J. Symbolic Computation (1992) 13, 235-254

A Resolution Framework for Finitely-Valued First-Order Logics

PETER W. O'HEARN[†] and ZBIGNIEW STACHNIAK[‡]

[†]*School of Computer and Information Science*
Syracuse University, Syracuse, NY 13244, USA

[‡]*Department of Computer Science*
York University, North York, Ontario M3J 1P3, Canada

(*Received 3 January, 1990*)

In this paper we propose a resolution proof framework on the basis of which automated proof systems for finitely-valued first-order logics (*FFO* logics) can be introduced and studied. We define the notion of a first-order resolution proof system and we show that for every disjunctive *FFO* logic a refutationally complete resolution proof system can be constructed. Moreover, we discuss two theorem proving strategies, the polarity and set of support strategies, and we prove their completeness.

1. Introduction

The diversity of logical systems and their applications in computer science has given rise to the search for general proof theories on the basis of which logic-independent automated theorem proving techniques can be developed. Several resolution based theorem proving techniques for classes of logical systems have been proposed. For example, Fariñas del Cerro & Herzig (1988), Ohlbach (1988), Chan (1987), and Abadi & Manna (1986) present resolution proof systems for modal logics, Orlowska (1985) and Morgan (1976) define the resolution principle in the context of many-valued logics, and Stachniak & O'Hearn (1990) introduce a resolution framework for the class of strongly finite logics. In this paper we propose a theoretical foundation for the introduction, analysis, and implementation of resolution proof systems for finitely-valued first-order logics (*FFO* logics).

The novelty of our proposal is the unification of the algebraic theories of logical systems and of non-clausal resolution (Stachniak, 1991, 1988; Stachniak & O'Hearn, 1990). As a result rich algebraic techniques can be used to express and investigate metatheoretical properties of resolution proof systems for *FFO* logics, such as the existence of effective methods of constructing resolution counterparts of *FFO* logics or the applicability of theorem proving strategies for directing and restricting the application of inference rules. *FFO* logics, traditionally studied by mathematical and philosophical logicians, have made their way to computer science through numerous applications in artificial intelligence and logic programming (Przymusinski, 1989; Fitting, 1988; Mycroft, 1988;

0747-7171/92/030235+20 $03.00/0

Ginsberg, 1986; Schmitt, 1986; Belnap, 1977), switching theory, logic design and testing of computer circuits (see Current, 1991 or Silio, 1990 for an overview of applications of FFO logics in these and other areas of computer science).

The key notion of a resolution proof system for a first-order logic is introduced in Section 3. This notion generalizes the concept of a non-clausal resolution proof system for a propositional logic introduced in Stachniak & O'Hearn (1990) and the (basic) resolution proof system for classical predicate logic proposed in Manna & Waldinger (1986). In Section 4 we show that every FFO logic can be formalized as a resolution proof system. Moreover, we show that resolution counterparts of FFO logics can be effectively constructed from the semantic descriptions of these logics. This fact opens a door for an automated generation of theorem proving environments for FFO logics. In Section 5 we discuss two speed-up techniques, the set-of-support strategy and the polarity strategy. We show that while restricting the application of the resolution rule both strategies retain the refutational completeness of resolution proof systems.

2. Logical Preliminaries

In this section we review briefly the syntax and semantics of propositional and first-order logics. The approach to formalizing these logics follows Wójcicki (1988), Hawranek & Tokarz (1977), Waszkiewicz & Weglorz (1968), and Łoś & Suszko (1958).

A *logic* is a pair $\mathcal{P} = < L, C >$, where L is an infinite recursive set (of *formulas*) and $C : 2^L \rightarrow 2^L$ is a function which satisfies the following two conditions: for every $X, Y \subseteq L$,

(c1) $X \subseteq C(X) = C(C(X))$,

(c2) $X \subseteq Y$ implies $C(X) \subseteq C(Y)$.

