Information and Computation **162**, 185–205 (2000)
doi:10.1006/inco.1999.2862, available online at http://www.idealibrary.com on IDEAL®

# Optimal Axiomatizations of Finitely Valued Logics[1]



*Technische Universität Wien, Karlsplatz 13, A-1040 Vienna, Austria*

We investigate the problem of finding optimal axiomatizations for operators and distribution quantifiers in finitely valued first-order logics. We show that the problem can be viewed as the minimization of certain propositional formulas. We outline a general procedure leading to optimized operator and quantifier rules for the sequent calculus, for natural deduction, and for clause formation. The main tools are variants of two-valued and many-valued propositional resolution, as well as a novel rule called combination. In the case of operators and quantifiers based on semilattices, rules with a minimal branching degree can be obtained by instantiating a schema, which can also be used for optimal tableaux with sets-as-signs.


## 1. INTRODUCTION

Within the past several years multiple-valued logics, introduced in the 1920s independently by Łukasiewicz and Post, have attracted considerable attention by the computer science community due to their potential in the verification of software and hardware. This has brought about the necessity for automatizing deduction in these logics. Given the specification of a logic, its axiomatization in one's favorite calculus can be done automatically; i.e., rules for every combination of operators and quantifiers with truth values can be obtained by instantiating general schemas. The only flaw with this approach is that the constructed rules are of a high branching degree, resulting in an exploding proof length. Fortunately there are different ways to axiomatize a logic. The problem to solve is how a good—or even better, and optimal—axiomatization can be computed for a given logic.

This work is primarily concerned with the optimization of quantifier rules, but for the sake of completeness we also outline a method for optimizing operator rules. We show that in both cases the problem reduces to the minimization of propositional formulas and we describe general procedures applicable to all calculi based on conjunctive normal forms. For quantifiers based on semilattices we give a general schema yielding optimal axiomatizations. This result also extends to tableaux with

[1] Supported by FWF Grant P10282-MAT.

[2] E-mail: salzer@logic.at.



0890-5401/00 $35.00

sets-as-signs. As a by-product this schema can also be used to obtain optimal rules for operators based on semilattices. This is an important result since most quantifiers (like existential and universal ones) and many operators (like conjunction and disjunction) satisfy the preconditions of our theorems.

The paper is structured as follows. Section 2 defines semilattices and proves a basic result needed later on. It also defines the syntax and semantics of three kinds of logics: finitely valued first-order logics are the reason why we need optimized rules at all; signed formulas are used to define the semantics of rules; and finally, propositional logic is used as a convenient and abstract tool to investigate the optimization of signed formulas. Section 3 gives a precise definition of the problem to be solved and describes our results in more detail, drawing on the notions and notations introduced in Section 2. Section 4 outlines a general optimization procedure for operators. The next section, 5, explains by an example the main idea of using classical (two-valued) propositional formulas for characterizing many-valued distribution quantifiers. Section 6 is devoted to the optimization of these proposition formulas. The results obtained in this context are translated to the level of signed formulas in Section 7, leading to the main theorems. The final section relates our results to work done by others and discusses some consequences.

## 2. PRELIMINARIES

After introducing semilattices we present the notations and notions concerning finitely valued first-order logic, two-valued first-order logic based on signed formulas, and two-valued propositional logic. The order of these sections is determined by their abstraction level. Finitely valued logic is ranked at the bottom as it is the object language of the calculi under consideration. The logic of signed formulas comes second since we use certain signed formulas to characterize the semantics of many-valued calculi. Finally, in spite of being a special case of first-order logic, two-valued propositional logic ranks at the top of the hierarchy: we use propositional variables to abbreviate signed formulas of the form $(\exists x)\, SA(x)$.

### *2.1. Semilattices*

A (lower) *semilattice* is a partially ordered set $\langle W, \leqslant \rangle$ such that any two elements in $W$ have a unique greatest lower bound (glb). We write $w_1 < w_2$ iff $w_1 \leqslant w_2$ and $w_1 \neq w_2$, $w_1 \geqslant w_2$ iff $w_2 \leqslant w_1$, and $w_1 > w_2$ iff $w_2 < w_1$. For $w \in W$, the set of its *successors* is defined as

$$\text{succ}(w) = \{u \in W \mid u > w \text{ and there is no } v \text{ such that } u > v > w\}.$$

An *interval* is a subset $U$ of $W$ such that $U = \{u \in W \mid w_1 \leqslant u \leqslant w_2\}$ or $U = \{u \in W \mid w_1 \leqslant u\}$ for some $w_1, w_2 \in W$.[3] We extend $\leqslant$ to intervals by defining $U \leqslant w$ iff $u \leqslant w$ for some $u \in U$; we write $U < w$ iff $U \leqslant w$ and $w \notin U$.

[3] In the case of a full lattice the second alternative of the definition is subsumed by the first one by choosing the unique top element for $w_2$.

LEMMA 1. *Let $U$ be an interval an let $I$ be a subset of $W$ such that* $\mathrm{glb}(I)$ *exists*[4].

(a) $\mathrm{glb}(I) \not\geqslant U$ *iff* $u \not\geqslant U$ *for some* $u \in I$.

(b) $\mathrm{glb}(I) > U$ *iff there is a successor* $u$ *of* $U$ *such that* $v \geqslant u$ *for all* $v \in I$.

*Proof.* (a) $\mathrm{glb}(I) \not\geqslant U$ iff $\mathrm{glb}(I) \not\geqslant v$ for all $v \in U$, iff there is some $u \in I$ such that $u \not\geqslant v$ for all $v \in U$, iff $u \not\geqslant U$ for some $u \in I$.

(b) $\mathrm{glb}(I) > U$ iff $\mathrm{glb}(I) > w$ for some $w \in U$, iff there is a successor $u$ of $U$ such that $\mathrm{glb}(I) \geqslant u$, i.e., $v \geqslant u$ for all $v \in I$. ▮

## 2.2. *Finitely Valued First-Order Logics*

The language of a first-order logic is based on an *alphabet* $\Sigma$ consisting of mutually disjoint, denumerable sets $P$, $F$, $V$, $O$, and $Q$ of predicate symbols, function symbols, variable symbols, operators, and quantifiers, respectively, as well as of parentheses and commas. With each operator and each predicate and function symbol $s$ a natural number is associated, called the arity of $s$ and denoted by $\mathrm{ar}(s)$.

Let $\mathscr{T}$ denote the set of terms over $F$ and $V$. The set $\mathscr{F}$ of *first-order formulas* (over $\Sigma$) is the smallest set satisfying:

— If $p \in P$ and $t_1, ..., t_{\mathrm{ar}(p)} \in \mathscr{T}$ then $p(t_1, ..., t_{\mathrm{ar}(p)}) \in \mathscr{F}$, called *atom* or *atomic formula.*

— If $\theta \in O$ and $\phi_1, ..., \phi_{\mathrm{ar}(\theta)} \in \mathscr{F}$ then $\theta(\phi_1, ..., \phi_{\mathrm{ar}(\theta)}) \in \mathscr{F}$.

— If $\lambda \in Q$, $x \in V$, and $\phi \in \mathscr{F}$ then $(\lambda x)\, \phi \in \mathscr{F}$.

A *matrix* $M$ for an alphabet $\Sigma$ consists of a nonempty, finite set $W$ of *truth values, truth functions* $\tilde{\theta}$: $W^{\mathrm{ar}(\theta)} \mapsto W$ for every operator $\theta$ in $O$, and *distribution functions* $\tilde{\lambda}$: $(2^W - \{\varnothing\}) \mapsto W$ for every quantifier $\lambda$ in $Q$.[5] An alphabet together with a corresponding matrix defines a $|W|$-*valued first-order logic.*

An *interpretation* for an alphabet $\Sigma$ and a set $W$ of truth values consists of a nonempty set $D$ called *domain*, functions $\tilde{f}$: $D^{\mathrm{ar}(f)} \mapsto D$ for every $f \in F$, predicates $\tilde{p}$: $D^{\mathrm{ar}(p)} \mapsto W$ for every $p \in P$, and values $\tilde{x} \in D$ for every $x \in V$.

Given an interpretation $I$ we define a valuation $\mathrm{val}_I$ assigning a domain element to each term and a truth value to each formula:

— $\mathrm{val}_I(x) = \tilde{x}$ for $x \in V$.

— $\mathrm{val}_I(s(t_1, ..., t_{\mathrm{ar}(s)})) = \tilde{s}(\mathrm{val}_I(t_1), ..., \mathrm{val}_I(t_{\mathrm{ar}(s)}))$ for $s \in F \cup P \cup O$.

— $\mathrm{val}_I((\lambda x)\, \phi) = \tilde{\lambda}(\mathrm{distr}_{I,x}(\phi))$ for $\lambda \in Q$.

$\mathrm{distr}_{I,x}(\phi) = \{\mathrm{val}_{I_d^x}(\phi) \mid d \in D\}$ is called the *distribution* of $\phi$ in $I$, where $I_d^x$ is identical to $I$ except for setting $\tilde{x} = d$.

[4] For the purposes of the paper this condition is void: $W$ is finite, therefore $\mathrm{glb}(I)$ always exists.

[5] These generalized quantifiers were introduced by Mostowski [10], called *distribution quantifiers* by Carnielli [4].

