \phi | S \subseteq N, \phi \in \mathbf{L}\}$ with the intended meaning

$$S : \phi \text{ is satisable iff } v(\phi) \in S \text{ for some } v.$$

Example 2. A sound and complete rule with premise $\{\frac{1}{2}\} : \phi \vee \psi$, where $\vee$ is three valued strong Kleene disjunction (which is dened $v(\phi \vee \psi) = \max(v(\phi), v(\psi))$) is

$$\frac{\{\frac{1}{2}\} : \phi \vee \psi}{\begin{array}{c|c} \{0, \frac{1}{2}\} : \phi & \{\frac{1}{2}\} : \phi \\ \{\frac{1}{2}\} : \psi & \{0, \frac{1}{2}\} : \psi \end{array}}$$

One way to visualize rules with a premise $S : (\phi \text{ op } \psi)$ uses coverings of those entries in the truth table of op that are members of the set S. Each of the rule extensions corresponds to a partial covering of these entries. The union of all coverings corresponds to the collection of extensions that make up the conclusion of a rule. In Example 2 above, the left extension covers the area indicated in the following diagram on the left, while the right extension covers the area shown in the diagram on right.



Now, tableau rules and tableaux correspond to DNF formulæ, while we are interested in CNF formulæ. What we need, therefore, are inverse tableau rules where the extensions are conjunctively connected and the extensions themselves are clauses over signed atoms. Consequently, we dene the language of clauses C to be the clauses over L_0^*. The $\vee$ in C will be interpreted classically, i.e. twovalued (like the implicit disjunctions of tableau branches in manyvalued tableaux which are also interpreted classically):

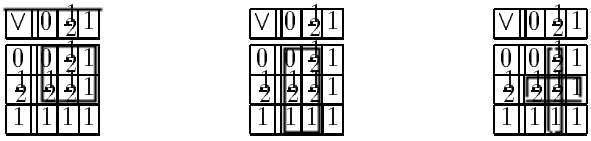
Denition 5 Satisability on C. An atomic signed formula $S : p$ is satised by v iff $v(p) \in S$. Let $D \in C$ and $D = S1 : p1 \vee \cdots \vee Sk : pk$. D is satised by v iff v satises at least one $Si : pi$ in D. A clause set $X \subseteq C$ is satised by v iff v satises simultaneously each member of X. We write $v \models X$ for this fact. X is satisable iff $v \models X$ for some v.

How can we compute inverse tableau rules? We can still use the technique with coverings, however, we must turn things around. Each extension (or clause) corresponds to a covering that contains at least the entries occurring in S. The intersection of all coverings must contain exactly the entries occurring in S.

Example 3. The inverse tableau rule with the premise of Example 2 is:

$$\frac{\{\frac{1}{2}\} : \phi \vee \psi}{\{0, \frac{1}{2}\} : \phi \;\Big\|\; \{0, \frac{1}{2}\} : \psi \;\Big\|\; \begin{array}{l}\{\frac{1}{2}\} : \phi \\ \{\frac{1}{2}\} : \psi\end{array}}$$

The extensions correspond to the following coverings:



The conclusion of the rule corresponds to the following set of C clauses:

$$\{\{0, \tfrac{1}{2}\} : \phi,\ \{0, \tfrac{1}{2}\} : \psi,\ \{\tfrac{1}{2}\} : \phi \vee \{\tfrac{1}{2}\} : \psi\}$$

We draw inverse tableau rules with double vertical bars to distinguish them from the ordinary rules.

The next step is to express logical equivalence within this framework. Consider the following denition of (strong) manyvalued equivalence:

$$v(\phi \leftrightarrow \psi) = \begin{cases} 1 & v(\phi) = v(\psi) \\ 0 & \text{otherwise} \end{cases}$$

Let us give a formulation with C clauses of $\leftrightarrow$. It will be convenient to use the following abbreviations for signs:

Denition 6 Signs. Let $j \in N$ be arbitrary.

$$\boxed{\geq \mathsf{j}} := [j, 1] \cap N \qquad \boxed{> \mathsf{j}} := (j, 1] \cap N$$
$$\boxed{\leq \mathsf{j}} := [0, j] \cap N \qquad \boxed{< \mathsf{j}} := [0, j) \cap N$$

Now consider the $(2n - 2)$ C clauses of the following form:

$$\begin{array}{l} \boxed{\leq \mathsf{j}} : p \vee \boxed{> \mathsf{j}} : q \\ \boxed{> \mathsf{j}} : p \vee \boxed{\leq \mathsf{j}} : q \end{array} \quad \text{where } j < 1 \tag{4}$$

Let us denote this clause set with $X_{\leftrightarrow}$; it is easy to prove that $p \leftrightarrow q$ is $\{1\}$satisable iff $X_{\leftrightarrow}$ is satisable and $p \leftrightarrow q$ is $\{0\}$satisable iff $X_{\leftrightarrow}$ is unsatisable.