We call L and C the *language* and the *consequence operation* of $\mathcal{P}$, respectively. We view $C(X)$ as the set of all consequences of the set X of formulas and call X *inconsistent* (*consistent*) if $C(X) = L$ (if $C(X) \neq L$). Hence, C is the consequence operator in Tarski's sense. If $X \cup \{\alpha\} \subseteq L$, then we shall frequently write '$C(X, \alpha)$' instead of '$C(X \cup \{\alpha\})$'. The way a grammar for L is set (i.e. the way well-formed formulas of L are constructed), as well as a particular way the consequence operation is defined, is of limited interest to us at this place. Most (if not all) monotonic logical calculi studied in computer science can be represented as a pair $< L, C >$, and conversely, as observed by Łoś and Suszko, for every logic of the form $< L, C >$ there exists an axiom system defining it. Hence, the consequence operation framework provides us with an uniform way of dealing with both propositional and first-order logical calculi.

PROPOSITIONAL LOGICS: A language L is said to be propositional if its formulas are constructed in the usual manner by means of a countably infinite set $\{p_0, p_1, p_2, ...\}$ of propositional variables and a finite non-empty set $\{f_0, ..., f_k\}$ of logical connectives. If $\alpha \in L$, then we shall write $\alpha(p_0, ..., p_n)$ to indicate that the variables $p_0, ..., p_k$ occur in α, and $\alpha(p_0/\alpha_0, ..., p_n/\alpha_n)$ to denote the result of simultaneous replacement of every occurrence of every variable p_i of α by the corresponding formula α_i. We can view the connectives $f_0, ..., f_k$ as operations on L and hence $\mathcal{L} = < L, f_0, ..., f_k >$ forms an absolutely free algebra (frequently called an *algebra of formulas*) freely generated by the propositional variables. This observation implies that every function mapping the variables of L into an algebra similar to $\mathcal{L}$ can be extended to a homomorphism. In particular, every function e that maps variables of L into L, extends to the endomorphism e' of $\mathcal{L}$ such that for every formula $\alpha(p_0,...,p_n) \in L, e'(\alpha) = \alpha(p_0/e(p_0),...,p_k/e(p_n))$ (we assume here that $p_0,...,p_n$ are all the variables occurring in α). From the logical point of view e' can be thought of as a substitution function. For this reason we call endomorphisms of $\mathcal{L}$ *substitutions*. We shall often refer to the algebra of formulas $\mathcal{L}$ as a propositional language.

A logic $\mathcal{P} =< \mathcal{L}, C >$ is said to be *propositional* if $\mathcal{L}$ is a propositional language. We say that $\mathcal{P}$ is:

structural if for every $X \subseteq L$ and every substitution $e, \alpha \in C(X)$ implies $e(\alpha) \in C(e(X))$;

disjunctive if there is a binary connective $\vee$ (written as $\alpha \vee \beta$) such that for every $X \cup \{\alpha, \beta\} \subseteq L$, $C(X, \alpha \vee \beta) = C(X, \alpha) \cap C(X, \beta)$.

A *logical matrix* for $\mathcal{L}$ is a pair $\mathcal{M} =< \mathcal{A}, D >$, where $\mathcal{A}$ is an algebra similar to $\mathcal{L}$ and D is a subset of the base set $|\mathcal{A}|$ of $\mathcal{A}$. The elements of $|\mathcal{A}|$ are called *logical values* while the elements of D are called the *designated logical values*. Since $\mathcal{L}$ is an absolutely free algebra, every function mapping the set of variables of $\mathcal{L}$ into the set of truth-values of $\mathcal{M}$ can be extended to a function (to be precise, to a homomorphism of $\mathcal{L}$ into $\mathcal{A}$) that assigns truth-values to all the formulas of $\mathcal{L}$. For this reason the homomorphisms of $\mathcal{L}$ into $\mathcal{A}$ are called *valuations* and we denote the set of all valuations by $Hom(\mathcal{L}, \mathcal{M})$. With every matrix $\mathcal{M} =< \mathcal{A}, D >$ we associate the consequence operation $C_{\mathcal{M}}$ on $\mathcal{L}$ defined in the following way: for every $X \cup \{\alpha\} \subseteq L$,

$\alpha \in C_{\mathcal{M}}(X)$ *iff* for every $h \in Hom(\mathcal{L}, \mathcal{M})$, $h(\alpha) \in D$ if $h(X) \subseteq D$.