An operator $\theta$ is *based on the semilattice* $\langle W, \leqslant \rangle$ iff $\widetilde{\theta}(w_1, ..., w_{\mathrm{ar}(\theta)}) = \mathrm{glb}(\{w_1, ..., w_{\mathrm{ar}(\theta)}\})$ for all $w_1, ..., w_{\mathrm{ar}(\theta)} \in W$.[6] A quantifier $\lambda$ is *based on the semilattice* $\langle W, \leqslant \rangle$ iff $\widetilde{\lambda}(U) = \mathrm{glb}(U)$ for all $U \subseteq W$ with $U \neq \varnothing$.

EXAMPLE 1. The universal quantifier in classical two-valued logic is based on the (semi)lattice $f < t$, i.e., $\widetilde{\forall}$ is defined by $\widetilde{\forall}(\{t\}) = t$ and $\widetilde{\forall}(\{f\}) = \widetilde{\forall}(\{f, t\}) = f$. Conjunction is an operator based on the same lattice. The existential quantifier and disjunction are both based on the lattice $t < f$.

As a more complicated example consider the four-valued Belnap logic. The set of truth values, $\{f, u, \bot, t\}$, carries the structure of a bilattice with two partial orders $\leqslant_t$ and $\leqslant_k$, defined by $f <_t u$, $\bot <_t t$ and $u <_k t$, $f <_k \bot$. Conjunction and disjunction are defined as the greatest lower and least upper bound w.r.t. $\leqslant_t$, whereas quantifier $\mathsf{U}$ (uniformity) is the least upper bound w.r.t. $\leqslant_k$, i.e., $\tilde{\mathsf{U}}$ is given by $\tilde{\mathsf{U}}(\{u\}) = u$, $\tilde{\mathsf{U}}(\{t\}) = \tilde{\mathsf{U}}(\{t, u\}) = t$, $\tilde{\mathsf{U}}(\{f\}) = \tilde{\mathsf{U}}(f, u) = f$, and $\tilde{\mathsf{U}}(U) = \bot$ if $\bot \in U$ or $U = \{t, f\}$.

## 2.3. Signed Formulas

Let $\mathscr{F}$ be the set of first-order formulas over some alphabet $\Sigma$, and let $W$ be a set of truth values. The set of *signed formulas* over $\Sigma$ and $W$ is inductively defined as

— If $S \subseteq W$ and $\phi \in \mathscr{F}$, then $S\phi$ is a signed formula, called an *atomic* signed formula. $S$ is called *sign*.

— $\bot$ and $\top$ are signed formulas.

— If $\psi_1$ and $\psi_2$ are signed formulas, then $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2$, $\psi_1 \equiv \psi_2$, and $\neg\psi_1$ are signed formulas.

— If $x$ is a variable and $\psi$ is a signed formula, then $(\forall x)\,\psi$ and $(\exists x)\,\psi$ are signed formulas.

We use $\bigwedge_{i=1}^n \psi_i$ and $\bigwedge\{\psi_1, ..., \psi_n\}$ as an abbreviation for $\psi_1 \wedge \cdots \wedge \psi_n$, and $\bigvee_{i=1}^n \psi_i$ and $\bigvee\{\psi_1, ..., \psi_n\}$ for $\psi_1 \vee \cdots \vee \psi_n$, with the understanding that the first equals $\top$ and the second equals $\bot$ if $\{\psi_1, ..., \psi_n\}$ is empty.

Let $I$ be an interpretation for $\Sigma$ and $W$. The semantics of signed formulas is given by a valuation $\mathrm{sval}_I$ assigning either true or false to each signed formula, defined as

— $\mathrm{sval}_I(S\phi) = \mathrm{true}$ iff $\mathrm{val}_I(\phi) \in S$,

— $\mathrm{sval}_I(\top) = \mathrm{true}$ and $\mathrm{sval}_I(\bot) = \mathrm{false}$,

— $\mathrm{sval}_I(\psi_1 \wedge \psi_2) = \mathrm{true}$ iff $\mathrm{sval}_I(\psi_1) = \mathrm{true}$ and $\mathrm{sval}_I(\psi_2) = \mathrm{true}$,
$\mathrm{sval}_I(\psi_1 \vee \psi_2) = \mathrm{false}$ iff $\mathrm{sval}_I(\psi_1) = \mathrm{false}$ and $\mathrm{sval}_I(\psi_2) = \mathrm{false}$,
$\mathrm{sval}_I(\psi_1 \equiv \psi_2) = \mathrm{true}$ iff $\mathrm{sval}_I(\psi_1) = \mathrm{sval}_I(\psi_2)$,
$\mathrm{sval}_I(\neg\psi) = \mathrm{true}$ iff $\mathrm{sval}_I(\psi) = \mathrm{false}$,

— $\mathrm{sval}_I((\forall x)\,\psi) = \mathrm{true}$ iff $\mathrm{sval}_{I_d^x}(\psi) = \mathrm{true}$ for all $d \in D$,
$\mathrm{sval}_I((\exists x)\,\psi) = \mathrm{false}$ iff $\mathrm{sval}_{I_d^x}(\psi) = \mathrm{false}$ for all $d \in D$.
($I_d^x$ is the interpretation identical to $I$ except for setting $\tilde{x} = d$.)

[6] An operator can be defined this way iff it is associative, commutative, and idempotent.

A signed formula $\psi$ is *valid* iff $\mathrm{sval}_I(\psi) = \mathrm{true}$ for all $I$; it is *satisfiable* iff $\mathrm{sval}_I(\psi) = \mathrm{true}$ for some $I$ and *unsatisfiable* otherwise.

It is not hard to see that all the familiar equivalences of classical logic, such as the distribution of $\exists$ over $\vee$, of $\forall$ over $\wedge$, de Morgan's laws etc., also hold for signed formulas. Additionally we have the following tautologies.

LEMMA 2. *The following equivalences are valid*:

(a) $\{\}\phi \equiv \bot$

(b) $W\phi \equiv \top$

(c) $\neg S\phi \equiv (W - S)\phi$

(d) $S_1\phi \vee S_2\phi \equiv (S_1 \cup S_2)\phi$, *in particular* $\{w_1\}\phi \vee \cdots \vee \{w_n\}\phi \equiv \{w_1, ..., w_n\}\phi$

(e) $S_1\phi \wedge S_2\phi \equiv (S_1 \cap S_2)\phi$

(f) $(\forall x)\ S_1\phi \wedge (\exists x)\ S_2\phi \equiv (\forall x)(S_1 \cap S_2)\phi$

(g) $(\exists x)\ S_1\phi \vee (\forall x)\ S_2\phi \equiv (\exists x)\ S_1\phi \vee (\forall x)(S_1 \cup S_2)\phi$.

The third equivalence shows that negations can be completely eliminated from signed formulas. The fourth allows the elimination of all nonsingleton signs by introducing disjunctions. The last two tautologies express that in the presence of $(\forall x)\ S_1\phi$ (resp. $(\exists x)\ S_1\phi$) truth values not occurring in $S_1$ (resp. occurring in $S_1$) can be removed or added *ad libitum* in certain other formulas.

## 2.4. Classical Propositional Logic

Let $W$ be a finite set of propositional variables.[7] The set of propositional formulas over $W$ is the smallest set containing $W$, $\bot$, and $\top$ and being closed under negation, disjunction, and conjunction, i.e., if $F$, $G$ are propositional formulas then so are $\bar{F}$, $F \wedge G$, and $F \vee G$. An interpretation $I$ is any subset of $W$. Each interpretation defines a valuation $\mathrm{pval}_I$ in the usual way, assigning true or false to each propositional formula. In particular, for a propositional variable $w$ we have $\mathrm{pval}_I(w) = \mathrm{true}$ iff $w \in I$. If $\mathrm{pval}_I(F) = \mathrm{true}$ then $I$ is called model of $F$. A formula is called tautology iff it is true in all interpretations.

A literal is either a variable (called positive literal) or the negation of a variable (called negative literal). A clause is a disjunction of literals. A conjunctive normal form (CNF) is a conjunction of clauses; it is complete if every clause contains all variables. A disjunctive normal form (DNF) is a disjunction of conjunctions of literals; it is complete if each conjunction contains all variables. Sometimes we regard clauses as sets and CNFs as sets of clauses. Two clauses (CNFs) are identical iff they are identical as sets. Clause $C$ subsumes clause $D$ iff either $D$ is a tautology or $C \subseteq D$.

An extended literal is either a disjunction of variables (called positive literal) or the negation of a disjunction of variables (called negative literal). Clearly, extended clauses can be viewed as having at most one positive literal, which contains all

[7] The set of truth values and the set of propositional variables are deliberately denoted by the same mathematical symbol $W$: later, a propositional variable $w$ will represent the signed formula $(\exists x)\ \{w\}A(x)$.

unnegated variables. Extended clauses and CNFs are defined as above using extended literals in place of literals. The clauses represented by an extended clause $C$, denoted by $\mathrm{cls}(C)$, are those obtained from $C$ by deleting in each negative literal all but one variable. $C$ subsumes $D$ iff for each $D' \in \mathrm{cls}(D)$ there is a clause $C' \in \mathrm{cls}(C)$ such that $C'$ subsumes $D'$. By $\mathrm{norm}(C)$ we denote the normalized variant of $C$, which is obtained by first removing in the negative literals all variables which also occur in some positive literal and then removing all negative literals whose variables form a superset of the variables in some other negative literal. If one of the negative literals gets empty then $\mathrm{norm}(C) = \top$.