We have now all prerequisites for mimicking the structure preserving clause form translation described in (1) and (2) in the manyvalued case. We can apply the very same procedure for computing a CNF over Cclauses for some formula ϕ as in the classical case. For each formula of the form (1) we simply expand the signed formula

$$\{1\} : (p_i \leftrightarrow (p_j \text{ op } p_k)) \tag{5}$$

using the inverse tableau rules for $\leftrightarrow$ and op (recall that the conclusion corresponds to a set of Cclauses, namely to $X_{\leftrightarrow}$). This process yields a set $X_{\phi,i}$ for each nonatomic subformula ϕ_i of ϕ. To establish Ssatisability of ϕ for some $S \subseteq N$ it is sufcient to show that

$$X_\phi := \left(\bigcup_i X_{\phi,i} \right) \cup \{S^c : p1\} \tag{6}$$

is unsatisable (where $p1$ is the renaming of ϕ). Since the branching factor of each inverse tableau rule is at most n we have:

Corollary 7. In every $n$valued logic $\mathcal{L}$ for every $S : \phi \in L^*$ there is a CCNF representation X_ϕ of ϕ of length $\mathcal{O}(n^2|\phi|)$ such that $S : \phi$ is valid iff $X_\phi \vdash \Box$.

Let us illustrate the method with an example.

Example 4. Consider threevalued strong Kleene logic with an extra negation '$\sim$' (called $\widetilde{\text{SKL}}$). For convenience we repeat the semantic denitions (which are valid for any number of truth values): $v(\neg\phi) = 1 - v(\phi)$, $v(\sim \phi) = \ulcorner 1 - v(\phi) \urcorner$, $v(\phi \wedge \psi) = \min(v(\phi), v(\psi))$, $v(\phi \vee \psi) = \max(v(\phi), v(\psi))$, $v(\phi \supset \psi) = \max(1 - v(\phi), v(\psi))$.

To establish that $\neg p \supset (\sim p \wedge \neg p)$ is a tautology (in $\widetilde{\text{SKL}}$ both $\frac{1}{2}$ and 1 are designated truth values, i.e. support validity) we need to show that it is a $\{\frac{1}{2}, 1\}$tautology which is the case iff the CCNF of $\{0\} : \neg p \supset (\sim p \wedge \neg p)$ is unsatisable. To simplify things a bit we introduce no new variables for negated atoms. Thus we have

$$\begin{array}{l} \{0\} : q \\ \{1\} : (q \leftrightarrow (\neg p \supset r)) \\ \{1\} : (r \leftrightarrow (\sim p \wedge \neg p)) \end{array} \tag{7}$$

We begin to expand the second formula (cf. (4)):

$$\{1\} : (q \leftrightarrow (\neg p \supset r))$$

$\{\frac{1}{2}, 1\} : q$	$\{0\} : q$	$\{0, \frac{1}{2}\} : q$	$\{1\} : q$
$\{0\} : \neg p \supset r$	$\{\frac{1}{2}, 1\} : \neg p \supset r$	$\{1\} : \neg p \supset r$	$\{0, \frac{1}{2}\} : \neg p \supset r$

The formulæ containing an implication have to be expanded further in order to yield the clause set X_q. Similarly, we compute the set X_r. Together with the rst clause in (7) we arrive at the following set of signed clauses which characterizes the original problem:

1. $\{0\} : q$
2. $\{\frac{1}{2}, 1\} : q \vee \{0\} : r$
3. $\{\frac{1}{2}, 1\} : q \vee \{0\} : p$
4. $\{0\} : q \vee \{\frac{1}{2}, 1\} : p \vee \{\frac{1}{2}, 1\} : r$
5. $\{0, \frac{1}{2}\} : q \vee \{1\} : p \vee \{1\} : r$
6. $\{1\} : q \vee \{0, \frac{1}{2}\} : p$
7. $\{1\} : q \vee \{0, \frac{1}{2}\} : r$
8. $\{1\} : p \vee \{\frac{1}{2}, 1\} : r$
9. $\{0, \frac{1}{2}\} : p \vee \{0\} : r$
10. $\{0\} : p \vee \{0, \frac{1}{2}\} : r$
11. $\{\frac{1}{2}, 1\} : p \vee \{1\} : r$