If $\mathcal{K}$ is a set of matrices for $\mathcal{L}$, then we let $C_{\mathcal{K}}$ denote the consequence operation defined by:

$C_{\mathcal{K}}(X) = \bigcap\{C_{\mathcal{M}}(X) : \mathcal{M} \in \mathcal{K}\}$, all $X \subseteq L$.

If $< \mathcal{L}, C >$ is a structural logic, then there exists a class $\mathcal{K}$ of matrices such that $C = C_{\mathcal{K}}$. Conversely, for every class $\mathcal{K}$ of matrices for $\mathcal{L}, < \mathcal{L}, C_{\mathcal{K}} >$ forms a structural logic. The logics defined by finite sets of finite matrices (i.e. matrices having a finite number of truth-values) are of special interest to us. These calculi, called *strongly finite logics*, in addition to being structural are compact in the following sense. A logic $< \mathcal{L}, C >$ is said to be *compact* if for every $X \subseteq L$:

$\alpha \in C(X)$ *iff* for some finite $X_f \subseteq X, \alpha \in C(X_f)$.

A collection $\mathcal{K}$ of matrices with the same underlying algebra can be conveniently represented as a single generalized matrix. A *generalized matrix* is a pair $\mathcal{G} =< \mathcal{A}, \mathcal{D} >$ where $\mathcal{A}$ is an algebra and $\mathcal{D}$ is a family of subsets of $|\mathcal{A}|$. With $\mathcal{G}$ we associate the consequence operation $C_{\mathcal{G}} = C_{\mathcal{K}}$, where $\mathcal{K} = \{< \mathcal{A}, D >: D \in \mathcal{D}\}$. It can be easily shown that for every finite set $\mathcal{F}$ of logical matrices there exists a generalized matrix $\mathcal{G}$ such that $C_{\mathcal{F}} = C_{\mathcal{G}}$. This means that strongly finite logics can be semantically defined by single generalized matrices. For the reader unfamiliar with the wider territory of matrix semantics for propositional logics we recommend Wójcicki (1988).

FFO LOGICS: A first order-language $\mathcal{L}$ is specified by the sets S_v of individual variables, S_c of logical connectives, and the sets S_p and S_f of predicate and function symbols, respectively. Let i be a function which assigns a natural number $i(f)$, the arity of f, to every symbol in $S_c \cup S_p \cup S_f$. We assume the usual recursive definition of the sets TR of terms and L of formulas of $\mathcal{L}$, and the standard definitions of an atomic formula, ground formula, and subformula of a formula. With every term $y \in TR$ we associate its degree, $d(y)$, in the following way. If y is an individual variable or a constant, then $d(y) = 0$. If $y = f(y_0, ..., y_k)$, then $d(y) = 1 + max(\{d(y_i) : i \leq k\})$. Let us note that $\mathcal{L}$ is quantifier-free. We assume that every formula of $\mathcal{L}$ is implicitly universally quantified. (This assumption will be reflected in the definition of a model of a formula of a first-order language.) This presentation of first-order languages reflects our desire to keep the theory of resolution proof systems separate from the part of logical theory that deals with the problem of quantifier elimination, the problem we shall briefly discuss at the end of Section 3.