LEMMA 3. *Let $C$ and $D$ be extended clauses.*

(a) *$C$ subsumes $D$ iff either $D$ is a tautology or there is a clause $C'$ obtained from $C$ by removing some variables in its negative literals such that $C' \subseteq \mathrm{norm}(D)$.*

(b) *$C$ is logically equivalent to $\bigwedge \mathrm{cls}(C)$ and to* $\mathrm{norm}(C)$.

(c) *$C$ is a tautology iff* $\mathrm{norm}(C) = \top$.

## 3. THE PROBLEM AND OUR RESULTS

### 3.1. Problem Description

It is a well-known fact that the truth of many-valued formulas can be reduced, in a sense, to the truth of formulas in a two-valued logic. The idea is to use classical logic based on atomic propositions of the form $S\phi$ with the meaning "$\phi$ takes a truth value occurring in $S$"; clearly, such a proposition can be either true or false. It turns out that on the one hand the truth functions and distribution functions of a finitely valued logic can be defined using such formulas and that on the other hand certain normal forms of these formulas can be immediately translated to rules for sequent calculi, for natural deduction, for clause formation, and for tableau systems.

The starting point for the construction of a rule is a two-valued formula characterizing all situations where an operator or a quantifier yields a certain truth value or a certain set of truth values. This approach was first taken in [14], where so-called partial normal forms were used. We choose a slightly more general framework to include also recent developments such as sets-as-signs [6].

DEFINITION 4. Let $\theta$ be an operator, $\lambda$ a quantifier, $A_1, ..., A_{\mathrm{ar}(\theta)}$ nullary predicate symbols, and $A$ a unary predicate symbol. Furthermore, let $S$, $S_{i,j}$, $S_{i,j,k}$ denote signs and $A_{i,j}$ elements of $\{A_1, ..., A_{\mathrm{ar}(\theta)}\}$. A signed formula $\psi$ is called

(a) *CNF for $\theta$ and $S$* iff $\psi = \bigwedge_i \bigvee_j S_{i,j} A_{i,j}$ and $S\theta(A_1, ..., A_{\mathrm{ar}(\theta)}) \equiv \psi$ is valid.

(b) *DNF for $\theta$ and $S$* iff $\psi = \bigvee_i \bigwedge_i S_{i,j} A_{i,j}$ and $S\theta(A_1, ..., A_{\mathrm{ar}(\theta)}) \equiv \psi$ is valid.

(c) *CNF for $\lambda$ and $S$* iff $\psi = \bigwedge_i \bigvee_j \psi_{i,j}$, where $\psi_{i,j} = (\forall x)\, S_{i,j} A(x)$ or $\psi_{i,j} = (\exists x)\, S_{i,j} A(x)$, and $S(\lambda x)\, A(x) \equiv \psi$ is valid.

(d) *DNF for $\lambda$ and $S$* iff $\psi = \bigvee_i \bigwedge_j \psi_{i,j}$, where $\psi_{i,j} = (\forall x)(\bigwedge_k S_{i,j,k} A(x))$ or $\psi_{i,j} = (\exists x)(\bigwedge_k S_{i,j,k} A(x))$, and $S(\lambda x)\, A(x) \equiv \psi$ is valid.

The asymmetry between CNFs and DNFs for quantifiers stems from the fact that for CNFs, expressions of the form $\bigvee_k S_{i,j,k}A(x)$ can be simplified to $S_{i,j}A(x)$ where $S_{i,j} = \bigcup_k S_{i,j,k}$ (see Lemma 2(d)).

The *sequent calculus* and the *natural deduction calculus* are important tools in proof theory. Rules for both calculi are obtained from CNFs as defined above, where $S$ is a singleton set. CNFs for quantifiers are Skolemized preserving validity, i.e., $(\forall x)\, S_{i,j}A(x)$ is replaced by $S_{i,j}A(\alpha)$ where $\alpha$ is a new Skolem constant (called eigenvariable), and $(\exists x)\, S_{i,j}A(x)$ is replaced by $S_{i,j}A(\tau)$ where $\tau$ is a term variable. Proofs for the completeness and correctness of the resulting calculi are given in [3, 15, 18].

*Clause formation calculi* transform arbitrary formulas of some finitely valued logic into clausal form, which then can be used as input to a resolution theorem prover. Clause formation rules use CNFs in a Skolemized form preserving satisfiability. For details see [1].

Another approach to automatizing many-valued logics is *tableau systems*. Tableaux operate directly on first-order formulas and do not require the transformation to some normal form as resolution does. Standard tableaux [4] are based on DNFs where all signs are singletons. DNFs with arbitrary signs correspond to tableaux with sets-as-signs [6]. In both cases the DNFs for quantifiers are Skolemized preserving satisfiability: $\forall$-bound variables become term variables and $\exists$-bound variables become Skolem constants.

EXAMPLE 2. Let the signed formula

$$\psi = (\forall x)\ \{b, d, e, g\} A(x) \wedge (\exists x)\ \{b, d\} A(x) \wedge (\exists x)\ \{b, e\} A(x)$$

be a CNF for quantifier $\lambda$ and sign$\{b\}$. We obtain the following rule of sequent calculus,

$$\frac{\Gamma, A(\alpha)^b, A(\alpha)^d, A(\alpha)^e, A(\alpha)^g \quad \Gamma, A(\tau_1)^b, A(\tau_1)^d \quad \Gamma, A(\tau_2)^b, A(\tau_2)^e}{\Gamma, ((\lambda x)\, A(x))^b},$$

where $\alpha$ denotes an eigenvariable and $\tau_1$, $\tau_2$ denote arbitrary terms. Now $A$ no longer is a unary predicate symbol but a place holder for an arbitrary formula containing the variable $x$.

On the other hand, $\psi$ can also be interpreted as a DNF with sets-as-signs for $\lambda$ and $\{b\}$, consisting of a single conjunction. We obtain the following rule for a tableau system with sets-as-signs;

$$\frac{\{b\}(\lambda x)\, A(x)}{\begin{array}{c}\{b, d, e, g\}\ A(\tau)\\ \{b, d\}\ A(\alpha_1)\\ \{b, e\}\ A(\alpha_2)\end{array}}$$

where $\tau$ denotes a term variable and $\alpha_1$, $\alpha_2$ are constants.

From the arguments above it should be clear that the formulas in Definition 4 are the essence of many-valued calculi. The problem is not to find *any* such formulas. In principle they can be directly read off of the specification of the truth and distribution functions. However, the CNFs and DNFs obtained this way satisfy theoreticians at best: they tend to contain a large number of conjuncts and/or disjuncts, which for quantifiers may be exponential in the number of truth values. Since this number directly corresponds to the branching factor of the rules they are not very useful for actually proving theorems. *The real problem to be solved is to find optimal CNFs and DNFs*, where optimality is to be understood as minimality regarding the number of conjuncts and disjuncts, respectively.

### *3.2. Results*

In this paper we are primarily concerned with CNFs. We describe general procedures for computing optimal CNFs for arbitrary operators and arbitrary distribution quantifiers and arbitrary signs. In general, even optimal CNFs for quantifiers may consist of exponentially many conjuncts (exponential in the number of truth values). However, for the important subclass of semilattice-based quantifiers (containing the usual existential and universal quantifiers) the optimal CNF is immediately—without computations—given by a particular formula $\psi = \bigwedge_i \psi_i$, where $\psi_i$ is either $(\forall x)\, S_i A(x)$ or $(\exists x)\, S_i A(x)$. Arguments similar to those justifying $\psi$ lead to optimal CNFs for semilattice-based operators.

$\psi$ can also be viewed as a simple kind of DNF consisting of a single disjunct. This way we obtain optimal DNFs for all semilattice-based quantifiers and all signs forming an interval. For all other signs we can show that we need at most as many disjuncts as there are truth values in the sign.

As a consequence of our work we have a procedure for computing optimal rules for the sequent calculus, for natural deduction, and for clause formation. For operators and quantifiers based on semilattices, optimal rules are obtained by merely instantiating schemas. In the case of quantifiers based on semilattices our results also extend to tableaux with sets-as-signs, yielding optimal rules there as well.

Finally, this paper also contains a contribution on the methodological level. We introduce propositional formulas as an additional abstraction layer and show that virtually all aspects of quantifier optimization can be discussed in this simpler framework, leading for instance to shorter and better structured proofs. It also provides a direct link between the optimization of Boolean formulas and the optimization of multiple-valued quantifiers, the former being a well-developed field with numerous efficient algorithms [5, 16].

## 4. MINIMIZING OPERATOR RULES

Let $\theta$ be an operator and $\tilde{\theta}$: $W^{\mathrm{ar}(\theta)} \mapsto W$ its truth function, and let $S \subseteq W$ be a sign. We outline a procedure for computing an optimal CNF for $\theta$ and $S$. By Definition 4, this CNF has to be equivalent to $S\theta(A_1, ..., A_{\mathrm{ar}(\theta)})$, i.e., the two formulas must evaluate to the same truth value under all interpretations. The meaning of all symbols except $A_1, ..., A_{\mathrm{ar}}(\theta)$ is fixed; hence an interpretation is completely defined by the tuple $(w_1, ..., w_{\mathrm{ar}(\theta)})$ giving the values for $A_1, ..., A_{\mathrm{ar}(\theta)}$, i.e., $\mathrm{val}_I(A_i) = w_i$. In the following we identify $I$ with this tuple.