4 Signed Resolution

We still have to provide a resolution rule for Cclauses. The following rule is one of several possibilities and very close to standard binary resolution:

$$\frac{S_1 : p \vee D_1 \quad \cdots \quad S_m : p \vee D_m}{D_1 \vee \cdots \vee D_m} \qquad \text{if } S_1 \cap \cdots \cap S_m = \emptyset \tag{8}$$

For completeness we need a factoring rule due to similar reasons as in the classical case.

$$\frac{S_1 : p \vee \cdots \vee S_m : p \vee D}{(S_1 \cup \cdots \cup S_m) : p \vee D} \tag{9}$$

Soundness of rules (8) and (9) is straightforward to show. Completeness, too, is not hard to prove with a semantic tree argument that is readily generalized to more than two truth values by allowing $n$ary semantic trees. In [1] a similar result is proved with that method and can readily be adapted to the present case.

Example 5. We continue Example 4 and show that the set of Cclauses generated there is unsatisable:

12.	$[1,2]$	$\{0\} : r$	14.	$[8,12]$	$\{1\} : p$
13.	$[1,3]$	$\{0\} : p$	15.	$[13,14]$	$\square$

Note that only the input clauses 1., 2., 3. and 8. have actually been used in the derivation. We will come back to this issue in the next section.

5 Improvements

We present a simplication which parallels (3). We have already seen in Example 5 that most of the clauses were not redundant. We have to dene a generalized notion of polarity in the presence of more than two truth values.

Denition 8 ManyValued Polarity. Let $S : \phi \in \mathrm{L}^*$ and let $\mathbf{T}$ be a fully expanded inverted tableau for $S : \phi$. For each subformula ψ of ϕ, if the occurrences[2] of ψ in $\mathbf{T}$ are $S_1 : \psi, \ldots, S_m : \psi$, we say that ψ occurs with polarity $R = \langle S_1, \ldots, S_m \rangle$ in ϕ. We abbreviate this fact with $R : \psi \prec S : \phi$.

Note that by denition of inverted tableau rules $\emptyset \not\subseteq S_i \not\subseteq N$ holds for each S_i in R. For each polarity R we dene a binary connective $\Rightarrow_R$ by

$$v(\phi \Rightarrow_R \psi) = \begin{cases} 0 & \text{if } v(\psi) \notin S_i,\ v(\phi) \in S_i,\ S_i \text{ occurs in } R \\ 1 & \text{otherwise} \end{cases}$$

We observe that in twovalued logic $\Rightarrow_{\langle\{1\}\rangle}$ is the same connective as $\supset$ and $\Rightarrow_{\langle\{0\}\rangle}$ is the same connective as $\subset$. Moreover, $\Rightarrow_{\langle\{0\},\{1\}\rangle}$ is the same connective as $\leftrightarrow$. Now we replace (5) by

$$\{1\} : (p_i \Rightarrow_R (p_j \text{ op } p_k)), \text{ if } R : (p_j \text{ op } p_k) \prec S : \phi. \tag{10}$$

In twovalued logic we can get rid of the signs simply by writing everywhere p for $\{1\} : p$ and $\sim p$ for $\{0\} : p$. Together with the observation above (10) collapses into (3) for twovalued logic, if we associate positive polarity with $\langle\{1\}\rangle$, negative polarity with $\langle\{0\}\rangle$ and both polarities with $\langle\{0\},\{1\}\rangle$.

[2] Be aware that identical strings can be different subformulas, such as p in $p \supset p$. On the other hand, the same subformula can occur multiply in the tableau, since they are copied in some rules, such as ϕ in Example 2.

Example 6. Let us apply Denition 8 to (7) from Example 4. Obviously, $\neg p \supset (\sim p \wedge \neg p)$ occurs with polarity $\langle\{0\}\rangle$ in $\{0\} : \neg p \supset (\sim p \wedge \neg p)$. To see the polarity of $\sim p \wedge \neg p$ we begin to compute the inverse tableau for $\{0\} : \neg p \supset (\sim p \wedge \neg p)$.

$$\frac{\{0\} : \neg p \supset (\sim p \wedge \neg p)}{\{1\} : \neg p \,\|\, \{0\} : \sim p \wedge \neg p}$$

We see that the polarity of $\sim p \wedge \neg p$ is $\langle\{0\}\rangle$, too. Therefore, we substitute both occurrences of $\leftrightarrow$ in (7) by $\Rightarrow_{\langle\{0\}\rangle}$:

$$\begin{array}{l} \{0\} : q \\ \{1\} : (q \Rightarrow_{\langle\{0\}\rangle} (\neg p \supset r)) \\ \{1\} : (r \Rightarrow_{\langle\{0\}\rangle} (\sim p \wedge \neg p)) \end{array} \qquad (11)$$