Let $\mathcal{L}$ be an arbitrary first-order language and let $\mathcal{M} =< \mathcal{A}, D >$ be a finite matrix, where $\mathcal{A}$ is similar to $< \mathcal{L}, S_c >$. The operations of $\mathcal{A}$ provide the interpretations of the connectives in S_c. The semantics of $\mathcal{L}$ is defined by means of the notion of an interpretation of terms and formulas of $\mathcal{L}$ in a non-empty domain U and the matrix $\mathcal{M}$. An interpretation is a mapping π which assigns to every function symbol $f \in S_f$ a mapping $\pi(f) : U^{i(f)} \rightarrow U$, and to every predicate symbol $P \in S_p$, a mapping $\pi(P) : U^{i(P)} \rightarrow |\mathcal{A}|$. We call the pair $< U, \pi >$ an $\mathcal{M}$-frame. Any mapping $v : S_v \rightarrow U$ is called a *valuation* of individual variables. We extend v to a valuation v^* of $TR \cup L$ into an $\mathcal{M}$-frame $< U, \pi >$ in the usual way. An $\mathcal{M}$-frame $< U, \pi >$ is called an *$\mathcal{M}$-valued model* of a formula α (or simply a model of α) if for every valuation v into $< U, \pi >, v^*(\alpha) \in D$, i.e. α is 'true' in $\mathcal{M}$ under all valuations. We say that $< U, \pi >$ is a model of a set $X \subseteq L$, if $< U, \pi >$ is a model of every formula in X. Finally, we define a *first-order finitely-valued logic* (FFO logic) to be a pair $< \mathcal{L}, C >$, where $\mathcal{L}$ is a first-order language and C is a consequence operation such that for some finite generalized matrix $\mathcal{G} =< \mathcal{A}, \mathcal{D} >$ ($\mathcal{A}$ being similar to $< \mathcal{L}, S_c >$) and every $X \cup \{\alpha\} \subseteq L$,

$$\alpha \in C(X) \text{ iff for every } D \in \mathcal{D}, \text{ every } < \mathcal{A}, D >\text{-valued model of } X \text{ is a model of } \alpha.$$

Similar formalizations of FFO logics can be found in Hawranek & Tokarz (1977), Rescher (1969), Waszkiewicz & Weglorz (1968), and Wójcicki (1988). The class of FFO logics includes systems such as classical first-order predicate logic (defined by the matrix $< \mathcal{B}_2, \{1\} >$, where $\mathcal{B}_2$ is the two-element Boolean algebra and 1 is the greatest element of $\mathcal{B}_2$), modal logics defined by finite modal algebras, the families of Lukasiewicz, Gödel, and Post finitely-valued logics. Further examples can be found in Rescher (1969).

Throughout this paper we assume that all logics under consideration are disjunctive (with the disjunction connective denoted by $\vee$) and not pathological, i.e. they have at least one nonempty consistent and at least one finite inconsistent set.

3. Resolution Proof Systems

In this section we introduce the key definition of a resolution counterpart of a *FFO* logic. This notion generalizes the concept of a resolution proof system for a propositional logic introduced in Stachniak & O'Hearn (1990) and the (basic) resolution proof system for classical predicate logic proposed in Manna & Waldinger (1986) (see also Stachniak (1991) and Murray (1982)).

3.1 Propositional Resolution Proof Systems

The non-clausal resolution rule expressed in terms of classical connectives has the following form:

$$R_{\{F,T\}} : \quad \frac{\alpha_0(p), \alpha_1(p)}{\alpha_0(p/F) \vee \alpha_1(p/T)},$$

where T and F are formulas defining logical *truth* and *falsehood*, respectively (cf. Manna & Waldinger (1986), and Murray (1982)). To show that a formula α is a consequence of a finite set X we use the resolution rule to deduce F from $X \cup \{\neg\alpha\}$. For instance, to show that $p \vee \neg p$ is a theorem (i.e. is a consequence of $\emptyset$) we apply $R_{\{F,T\}}$ to $\alpha_0(p) = \alpha_1(p) = \neg(p \vee \neg p)$ and obtain $\neg(F \vee \neg F) \vee \neg(T \vee \neg T)$. This formula can be identified with F by the successive application of transformation rules, such as $T \vee \alpha \Rightarrow T$ or $\neg F \Rightarrow T$.