Now let $\mathscr{I}$ be the set of all interpretations making $S\theta(A_1, ..., A_{\mathrm{ar}(\theta)})$ true:

$$\begin{aligned}\mathscr{I} &= \{I \mid \mathrm{sval}_I(S\theta(A_1, ..., A_{\mathrm{ar}(\theta)})) = \mathrm{true}\}\\ &= \{(w_1, ..., w_{\mathrm{ar}(\theta)}) \in W^{\mathrm{ar}(\theta)} \mid \tilde{\theta}(w_1, ..., w_{\mathrm{ar}(\theta)}) \in S\}.\end{aligned}$$

Following the ideas of [14] we obtain immediately a CNF for $\theta$ and $S$,

$$\mathrm{cCNF}_{\mathscr{I}} = \bigwedge_{i \notin \mathscr{I}} \bigvee_{i=1}^{\mathrm{ar}(\theta)} (W - I^{(i)})\, A_i,$$

where $I^{(i)}$ denotes the $i$th component of $I$, i.e., $I(i) = w_i$. $\mathrm{cCNF}_{\mathscr{I}}$ is complete in the sense that each disjunction contains a maximal number of truth values for each $A_i$; adding any further truth values would make the disjunction a tautology.

EXAMPLE 3. Let $W = \{f, u, t\}$, let $S = \{f, t\}$, and let $\tilde{\theta}$ be given by the table

| $\tilde{\theta}$ | $f$ | $u$ | $t$ |
|---|---|---|---|
| $f$ | $f$ | $f$ | $f$ |
| $u$ | $f$ | $u$ | $u$ |
| $t$ | $f$ | $u$ | $t$ |

For the set of interpretations making $S\theta(A_1, A_2)$ true we obtain

$$\mathscr{I} = \{(f, f), (f, u), (f, t), (u, f), (t, f), (t, t)\}$$

and therefore

$$\begin{aligned}\mathrm{cCNF}_{\mathscr{I}} = {} & (\{f, t\}\, A_1 \vee \{f, t\}\, A_2) \wedge\\ & (\{f, t\}\, A_1 \vee \{f, u\}\, A_2) \wedge\\ & (\{f, u\}\, A_1 \vee \{f, t\}\, A_2).\end{aligned}$$

LEMMA 5. *Let $\mathscr{I}$ be defined as above. Then* $\mathrm{cCNF}_{\mathscr{I}}$ *is a CNF for $\theta$ and $S$.*

*Proof.* We show that $\mathrm{sval}_I(\mathrm{cCNF}_{\mathscr{I}}) = \mathrm{false}$ iff $I \notin \mathscr{I}$, the latter being equivalent to $\mathrm{sval}_I(S\theta(A_1, ..., A_{\mathrm{ar}(\theta)})) = \mathrm{false}$. Observe that each disjunction $\bigvee_{i=1}^{\mathrm{ar}(\theta)} (W - I^{(i)})\, A_i$ is falsified by exactly one interpretation, namely $I$. Hence, if $I \notin \mathscr{I}$ then $I$ falsifies the corresponding disjunction and therefore the whole formula. Conversely, assume that $\mathrm{cCNF}_{\mathscr{I}}$ evaluates to false. Then some disjunction has to be false. But each clause is falsified only by "its" interpretation, which is not in $\mathscr{I}$. ∎

$\mathrm{cCNF}_{\mathscr{I}}$ is maximal: there is no CNF for $\theta$ and $S$ with more conjuncts than $\mathrm{cCNF}_{\mathscr{I}}$. To find a small or even minimal CNF we use many-valued propositional resolution with sets-as-sings [11]. Given two disjunctions (i.e., clauses) $C \vee SA$ and

$S'A \vee D$, resolution infers the new clause $C \vee (S \cap S')\, A \vee D$. Literals of the form $\{\}\, A$ are equivalent to false and can be removed from clauses. A literal $WA$, on the other hand, is equivalent to true and makes the clause a tautology: the whole clause can be removed from the CNF. Finally, expressions $SA \vee S'A$ can be factorized to $(S \cup S')\, A$. (See also Lemma 2.)

Resolution usually derives many clauses which either themselves are redundant or which make other clauses redundant. An indispensable tool for detecting redundancy is the *subsumption* principle. A signed clause $C$ subsumes a clause $D$ if for every literal $SA$ in $C$ there is a literal $S'A$ in $D$ such that $S \subseteq S'$; subsumed clauses are redundant and may be deleted.

Resolution is complete in the following sense: given a CNF $\mathscr{C}$ (i.e., a set of clauses) and a nontautological clause $D$ logically following from $\mathscr{C}$, it is possible to derive from $\mathscr{C}$ a clause $C$ using resolution (deleting tautologies and subsumed clauses on the way) such that $C$ subsumes $D$. This leads us to the following optimization procedure.

*Step* 1. *Construct any CNF for $\theta$ and $S$; call it $\mathscr{C}_1$*. If $\tilde{\theta}$ is given by a truth table, choose $\mathrm{cCNF}_{\mathscr{I}}$ for $\mathscr{C}_1$. If $\tilde{\theta}$ is specified by some Boolean expression it might be easier to obtain a CNF from this specification directly instead of constructing the truth table explicitly. The next step does not require that the CNF is maximal.

*Step* 2. *Saturate $\mathscr{C}_1$ under resolution and remove tautologies as well as subsumed clauses; call the saturated set $\mathscr{C}_2$*. If $\mathscr{C}_1$ is maximal, i.e., $\mathscr{C}_1 = \mathrm{cCNF}_{\mathscr{I}}$, then the application of the resolution rule can be restricted to parent clauses differing only in the resolved literals: we only need to resolve clauses of the form $C \vee SA$ and $S'A \vee C$, leading to the clause $C \vee (S \cap S')\, A$. This way we only derive clauses smaller than the parent clauses; all other longer resolvents are either tautologies or have already been derived in an earlier stage. Moreover, $C \vee (S \cap S')\, A$ subsumes both of its parent clauses, which may be deleted in the end. This procedure is a direct generalization of the Quine–McCluskey algorithm [9, 13] to the many-valued case.

EXAMPLE 4. Let $\mathscr{C}_1$ be the CNF from Example 3 containing three clauses. There are four nontrivial resolvents. Resolving the first and the second clause we obtain $C_1 = \{f, t\}\, A_1 \vee \{f\}\, A_2$; the first and the third clause lead to $C_2 = \{f\}\, A_1 \vee \{f, t\}\, A_2$; and finally, the last two clauses give rise to two resolvents, $\{f, t\}\, A_1 \vee \{f\}\, A_2 \vee \{f, u\}\, A_1$ and $\{f, u\}\, A_2 \vee \{f\}\, A_1 \vee \{f, t\}\, A_2$, which simplify to $C_3 = \{f, u, t\}\, A_1 \vee \{f\}\, A_2$ and $C_4 = \{f, u, t\}\, A_2 \vee \{f\}\, A_1$, respectively. $C_3$ and $C_4$ are tautologies and can be deleted; we could have avoided computing them from the outset since $\mathscr{C}_1$ is a maximal CNF and the last two clauses of $\mathscr{C}_1$ differ in more than one literal. $C_1$ and $C_2$ subsume all clauses in $\mathscr{C}_1$, which can be deleted. Thus we obtain the simpler CNF $\mathscr{C}_2 = C_1 \wedge C_2$, which is equivalent to $\mathscr{C}_1$.

By the completeness of many-valued resolution, the CNF $\mathscr{C}_2$ obtained in Step 2 is complete in the sense that any of its logical consequences either are tautologies or are subsumed by some clause in $\mathscr{C}_2$. Furthermore, $C_2$ contains no redundant clauses w.r.t. subsumption. However, $C_2$ is not minimal in a global sense: some subset may imply the remaining clauses. This is a well-known phenomenon in the optimization of two-valued circuits.

EXAMPLE 5. Let $W = \{f, t\}$. Consider the CNF

$$(\{t\}A_1 \vee \{t\}A_3) \wedge (\{f\}A_1 \vee \{f\}A_2) \wedge (\{t\}A_1 \vee \{t\}A_2) \wedge (\{f\}A_2 \vee \{t\}A_3).$$

It is complete and minimal in the sense described above: it is saturated under resolution and contains neither tautologies nor subsumed clauses. However, it is not optimal: the first two clauses imply the fourth one, and the last two clauses imply the first one. Hence there are two optimal CNFs, one consisting of the first three clauses and one consisting of the last three clauses.

As another example, consider the CNF

$$(\{t\}A_1 \vee \{t\}A_3) \wedge (\{f\}A_1 \vee \{f\}A_2) \wedge (\{f\}A_2 \vee \{t\}A_3),$$

which is again saturated under resolution and contains no redundancies. The unique optimal CNF consists of the first two clauses, which imply the last one. This shows that the CNF obtained in Step 2 may contain clauses which are in no optimal CNF at all.

*Step* 3. *Select a subset* $\mathscr{C}_3$ *of the clauses in* $\mathscr{C}_2$ *which implies all other clauses in* $\mathscr{C}_2$ *and which contains a minimal number of clauses.* Several heuristics and strategies have been devised to address the problem of selecting minimal or nearly minimal subsets (see, e.g., [5, 12, 16]). A generalization to finitely valued logics seems possible in many cases, but is beyond the scope of the paper. Work in this direction can be found in [6, 7].