As an easy exercise the reader should verify that the clause set corresponding to $\{1\} : (q \Rightarrow_{\langle\{0\}\rangle} (\neg p \supset r))$ is $\{\{\frac{1}{2}, 1\} : q \vee \{0\} : r, \{\frac{1}{2}, 1\} : q \vee \{0\} : p\}$ and the clause set corresponding to $(r \Rightarrow_{\langle\{0\}\rangle} (\sim p \wedge \neg p))$ is $\{\{\frac{1}{2}, 1\} : r \vee \{1\} : p\}$. But these are exactly clauses 2., 3 and 8. from the old clause set (cf. Example 5) and they were exactly the ones used in the refutation. Hence we succeeded in eliminating all clauses that were redundant in Example 5.

Theorem 9. Let X^1_ϕ be a set of Cclauses corresponding to some $\phi \in L$ computed according to (5) and let X^2_ϕ be a set of Cclauses computed according to (10). Then $X^1_\phi \vdash \Box$ iff $X^2_\phi \vdash \Box$.

Unfortunately, in the worst case the number of generated clauses can still be quadratic wrt the number of truth values (use the same example as before). If, however, the maximal number of occurrences of either ϕ or ψ with different signs in the conclusion of each rule with premise $S : \phi$ op ψ for all signs S and connectives op in a logic is $k \leq n$ we can replace $\mathcal{O}(n^2|\phi|)$ by $\leq k^2|\phi| + l$ for some l in the corollary above. In classical logic we have $k = 1$ if no equivalences are present and $k = n = 2$ otherwise. See [4] for a class of manyvalued logics where $k = 1$.

We suspect that much of the work in [2] can be generalized to the manyvalued case, in particular, it should be possible to prove the optimality of certain manyvalued CNF translations under suitable restrictions on the connectives.

Other possible improvements regard the resolution rule. We state some well-known strategies from classical resolution in our manyvalued setting. This strengthens our claim that signed resolution is a natural extension of twovalued resolution.

Denition 10 Subsumption. Let D, E be two Cclauses. We say that D is subsumed by E iff for each literal $S_1 : p$ in E there is a literal $S_2 : p$ in D such that $S_1 \subseteq S_2$.

Having this denition at hand, we can formulate subsumption strategies as in the classical case.

Our nal point in this section is that among other strategies the setofsupport strategy, as well as a pure rule and deletion of tautologies (clauses containing a literal of the form $N : p$) may be formulated and proved as complete based on our notion of satisability just as in the twovalued case.

6 Related Work

We have seen that it is possible to compute short (C)CNF for nitelyvalued logics in a quite efcient way using truth value sets as signs and inverse tableau rules. The resulting set of signed clauses is attened out and provides no prooftheoretic insight. In [7] (for modal and intuitionistic logic) and [8] (for ukasiewicz logic) a different view is taken: there, a logic is characterized by clause sets whose syntax does not involve signs, instead, certain additional logical connectives like necessity $\Box$ [7] or truncated sum $\vee$ [8] are used. The problem with this approach is that it cannot be done schematically each new logic requires new ideas. Also a new completeness proof of the associated resolution rule has to be carried out each time.

In [1] a similar translation method and resolution rule as developed in Sections 3 and 4 can be found. It is stated in somewhat different terms and, what is more important, uses only single truth values as signs. Also the translation is not linear and does not employ polarity. Thus we conjecture that our own approach is more amenable to implementation. In [9] another variant of signed resolution is investigated.

In [11] a kind of manyvalued polarity is introduced for the purpose of pruning the enormous search space in nonclausal resolution. O'Hearn & Stachniak's polarity notion is dened on unsigned formulæ and close to the original denition of Murray [10]. We do not see any resemblance to the polarity notion developed in this paper.

In [6] the notion of presolution is dened in the context of automated reasoning in paraconsistent logics. These are truth value latticebased logics and resolving between literals involves not only mere set intersection and union as in rules (8,9), but meet and join operations on the truth value lattice. On the other hand, in the case of linear orders of truth values and restriction to signs $\boxed{\geq \mathsf{i}}$, $\boxed{\leq \mathsf{i}}$ presolution essentially coincides with rules (8,9). This fact suggests that our method can be expanded to the treatment of nonclausal paraconsistent logics ([6] assume to have the input in signed CNF).