In Stachniak (1991) this non-clausal interpretation of the resolution principle is generalized to non-classical logics in the following way. Let $\mathcal{L}$ be a propositional language. A *resolution algebra* on $\mathcal{L}$ is a pair of the form $Rs = < \mathcal{A}, \mathcal{F} >$, where $\mathcal{A}$ is a finite algebra of formulas of $\mathcal{L}$, $\mathcal{A}$ and $\mathcal{L}$ are similar algebras, and $\mathcal{F}$ is a family of subsets of $|\mathcal{A}|$ with the containment property, i.e. if $X \in \mathcal{F}$ and $X \subseteq Y$, then $Y \in \mathcal{F}$. (Henceforth, we shall identify the connectives of $\mathcal{L}$ with the respective operations of $\mathcal{A}$.) With every resolution algebra on $\mathcal{L}$ we associate a *resolution proof system*, i.e. a deductive system based on the following inference rules: the resolution rule, the transformation rules, and the $\Box$-rules. These rules are defined as follows.

THE RESOLUTION RULE: Let Ver denote the base set of $\mathcal{A}$. The *resolution rule* is the set of all sequents of the form:

$$R_{Ver} : \quad \frac{\alpha_0(p), ..., \alpha_n(p)}{\alpha_0(p/v_0) \vee ... \vee \alpha_n(p/v_n)},$$

$\alpha_0(p), ..., \alpha_n(p)$ are arbitrary formulas of $\mathcal{L}$, p does not occur in formulas of Ver, and $v_0, ..., v_n$ is a fixed enumeration of Ver. Intuitively, if a set $X = \{\alpha_0(p), ..., \alpha_n(p)\}$ of formulas is consistent, then the sentence $\alpha_0(p/v_0) \vee ... \vee \alpha_n(p/v_n)$ is satisfiable. The elements of Ver are called *verifiers* and their role is similar to T and F in the classical case; they 'witness' the consistency of X.

THE TRANSFORMATION RULES: A *transformation rule* is an expression of the form $f(v_1, ..., v_t) \Rightarrow v$, where f is a t-ary connective, $v_1, ..., v_t, v \in Ver$, and $f(v_1, ..., v_t) = v$ holds in $\mathcal{A}$. The role of transformation rules is to simplify formulas by replacing occurrences of $f(v_1, ..., v_t)$ with inferentially equivalent verifier v.

THE $\Box$-RULES: A *$\Box$-rule* is an expression of the form $Y \Rightarrow \Box$, where $Y \in \mathcal{F}$ and $\Box$ is a designated constant symbol (not in L) denoting falsehood. The $\Box$-rules provide a mechanism for terminating a deductive process.

From now on, we shall identify a resolution algebra with the resolution proof system it defines. We shall write $X \overset{Rs}{\Longrightarrow} \Box$ if there is a sequence $\alpha_0, ..., \alpha_m$ of formulas (a *refutation of* X) such that $\alpha_m = \Box$ and each α_i either is in X or is obtained from some formulas in the sequence earlier than α_i with the help of an inference rule of Rs.

Let $\mathcal{P} =< \mathcal{L}, C >$ be a propositional logic. A resolution proof system Rs on $\mathcal{L}$ is said to be a *resolution counterpart* of $\mathcal{P}$ if the following three conditions are satisfied (cf. Stachniak (1988, 1991)):

(r1) for every finite $X \subseteq L, C(X) = L$ *iff* $X \overset{Rs}{\Rightarrow} \Box$, provided that formulas of X and Ver do not share variables;

(r2) if $w_0 \Rightarrow w_1$ is a transformation rule, then for every $\alpha(p) \in L, C(\alpha(p/w_0)) = C(\alpha(p/w_1))$;

(r3) for every $V \subseteq Ver, C(V) = L$ *iff* $V \in \mathcal{F}$.