Using the arguments of Rousseau [15] one can show that for every operator $\theta$ and every sign there is a CNF consisting of at most $|W|^{(\mathrm{ar}(\theta)-1)}$ conjuncts; furthermore, there is an operator and a sign such that its minimal CNF has exactly $|W|^{(\mathrm{ar}(\theta)-1)}$ conjuncts. As we will see in Sections 6.2 and 7, there is a class of operators with at most $|W|-1$ conjuncts, namely lattice-based operators, which include the usual conjunction and disjunction operators.

## 5. MINIMIZATION OF QUANTIFIER RULES

Similar to operators, a CNF for a quantifier $\lambda$ and a sign $S$ can ben constructed automatically from the distribution function $\tilde{\lambda}$. In general its length is exponential in the number of truth values. One way to minimize the CNF could be to perform similar steps as described in the last section for operators: transform the CNF to pure clausal form by Skolemizing existential quantifiers and then saturate the clause set using general first-order resolution, eliminating at the same time subsumed clauses and tautologies. Finally, translate the result back to a quantified formula.

However, this approach is unnecessarily complicated. This becomes apparent when looking at the clauses obtained from the signed formula: there is only one unary predicate symbol, and all terms are either variables or constants. First-order resolution seems too strong a tool for such simple clauses. Indeed, it turns out that CNFs for quantifiers can be translated to certain *propositional* formulas, viz. conjunctions of extended clauses, in which truth values function as propositional variables. We illustrate this approach by an example before turning to the optimization of such propositional formulas.

Example 6. Let $W = \{f, u, t\}$, and let the quantifier $\lambda$ be defined by the distribution function

| $U$ | $\tilde{\lambda}(U)$ |
|---|---|
| $\{f, u, t\}$ | $f$ |
| $\{f, u\}$ | $f$ |
| $\{f, t\}$ | $f$ |
| $\{f\}$ | $f$ |

| $U$ | $\tilde{\lambda}(U)$ |
|---|---|
| $\{u, t\}$ | $u$ |
| $\{u\}$ | $u$ |
| $\{t\}$ | $t$ |

Now consider the meaning of a table entry like $\tilde{\lambda}(\{f, u\}) = f$. It defines the result of the quantifier for exactly that situation where $A(x)$ evaluates to $f$ and $u$ for some domain elements, but never takes value $t$. More formally, this situation can be characterized by the signed formula:

$$(\exists x)\{f\} A(x) \wedge (\exists x)\{u\} A(x) \wedge \neg(\exists x)\{t\} A(x).$$

The only essential, nonredundant parts of this formula are the truth values and the negation sign; we could write as well $f \wedge u \wedge \bar{t}$, taking $w$ as an abbreviation for $(\exists x)\{w\} A(x)$ and marking negation by a bar. Now it is easy to obtain a propositional DNF characterizing $\{u, t\}(\lambda x)\, A(x)$, for instance. Every distribution resulting in $u$ or $t$ contributes one conjunction to the formula:

$$(\bar{f} \wedge u \wedge t) \vee (\bar{f} \wedge u \wedge \bar{t}) \vee (\bar{f} \wedge \bar{u} \wedge t).$$

To obtain a CNF, one can either apply the distribution law to the DNF, which usually is expensive, or one could construct a DNF characterizing the distributions *not* resulting in $u$ or $t$, and then apply de Morgan's law:

$$(\bar{f} \vee \bar{u} \vee \bar{t}) \wedge (\bar{f} \vee \bar{u} \vee t) \wedge (\bar{f} \vee u \vee \bar{t}) \wedge (\bar{f} \vee u \vee t).$$

Now optimization can take place on the propositional, two-valued level. Afterward the optimized propositional formula is again expanded to a signed formula. In our example the four propositional clauses collapse to a single clause containing a single variable: $\bar{f}$. The corresponding signed formula is $\neg(\exists x)\{f\} A(x)$, saying that $(\lambda x)\, A(x)$ takes a value in $\{u, t\}$ iff for no $x$, $A(x)$ evaluates to $f$.

There is one more thing to pay attention to. By definition, a CNF for a quantifier and a sign may not contain negations. A formula like $\neg(\exists x)\{f\}\, A(x)$ has to be replaced by $(\forall x)(W - \{f\})\, A(x)$ (justified by Lemma 2(c)). This distorts the direct correspondence between propositional level and signed formulas: in general, $(\forall x)(W - \{w_1, ..., w_n\})\, A(x)$ corresponds to $\overline{w_1 \vee \cdots \vee w_n}$. In other words, our goal is not to optimize plain propositional CNFs, but CNFs with *extended literals* (see Section 2.4).

The next section discusses the construction of minimal CNFs with extended literals for arbitrary sets of distributions as well as for sets based on semilattices. Section 7 then defines the correspondence between the propositional and the signed level in a rigorous way and states the main theorems.

## 6. MINIMAL PROPOSITIONAL FORMULAS FOR SETS OF DISTRIBUTIONS

In this section we characterize sets of distributions with the help of propositional formulas: truth values are interpreted as propositional variables and distributions as propositional interpretations. We show how to compute minimal extended conjunctive normal forms for arbitrary sets of distributions. For certain sets, namely those defined via semilattices, no computations are necessary at all: the minimal normal form can be obtained by instantiating a schema.

### *6.1. Minimal CNFs for Arbitrary Sets of Interpretations*

Let $\mathscr{I}$ be an arbitrary set of propositional interpretations, either explicitly given as a set of sets of variables or implicitly specified by a propositional formula which is true in an environment iff it is in $\mathscr{I}$. We describe a four-step procedure for constructing a minimal extended CNF having exactly the interpretations in $\mathscr{I}$ as models.

*Step* 1. *Construct a CNF for* $\mathscr{I}$; *call it* $\mathscr{C}_1$. If $\mathscr{I}$ is specified by a formula, transform it to CNF using the usual propositional laws, calling the result $\mathscr{C}_1$. Otherwise let $\mathscr{C}_1 = \mathrm{cCNF}_{\mathscr{I}}$ where

$$\mathrm{cCNF}_{\mathscr{I}} = \bigwedge_{I \notin \mathscr{I}} \Bigl( \bigvee_{w \in I} \bar{w} \vee \bigvee_{w \notin I} w \Bigr).$$

Lemma 6. $\mathrm{cCNF}_{\mathscr{I}}$ *is a complete CNF, and* $\mathrm{pval}_I(\mathrm{cCNF}_{\mathscr{I}}) = \mathrm{true}$ *iff* $I \in \mathscr{I}$.

*Proof.* Obviously, $\mathrm{cCNF}_{\mathscr{I}}$ is in CNF and each clause contains all variables. We show that $\mathrm{pval}_I(\mathrm{cCNF}_{\mathscr{I}}) = \mathrm{false}$ iff $I \notin \mathscr{I}$. Observe that each clause $\bigvee_{w \in I} \bar{w} \vee \bigvee_{w \notin I} w$ is falsified by exactly one interpretation, namely $I$. So, if $I \notin \mathscr{I}$ then $I$ falsifies the corresponding clause and therefore the whole formula. Conversely, assume $\mathrm{cCNF}_{\mathscr{I}}$ evaluates to false. Then some clause has to be false. But each clause is falsified only by "its" interpretation, which is not in $\mathscr{I}$. ∎

*Step* 2. *Saturate* $\mathscr{C}_1$ *under resolution and remove tautologies as well as subsumed clauses*; *call the saturated set* $\mathscr{C}_2$. If $\mathscr{C}_1$ is a complete CNF and resolution is applied using the level saturation strategy, then only resolvents need to be computed which have strictly fewer literals than their parent clauses.[8]

$\mathscr{C}_2$ is minimal and complete in the following sense: no clause in $\mathscr{C}_2$ is subsumed by any other clause in $\mathscr{C}_2$, and any clause, which is a logical consequence of $\mathscr{C}_2$, is subsumed by some clause in $\mathscr{C}_2$ (this follows from the completeness of resolution).[9]

[8] In this case the saturation process coincides with the well-known Quine–McCluskey algorithm [9, 13].

[9] Note that by our definition of subsumption, tautologies are subsumed by any clause.

For the description of the next step we introduce an inference rule called *combination*.

DEFINITION 7 (Combination). Let $C_1 = \bigvee\{\overline{\bigvee M_i} \mid 1 \leqslant i \leqslant m\} \vee \bigvee P$ and $C_2 = \bigvee\{\overline{\bigvee N_j} \mid 1 \leqslant j \leqslant n\} \vee \bigvee Q$ be extended clauses, where $M_i$, $N_j$, $P$, and $Q$ are sets of variables. The extended clause $\bigvee\{\overline{\bigvee(M_i \cup N_j)} \mid 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n\} \vee \bigvee(P \cup Q)$ is called the *combination* of $C_1$ and $C_2$.