7 FirstOrder Logic

We consider manyvalued versions[3] of $\forall$, $\exists$ which have the simplied Skolem conditions given in Table 1. Other signs than $\boxed{\geq \mathsf{i}}$, $\boxed{\leq \mathsf{i}}$ are handled by splitting:

$$\frac{S : \phi}{\begin{array}{l} S_1 : \phi \\ \quad \vdots \\ S_k : \phi \end{array}}$$

where $S = S1 \cup \cdots \cup S_k$.

$$\frac{\{i\} : \phi}{\boxed{\geq \mathsf{i}} : \phi \;\Big\|\; \boxed{\leq \mathsf{i}} : \phi}$$

Adding these expansion rules extends the translation method given in Section 3 to rstorder formulæ.

Soundness and completeness proofs can easily be obtained by combining the usual results on Skolemization with the results of [4].

[3] Dened as $v_\beta((\forall y)\phi) = \min\{v_{\beta^u_y}(\phi) | u \in U\}$, $v_\beta((\exists y)\phi) = \max\{v_{\beta^u_y}(\phi) | u \in U\}$, where min, max are interpreted naturally on N and U is the underlying universe.

Table 1. Simplied Skolem rules for quantied formulæ.

$$\frac{\boxed{\geq i} : (\forall x)\phi(x)}{\boxed{\geq i} : \phi(x)} \qquad \frac{\boxed{\leq i} : (\forall x)\phi(x)}{\boxed{\leq i} : \phi(f(x1, \ldots, x_n))}$$

$$\frac{\boxed{\leq i} : (\exists x)\phi(x)}{\boxed{\leq i} : \phi(x)} \qquad \frac{\boxed{\geq i} : (\exists x)\phi(x)}{\boxed{\geq i} : \phi(f(x1, \ldots, x_n))}$$

(f is a new function symbol and $x1, \ldots, x_n$ are the variables which occur freely in the premise of the rule.)

8 Conclusion and Future Work

We presented the rst steps towards an efcient resolution system for manyvalued logics by specifying a way to produce sets of clauses of feasible size which characterize the original problem. The use of short normal form algorithms is not very widespread in classical logic, since they are not really necessary there for most problems. For many valued logics, however, their use becomes essential, in particular when applications like hardware verication is aimed at, where large formulæ are to be expected.

In our CNF algorithm we dened a generalized notion of polarity which allows to remove redundant clauses. This is, however, only the rst part. The next step would be to translate the topdown renamings of [2] in a manyvalued setting. For the resolution procedure we sketched some familiar improvements like subsumption in the many valued version. Further work could include the investigation of successful strategies such as hyperresolution and ordered resolution; see [1] for rst steps in that direction.

References

1. M. Baaz and C. G. Fermüller. Resolution for manyvalued logics. In Proc. LPAR'92, pp. 107 118. Springer, LNAI 624, 1992.
2. T. Boy de la Tour. Minimizing the number of clauses by renaming. In Proc. 10^{th} CADE, Kaiserslautern, pp. 558 572. Springer, Heidelberg, July 1990.
3. R. Hähnle. Towards an efcient tableau proof procedure for multiplevalued logics. In Proc. Workshop Computer Science Logic, Heidelberg, pp. 248 260. Springer, LNCS 533, 1990.
4. R. Hähnle. Uniform notation of tableaux rules for multiplevalued logics. In Proc. 20^{th} ISMVL, Victoria, pp. 238 245. IEEE Press, 1991.
5. R. Hähnle and W. Kernig. Verication of switchlevel circuits with multiplevalued logics. To appear, 1993.
6. J. J. Lu, L. J. Henschen, V.S. Subrahmanian, and N. C. A. da Costa. Reasoning in paraconsistent logics. In Automated Reasoning: Essays in Honor of Woody Bledsoe, pp. 181 210. Kluwer, 1991.
7. G. Mints. Gentzentype systems and resolution rules, part 1: Propositional logic. In Proc. COLOG88, Tallin, pp. 198 231. Springer, LNCS 417, 1990.
8. D. Mundici. Normal forms in innitevalued logic: The case of one variable. In Proc. Workshop Computer Science Logic 91, Berne. Springer, LNCS, 1991.
9. N. Murray and E. Rosenthal. Resolution and pathdissolution in multiplevalued logics. In Proc. ISMIS, Charlotte, 1991.
10. N. V. Murray. Completely non-clausal theorem proving. AI, 18:67 85, 1982.
11. P. O'Hearn and Z. Stachniak. A resolution framework for nitelyvalued rstorder logics. Journal of Symbolic Computing, 13:235 254, 1992.