The condition (r1), called *refutational completeness,* expresses the direct correspondence between finite inconsistent and refutable sets. We shall call every set X satisfying $Var(X) \cap Var(Ver) = \emptyset$ *clean* in Rs. The condition (r2) says that transformation rules preserve inferential equivalence, i.e., if β is obtained from a formula α by the application of a transformation rule, then β and α are inferentially equivalent. The meaning of (r3) is obvious.

Stachniak & O'Hearn (1990) and Stachniak (1988,1991) are devoted entirely to the study of propositional resolution proof systems and the reader may refer to these papers for further discussion on this topic. What we need for the introduction of resolution proof systems for FFO logics are the following two facts concerning strongly finite logics.

THEOREM 3.1 (Stachniak & O'Hearn, 1990): *Resolution counterparts of strongly finite disjunctive logics can be effectively constructed.*

With every resolution proof system $Rs =< \mathcal{A}, \mathcal{F} >$ on $\mathcal{L}$ we can associate the generalized matrix $\mathcal{M}_{Rs} =< \mathcal{A}, \mathcal{D} >$, the *matrix induced* by Rs, defined in the following way. We call a set $D \subseteq |\mathcal{A}|$ *maximal consistent* in Rs if $D \notin \mathcal{F}$ and for every $D' \subseteq |\mathcal{A}|$, if $D \subseteq D'$ and $D \neq D'$, then $D' \in \mathcal{F}$. The family $\mathcal{D}$ consists of all maximal consistent sets in Rs.

THEOREM 3.2 (Stachniak, 1991): *Let Rs be a resolution counterpart of a structural disjunctive logic $< \mathcal{L}, C >$ and let $\mathcal{M}_{Rs} =< \mathcal{A}, \mathcal{D} >$ be the matrix induced by Rs. Then:*

(i) *for every finite $X \subseteq L, C(X) = L$ iff $C_{\mathcal{M}_{Rs}}(X) = L$;*

(ii) *for every $D \in \mathcal{D}$ and every $h \in Hom(\mathcal{L}, \mathcal{A}), h(L) \not\subseteq D$.*

3.2 First-Order Resolution

Resolution proof systems for FFO logics can be introduced following the the formalization of propositional resolution proof systems. Before we go in this direction let us look at the non-clausal resolution proof system for quantifier-free first-order logic introduced in Manna & Waldinger (1986). This system is based on two types of inference rules:

- the transformation rules, i.e. expressions of the form $v_0 \Rightarrow v_1$, which allow to simplify formulas by replacing occurrences of v_0 with inferentially equivalent but syntactically simpler v_1;
- the resolution rule, which has the following general form:

$$R_{\{F,T\}}: \frac{\alpha(\phi), \beta(\psi)}{\theta(\alpha)(\theta(\phi)/F) \vee \theta(\beta)(\theta(\psi)/T)}.$$

In this rule, ϕ and ψ are subformulas of α and β, respectively; α and β do not share variables (are standardized apart), and θ is a most general unifier of ϕ and ψ (hence $\theta(\phi) = \theta(\psi)$). The conclusion of this rule is obtained by replacing all occurrences of $\theta(\phi)$ in $\theta(\alpha)$ by F, replacing all occurrences of $\theta(\psi)$ in $\theta(\beta)$ by T, and taking the disjunction of the results; T and F are formulas defining *truth* and *falsehood*, respectively.

This resolution proof system operates on a finite set of quantifier-free formulas called the ***deduced set***. A proof consists of a finite number of applications of the inference rules to formulas in the deduced set. Each such application adds new formulas to the deduced set without altering its consistency status. In a proof we attempt to show that a given deduced set is inconsistent by trying to add F to the set.