LEMMA 8. *Let $C$ be the combination of $C_1$ and $C_2$.*

(a) *Combination is associative and commutative.*

(b) *$C$ is a logical consequence of $C_1$ and $C_2$.*

(c) *If $C_1$ and $C_2$ contain the same positive literals then $C_1$ and $C_2$ both are logical consequences of $C$; i.e., in this case $C_1 \wedge C_2$ is logically equivalent to $C$.*

EXAMPLE 7. The clauses $\bar{p}$ and $\bar{q} \vee \bar{r}$ can be combined to the extended clause $\overline{p \vee q} \vee \overline{p \vee r}$, which subsumes the former clauses.

The CNF $(\bar{a} \vee \bar{c} \vee e) \wedge (\bar{a} \vee \bar{d} \vee e) \wedge (\bar{b} \vee \bar{c} \vee e) \wedge (\bar{b} \vee \bar{d})$ can be compressed to the equivalent extended CNF $(\overline{a \vee b} \vee \overline{c \vee d} \vee e) \wedge (\bar{b} \vee \bar{d})$. The extended clause is obtained by combining all four clauses in the original CNF and normalizing the result. Since $\bar{b} \vee \bar{d}$ is not subsumed by the extended clause it has to be retained.

*Step* 3. *Saturate $\mathscr{C}_2$ under combination while keeping clauses in normalized form; remove all subsumed clauses; call the resulting set $\mathscr{C}_3$.* In the saturation process we can use Lemma 8(a) to avoid computing the same extended clause several times in different ways.

$\mathscr{C}_3$ is minimal and complete in the following sense: no clause in $\mathscr{C}_3$ is subsumed by any other clause in $\mathscr{C}_3$, and any extended clause, which is a logical consequence of $\mathscr{C}_3$, is subsumed by some clause in $\mathscr{C}_3$. To see this, assume $C$ is a logical consequence of $\mathscr{C}_3$. By the completeness of $\mathscr{C}_2$, each clause in cls($C$) is subsumed by some clause in $\mathscr{C}_2$. Let $D$ be the combination of the latter clauses. $D$ subsumes $C$ and is computed in step 3. So either $D$ or a clause subsuming $D$ is in $\mathscr{C}_3$.

*Step* 4. *Select a subset $\mathscr{C}_4$ of the clauses in $\mathscr{C}_3$ which implies all other clauses in $\mathscr{C}_3$ and which satisfies some criteria for optimality, like minimal number of clauses.* In principle one could check all subsets of $\mathscr{C}_3$. There are, however, more efficient methods taking into account the derivation history of the clauses (see, e.g., [5, 12, 16]).

The last step requires considerable search efforts, especially if one is interested in global optimality. Heuristics and further redundancy elimination might speed up the process.

How many clauses can a minimal extended CNF contain? The following two lemmas show that the tight upper bound is $2^{|W|-1}$.

LEMMA 9. *For any set of interpretations over $W$ an extended CNF minimal in the number of clauses contains at most $2^{|W|-1}$ clauses.*

*Proof.* Let $\mathscr{I}$ be a set of $2^n - m$ interpretations, where $n = |W|$. cCNF$_{\mathscr{I}}$ consists of $m$ clauses each containing $n$ literals. Suppose $m > 2^{n-1}$. Let $w$ be some arbitrary, but fixed, propositional variable. Disregarding the (negated or unnegated) occurrences of $w$ in $\mathrm{cCNF}_{\mathscr{I}}$ there are at most $2^{n-1}$ different clauses in $\mathrm{cCNF}_{\mathscr{I}}$. By the pigeonhole principle there have to be at least $m-2^{n-1}$ clauses of the form $C \vee w$ such that also $C \vee \bar{w}$ occurs in $\mathrm{cCNF}_{\mathscr{I}}$. But $(C \vee \bar{w}) \wedge (C \vee \bar{w})$ is logically equivalent to $C$. By replacing all pairs $C \vee w$ and $C \vee \bar{w}$ by $C$ we obtain a CNF logically equivalent to $\mathrm{cCNF}_{\mathscr{I}}$, which has $2^{n-1}$ or fewer clauses. Since each CNF is also an extended CNF, we are done. ∎

Lemma 10. *There is a set of interpretations over $W$ such that its minimal extended CNF contains $2^{|W|-1}$ clauses.*

*Proof.* Let $\mathscr{I} = \{I \mid |I| \text{ even}\}$. We show that $\mathrm{cCNF}_{\mathscr{I}}$ is an extended CNF minimal in the number of clauses. Observe the following facts:

— $\mathrm{cCNF}_{I}$ contains $2^n - |\mathscr{I}| = 2^n - 2^{n-1} = 2^{n-1}$ clauses, where $n = |W|$.

— $\mathrm{cCNF}_{\mathscr{I}}$ contains no redundant clauses since every interpretation not in $\mathscr{I}$ falsifies exactly one clause of $\mathrm{cCNF}_{\mathscr{I}}$.

— Every clause containing more than $n$ literals is a tautology.

— Every nontautological clause of length $n$ is either in $\mathrm{cCNF}_{\mathscr{I}}$ or it is falsified by an interpretation in $\mathscr{I}$ and therefore cannot appear in any CNF for $\mathscr{I}$.

— Every nontautological clause with fewer than $n$ literals is falsified by some interpretation in $\mathscr{I}$ and therefore cannot appear in any CNF. To see this let $C$ be such a clause, let $w$ be a variable not occurring in $C$, and let $I'$ be an interpretation falsifying $C$. Consider the interpretation $I$ defined by $I = I'$ if $|I'|$ even and $I = I' \cup \{w\}$ otherwise. $I$ falsifies $C$ in the same way as $I'$, but by its construction $I \in \mathscr{I}$.

— Every combination of clauses in $\mathrm{cCNF}_{\mathscr{I}}$ is a tautology. For any two clauses in $\mathrm{cCNF}_{\mathscr{I}}$ the set of positive literals is different. The combination operation takes the union of all positive literals, i.e., the combination of two clauses contains more positive literals than the clauses themselves. But then all clauses represented by the combination have more than $n$ literals and therefore are tautologies.

From the arguments above it follows that $\mathrm{cCNF}_{\mathscr{I}}$ is saturated under resolution, combination, and subsumption, and no clause can be removed without changing the set of interpretations satisfying the formula. In other words, $\mathrm{cCNF}_{\mathscr{I}}$ is the unique minimal extended CNF for $\mathscr{I}$. ∎

In spite of the exponential upper bound for the number of conjuncts experience shows that most relevant distribution quantifiers—i.e., sets of interpretations in our propositional terminology—can be specified in a compact manner using semilattices and that the corresponding inference rules (CNFs) are rather small.

### *6.2. Minimal CNFs for Sets of Interpretations Based on Semilattices*

Suppose the set of propositional variables forms a semilattice $\langle W, \leqslant \rangle$, and suppose $\mathscr{I}$ is defined as $\{I \subseteq W \mid \mathrm{glb}(I) \in U\}$ for some interval $U \subseteq W$. Let

$$\mathrm{mCNF}(U) = \bigwedge \{ C_u^- \mid u \ngeqslant U \} \wedge \bigwedge \{ C_u^+ \mid u \in \mathrm{succ}(U) \},$$

where $C_u^- = \bar{u}$ and $C_u^+ = \bigvee \{ v \mid v \ngeqslant u, v \geqslant U \}$.

**Theorem 11.** mCNF($U$) *is the unique minimal CNF for* $\mathscr{I}$, *i.e.*:

(a) *Correctness*: $\text{glb}(I) \in U$ *iff* $\text{pval}_I(\text{mCNF}(U)) = \text{true}$.

(b) *Minimality*: *among all CNFs for* $\mathscr{I}$, mCNF($U$) *is the unique CNF containing the least number of clauses and the least number of literals.*

*Proof.* (a) We show that $\text{glb}(I) \notin U$ iff $\text{pval}_I(\text{mCNF}(U)) = \text{false}$. $\text{glb}(I) \notin U$ may hold for one of two reasons.

$\text{glb}(I) \not\geqslant U$. By Lemma 1(a) this is equivalent to $u \not\geqslant U$ for some $u \in I$, i.e., $\text{pval}_I(C_u^-) = \text{false}$ for some $u \in I$.

$\text{glb}(I) > U$. By Lemma 1(b) there is a successor $u$ of $U$ such that $v \geqslant u$ for all $v \in I$, i.e., none of the variables in the set $\{v \mid v \not\geqslant u, v \geqslant U\}$ occurs in $I$. But this is equivalent to $\text{pval}_I(C_u^+) = \text{false}$.

Putting all pieces together we have: $\text{glb}(I) \notin U$ iff $\text{glb}(I) \not\geqslant U$ or $\text{glb}(I) > U$, iff $\text{pval}_I(C_u^-) = \text{false}$ for some $u \not\geqslant U$ or $\text{pval}_I(C_u^+) = \text{false}$ for some $u \in \text{succ}(U)$, iff $\text{pval}_I(\text{mCNF}(U)) = \text{false}$.

(b) We first show that mCNF($U$) contains no redundant clauses, i.e., for every clause $C \in \text{mCNF}(U)$ there is an interpretation $I$ falsifying $C$, but no other clause in mCNF($U$). We distinguish two cases.

$C = C_u^-$ *for some* $u \not\geqslant U$. Let $I = \{v \mid v \not\geqslant U, v \neq u\}$. Clearly, $C$ is the only clause in mCNF($U$) falsified by $I$.

$C = C_u^+$ *for some* $u \in \text{succ}(U)$. Let $I = \{v \mid v \geqslant u\}$. By definition none of the variables in $C$ occurs in $I$; hence $\text{pval}_I(C) = \text{false}$. On the other hand, $\text{pval}_I(C_{u'}^+) =$ true for all successors $u'$ different from $u$: we have $u \not\geqslant u'$, $u \geqslant U$, and $u \geqslant u$; i.e., $u$ occurs in $C_{u'}^+$ as well as in $I$. Furthermore, none of the variables $v \not\geqslant U$ is in $I$ since $v \geqslant u$ would imply $v > U$; therefore $\text{pval}_I(C_v^-) = \text{true}$.

Consequently, none of the clauses in mCNF($U$) is redundant.

Observe that mCNF($U$) is saturated under resolution. By the completeness of resolution, every clause being a logical consequence of mCNF($U$) either is a tautology or is subsumed by some clause in mCNF($U$). Now let $F$ be another minimal CNF containing fewer clauses. Each clause in $F$ is subsumed by some clause in mCNF($U$). Let $F'$ be the set of these clauses. $F'$ is logically equivalent to mCNF($U$) and is a proper subset of mCNF($U$). This, however, implies that some clause of mCNF(U) is redundant. Contradiction.

Finally, suppose $F'$ is a CNF with the same number of clauses as mCNF($U$), but with fewer literals. Every clause in mCNF($U$) subsumes exactly one clause in $F'$; therefore the latter cannot have fewer literals than the former. Contradiction. ∎

**Corollary 12.** $\text{meCNF}(U) = \overline{C^- \wedge \bigwedge\{C_u^+ \mid u \in \text{succ}(U)\}}$ *is the unique minimal extended CNF for* $\mathscr{I}$, *where* $C^- = \bigvee\{u \mid u \not\geqslant U\}$.

Sometimes one is interested in maximizing the number of literals per clause, while keeping the number of clauses at a minimum.

COROLLARY 13. *Let* mCNF′($U$) *be obtained from* mCNF($U$) *by replacing* $C_u^+$ *by* $D_u^+$, *where* $D_u^+ = \bigvee\{v \mid v \ngeqslant u\}$. mCNF′($U$) *is the uniquely determined CNF for* $\mathscr{I}$ *with the most literals among all CNFs with a minimal number of clauses.*

*Proof.* Obviously, $D_u^+ = C_u^+ \cup \{v \mid v \geqslant U\}$. Saturating mCNF′($U$) under resolution and subsumption again yields mCNF($U$) since the additional literals can be resolved away using $C_v^-$. Therefore the clause sets are logically equivalent. Clearly, they also have the same number of clauses. Concerning the maximality of literals, observe that adding further literals to the clauses in mCNF′($U$) leads to a clause set which is no longer equivalent to mCNF($U$): the additional literals cannot be removed by resolution and therefore also appear in the saturated clause set. ∎

Between mCNF($U$) and mCNF′($U$) we have a spectrum of CNFs for $\mathscr{I}$, which are all minimal in the number of clauses, but differ in the number of positive literals.

## 7. FROM PROPOSITIONAL LOGIC TO SIGNED FORMULAS

The following lemma provides the link between propositional and signed formulas.

LEMMA 14. *Let* $A_1, ..., A_n$ *be nullary predicate symbols and* $A$ *be a unary one, let* $W$ *be the set of all truth values, and let* $S$ *be a sign. For all first-order interpretations* $I$ *we have*:

(a) $(\exists x)\, SA(x)$ *is true in* $I$ *iff* $\bigvee S$ *is true in* $\mathrm{distr}_{I,x}(A(x))$.

(b) $(\forall x)(W-S)\, A(x)$ *is true in* $I$ *iff* $\overline{\bigvee S}$ *is true in* $\mathrm{distr}_{I,x}(A(x))$.

(c) $V_{i=1}^n\, SA_i$ *is true in* $I$ *iff* $\bigvee S$ *is true in* $\{\mathrm{val}_i(A_i) \mid 1 \leqslant i \leqslant n\}$.

(d) $\bigwedge_{i=1}^n (W-S)\, A_i$ *is true in* $I$ *iff* $\overline{\bigvee S}$ *is true in* $\{\mathrm{val}_I(A_i) \mid 1 \leqslant i \leqslant n\}$.

*Proof.* Let $I' = \mathrm{distr}_{I,x}(A(x))$ and $I'' = \{\mathrm{val}_I(A_i) \mid 1 \leqslant i \leqslant n\}$.

(a) $\mathrm{sval}_I((\exists x)\, SA(x)) = \mathrm{false}$ iff $\mathrm{sval}_{I_d^x}(SA(x)) = \mathrm{false}$ for all $d \in D$, iff $\mathrm{val}_{I_d^x}(A(x)) \notin S$ for all $d \in D$, iff $I' \cap S = \varnothing$, iff $\mathrm{pval}_{I'}(\bigvee S) = \mathrm{false}$.

(b) Using Lemma 2(c) we obtain:
$\mathrm{sval}_I((\forall x)(W-S)\, A(x)) = \mathrm{true}$ iff $\mathrm{sval}_I((\forall x) \neg SA(x)) = \mathrm{true}$,
iff $\mathrm{sval}_I(\neg(\exists x)\, SA(x)) = \mathrm{true}$, iff $\mathrm{sval}_I((\exists x)\, SA(x)) = \mathrm{false}$,
iff $\mathrm{pval}_{I'_n}(\bigvee S) = \mathrm{false}$, iff $\mathrm{pval}_{I'}(\overline{\bigvee S}) = \mathrm{true}$.

(c) $\mathrm{sval}_I(\bigvee_{i=1}^n SA_i) = \mathrm{false}$ iff $\mathrm{val}_I(A_i) \notin S$ for $1 \leqslant i \leqslant n$,
iff $I'' \cap S = \varnothing$, iff $\mathrm{pval}_{I''}(\bigvee S) = \mathrm{false}$.

(d) Analogous to (b). ∎

Putting all pieces together we obtain the following theorems.

THEOREM 15. *Let* $\lambda$ *be a quantifier and* $\tilde{\lambda}$ *be its distribution function. Let* $S$ *and* $S_{i,j}$ *denote signs. Let* $\mathscr{I}$ *be the set of all distributions* $U$ *such that* $\tilde{\lambda}(U) \in S$, *i.e.*, $\mathscr{I} = \{U \subseteq W \mid \tilde{\lambda}(U) \in S\}$.

*If the propositional formula* $F = \bigwedge_i \bigvee_j F_{i,j}$ *is true in* $U$ *iff* $U \in \mathscr{I}$, *where* $F_{i,j}$ *is either of the form* $\overline{\bigvee S_{i,j}}$ *or* $\bigvee S_{i,j}$, *then the signed formula* $\psi = \bigwedge_i \bigvee_j \psi_{i,j}$ *is a CNF* *for* $\lambda$ *and* $S$, *where* $\psi_{i,j} = (\forall x)(W - S_{i,j})\, A(x)$ *iff* $F_{i,j} = \overline{\bigvee S_{i,j}}$ *and* $\psi_{i,j} = (\exists x)\, S_{i,j} A(x)$ *iff* $F_{i,j} = \bigvee S_{i,j}$.

*Proof.* Let $I$ be a first-order interpretation and let $U = \mathrm{distr}_{I,x}(A(x))$. We have: $\mathrm{sval}_I(S(\lambda x)\, A(x)) = \mathrm{true}$ iff $\mathrm{val}_I((\lambda x)\, A(x)) \in S$, iff $\tilde{\lambda}(U) \in S$, iff $U \in \mathscr{I}$, iff $\mathrm{pval}_U(F) = \mathrm{true}$, iff $\mathrm{sval}_I(\psi) = \mathrm{true}$. In the last step we used Lemma 14 and the fact that the correspondence between propositional and signed formulas remains intact when forming disjunctions and conjunctions. Consequently, $\mathrm{sval}_I(S(\lambda x)\, A(x)) = \mathrm{sval}_I(\psi)$ for all $I$, i.e., $S(\lambda x)\, A(x) \equiv \psi$ is valid. ▮

$\psi$ inherits from $F$ all properties concerning minimality, with one minor exception: if a negative extended literal contains $m$ variables, then the sign of the corresponding universal formula contains $|W| - m$ truth values. In particular we obtain as a corollary of Lemma's 9 and 10 that the tight upper bound for the number of conjuncts is $2^{|W|-1}$. This bound has already been obtained in [1]; our proof, however, is better structured as it separates the propositional content from the many-valued one.

THEOREM 16. *Let* $\lambda$ *be a quantifier based on a semilattice* $\langle W, \leqslant \rangle$. *Furthermore, let the sign* $S$ *be an interval. Then*

$$\psi_S = (\forall x)\{u \mid u \geqslant S\}\, A(x) \wedge \bigwedge_{s \in \mathrm{succ}(S)} (\exists x)\{u \mid u \ngeqslant s, u \geqslant S\}\, A(x)$$

*is a CNF and a DNF for* $\lambda$ *and* $S$. $\psi_S$ *is minimal among all CNFs regarding the number of clauses and the total number of truth values in the signs.*

*Proof.* Obtained from Corollary 12 using Theorem 15. ▮

The number of truth values can be maximized by dropping the condition $u \geqslant S$ in the signs of the existential formulas. This can be justified either by using Corollary 13 or by applying Lemma 2(f). Formula $\psi_S$ in this latter form (i.e., with maximal signs) for singleton signs $S$ was already used in [1, 18] to obtain an upper bound on the number of conjuncts for semilattice-based quantifiers. The proof for the optimality of $\psi_S$ as well as the extension to intervals is new, however.

COROLLARY 17. *Let* $S_1, \ldots, S_n$ *be intervals. Then* $\psi_{S_1} \vee \cdots \vee \psi_{S_n}$ *is a DNF for* $\lambda$ *and* $S_1 \cup \cdots \cup S_n$.

Note that each singleton sign, i.e., each sign containing only one truth value, is already an interval. Therefore any sign can be decomposed into intervals. This way we obtain DNFs for arbitrary signs, which consist of at most as many disjuncts as there are truth values in the signs.

THEOREM 18. *Let* $\theta$ *be an operator based on a semilattice* $\langle W, \leqslant \rangle$. *Furthermore, let the sign* $S$ *be an interval. Then*

$$\psi = \bigwedge_{i=1}^{\mathrm{ar}(\theta)} \{u \mid u \geqslant S\}\, A_i \wedge \bigwedge_{s \in \mathrm{succ}(S)} \bigvee_{i=1}^{\mathrm{ar}(\theta)} \{u \mid u \ngeqslant s, u \geqslant S\}\, A_i$$

*is a minimal CNF for* $\theta$ *and* $S$.

*Proof.* Obtained from Corollary 12 using Lemma 14. ▮

EXAMPLE 8. Consider the $\wedge$-operator and the U-quantifier in Belnap logic (see Example 1). Theorems 16 and 18 yield the following sequent rules for U with sign$\{f\}$ and $\wedge$ with sign$\{u\}$:

$$\frac{\Gamma, A(\alpha)^f, A(\alpha)^u \quad \Gamma, A(\tau)^f}{\Gamma, ((\mathsf{U}x)\, A(x))^f} \qquad \frac{\Gamma, A_1^u, A_1^t \quad \Gamma, A_2^u, A_2^t \quad \Gamma, A_1^u, A_2^u}{\Gamma, (A_1 \wedge A_2)^u}.$$

For the signs $\{f, u\}$ and $\{f, t\}$ Theorem 16 and Corollary 17 yield the following tableaux for U (observe that $\mathrm{succ}_{\leqslant k}(\{f, u\}) = \{t\}$):

$$\begin{array}{c} \{f, u\}(\mathsf{U}x)\, A(x) \\ \hline \{f, u, \bot, t\}\ A(t) \\ \{f, u\}\ A(c) \end{array} \qquad \begin{array}{c|c} \multicolumn{2}{c}{\{f, t\}(\mathsf{U}x)\, A(x)} \\ \hline \{f, u\}\ A(t_1) & \{t, u\}\ A(t_2) \\ \{f\}\ A(c_1) & \{t\}\ A(c_2) \end{array}$$

The signed formula $\{f, u, \bot, t\}\ A(t)$ in the first tableau is a tautology and may be removed. Note that in all rules and tableaux the symbols $A$, $A_1$, and $A_2$ no longer denote predicate symbols but are place holders for arbitrary formulas.

## 8. CONCLUSION

In this paper we investigated methods for computing optimized rules for distribution quantifiers in many-valued logics. It turned out that the optimization of quantifiers can be viewed as the problem of minimizing certain types of propositional formulas. We outlined a procedure yielding optimized CNFs for quantifiers, resulting in improved rules for the sequent calculus, for natural deduction, and for clause formation. Generalizing an idea from [1] we obtained a schema for quantifiers based on semilattices and showed that it is minimal regarding the number of conjuncts and the number of literals. This schema not only applies to the above-mentioned calculi but also to tableaux with sets-as-sings, giving single extension rules for all signs forming an interval. As a by-product, a modified version of the schema also gives optimal rules for operators based on semilattices. Most of the obtained results have been implemented in the MUltlog system [2].

Distribution quantifiers have been around already for about 40 years [10], and the problem of their axiomatization has been solved in a general way for all usual calculi. The general method produces bulky rules with a high branching factor even when small ones exist. For the purpose of automatizing many-valued logics, however, slim rules are a must: the branching factor directly corresponds to the proof length. Nevertheless, only little work has been done in this direction up to now.

Zabel [17] gave simplified rules for singleton signs in the case where the distributions form a sublattice of the Boolean set lattice. Recently, Hähnle [8] generalized Zabel's result to distributive lattices allowing also up- and down-sets as signs. His schema produces single extension rules where all signs are again up-sets, down-sets, or singletons. This also extends his work on quantifier rules for regular logics [6], where the set of truth values is totally ordered.

In a sense, our work subsumes Hähnle's: Theorem 16 requires only the structure of a semilattice and can be used with any sign forming an interval; singletons, up-sets, and down-sets are just special cases of intervals. On the other hand Hähnle's work complements ours as it shows that distributive lattice are economical: all signs occurring in $\psi_S$ are just up- and down-sets, i.e., for distributive lattices only $2\,|W|$ signs out of $2^{|W|}$ are really needed. This does not hold for semilattices in general.

This paper and Hähnle's work shed new light on theorem proving with sets-as-signs. Up to now it seemed one of the foremost goals to keep the number of different signs at a minimum as each new sign required the computation and storage of new rules for all operators and quantifiers. Lattice-based quantifiers, however, need not be computed and stored in advance. They can be generated on the fly during proof search by mere instantiation for whatever sign needed, with the guarantee that the number of extensions is small. Instead of using a minimal number of signs one can now concentrate on minimizing the signs themselves in the hope of reducing the proof length. Of course this new approach is highly dependent on the operators present in the logic. If their rules are expensive to compute one has to care for the number of different signs. What is needed for the future is the investigation of classes of logics (like regular logics) as opposed to classes of quantifiers. The latter can only be regarded as a piece of the puzzle.

## ACKNOWLEDGMENTS

I thank Matthias Baaz and Richard Zach for enlightening discussions on this topic. Furthermore I am indebted to Chris Fermüller, Reiner Hähnle, and three anonymous referees for carefully reading the paper and suggesting several improvements.

Received July 27, 1998; published online August 4, 2000

## REFERENCES

1. Baaz, M., and Fermüller, C. G. (1995), Resolution-based theorem proving for many-valued logics, *J. Symbolic Comput*. **19**, 353–391.
2. Baaz, M., Fermüller, C. G., Salzer, G., and Zach, R. (1996), MUltlog 1.0: Towards an expert system for many-valued logics, *in* "13th Int. Conf. on Automated Deduction (CADE'96)," Lecture Notes in Artificial Intelligence, Vol. 1104, pp. 226–230, Springer-Verlag, Berlin/New York.
3. Baaz, M., Fermüller, C. G., and Zach, R. (1993), Systematic construction of natural deduction systems for many-valued logics, *in* "Proc. 23rd International Symposium on Multiple-valued Logic, May 24–27 1993," pp. 208–215, IEEE Comput. Soc. Press, Los Alamitos, CA.
4. Carnielli, W. A. (1987), Systematization of finite many-valued logics through the method of tableaux, *J. Symbolic Logic* **52**, 473–493.
5. Hachtel, G. D., and Somenzi, F. (1996), "Logic Synthesis and Verification Algorithms," Kluwer Academic, Dordrecht/Norwell, MA.
6. Hähnle, R. (1993), "Automated Deduction in Multiple-valued Logics," Clarendon Press, Oxford.
7. Hähnle, R. (1994), Short conjunctive normal forms in finitely-valued logics, *J. Logic Comput*. **4**, 905–927.

8. Hähnle, R. (1996), Commodious axiomatization of quantifiers in multiple-valued logic, *in* "Proc. 26th Int. Symp. on Multiple-Valued Logics, Santiago de Compostela, Spain, May 1996," IEEE Press, Los Alamitos.
9. McCluskey, E. J. (1956), Minimization of boolean functions, *Bell System Tech. J.* **35**, 1417–1444.
10. Mostowski, A. (1957), On a generalization of quantifiers, *Fund. Math.* **44**, 12–36.
11. Murray, N. V., and Rosenthal, E. (1993), Signed formulas: A liftable meta logic for multiple-valued logics, *in* "Proceedings ISMIS'93, Trondheim, Norway," Lecture Notes in Computer Science, Vol. 689, pp. 275–284, Springer-Verlag, Berlin/New York.
12. Nutz, K. F. (1984), "Digitaltechnik, BASIC: Berechnung und Optimierung von Digitalschaltungen mit Hilfe von BASIC-Programmen," Oldenbourg, Munich.
13. Quine, W. V. (1952), The problem of simplifying truth functions, *Amer. Math. Monthly* **59**, 521–531.
14. Rosser, J. B., and Turquette, A. R. (1952), "Many-Valued Logics," North-Holland, Amsterdam.
15. Rousseau, G. (1967), Sequents in many valued logic I, *Fund. Math.* **60**, 23–33.
16. Sasao, T., ed. (1993), "Logic Synthesis and Optimization," Kluwer Academic, Dordrecht/Norwell, MA.
17. Zabel, N. (1993), "Nouvelles Techniques de Déduction Automatique en Logiques Polyvalentes Finies et Infinies du Premier Ordre," Ph.D. thesis, Institut National Polytechnique de Grenoble.
18. Zach, R. (1993), "Proof Theory of Finite-valued Logics," Diplomarbeit, Technische Universität Wien.