Having developed the notion of a propositional resolution proof system, we can model the notion of a resolution counterpart of a FFO logic on the example just discussed. Let $\mathcal{L}$ be a first-order language. A resolution proof system on $\mathcal{L}$ is a deductive system of the form $Rs = <\mathcal{A}, \mathcal{F}>$, where $\mathcal{A}$ is a finite algebra of ground formulas of $\mathcal{L}$, whose operations correspond to the connectives of $\mathcal{L}$, and $\mathcal{F}$ is a nonempty family of subsets of $|\mathcal{A}|$ with the containment property. The transformation and $\Box$-rules are determined by Rs similarly to the propositional case.

The generalized resolution rule has the following form:

$$R_{Ver}: \frac{\alpha_0(\phi_0), ..., \alpha_n(\phi_n)}{\theta(\alpha_0)(\theta(\phi_0)/v_0) \vee ... \vee \theta(\alpha_n)(\theta(\phi_n)/v_n)}.$$

In this rule, $\alpha_0, ..., \alpha_n$ are standardized apart, ϕ_i is an atomic subformula of α_i which does not share subformulas with verifiers, and θ is a most general unifier of $\phi_0, ..., \phi_n$ (i.e. $\theta(\phi_0) = ... = \theta(\phi_n)$). The conclusion of this rule is obtained by replacing all occurrences of $\theta(\phi_i)$ in $\theta(\alpha_i)$ with the ***i − th*** verifier v_i, and taking the disjunction of the results.

Let Rs be a resolution proof system on $\mathcal{L}$ and let $X \subseteq L$. As in the propositional case, we shall write $X \overset{Rs}{\Rightarrow} \Box$, to denote the existence of a refutation of X in Rs, i.e. the existence of a sequence $\alpha_0, ..., \alpha_m$ of formulas such that $\alpha_m = \Box$ and each α_i either is in X or is obtained from some formulas in the sequence earlier than α_i with the help of an inference rule of Rs. We shall call Rs a ***resolution counterpart*** of a FFO logic $<\mathcal{L}, C>$ if the conditions (r1)-(r3) are satisfied (in (r1) we assume that X is clean in Rs, i.e. the formulas of X and Ver do not share atomic subformulas; in (r2) the symbol 'p' represents a ground atomic formula). In the next section we show that this definition of a resolution counterpart is sufficient to provide a resolution formalization of every FFO logic.

From the refutational completeness condition (r1) it follows that resolution counterparts of FFO logics can be explicitly used only to determine the consistency status of the deduced set. The fundamental principle on which refutational theorem proving techniques are based is the existence of an effective (and preferably efficient) method for reducing the problem of entailment to inconsistency. In classical predicate logic this reduction proceeds by first forming the set $X \cup \{\neg\alpha\}$, and then eliminating quantifiers from formulas of this set *via* conversion into prenex normal form and skolemization. The resulting set of quantifier-free formulas is inconsistent if and only if α is a consequence of X. For other first-order logics the existence of an effective reducibility method based on quantifier elimination depends, among other things, on the type and logical properties of quantifiers as well as on the properties of connectives. For some calculi, such as the many-valued logics of Post (Orłowska 1985) or the three-valued logic discussed in Schmitt (1986), quantifier elimination is readily available, while for other logics (e.g. some modal logics) it is not.

There are two reasons why we have restricted our attention to quantifier-free fragments of first-order logics. First, as we intend to show in this paper, resolution counterparts of quantifier-free FFO logics can be introduced and studied within an uniform algebraic framework. Second, we want to separate the theory of resolution from purely logical question about reducibility which, to the best of our knowledge, has not been generally studied in the context of non-classical logics and is dealt only on a logic-by-logic case.

While reduction of entailment to inconsistency through quantifier elimination is one of the most commonly used techniques, there are other directions that might be successfully explored. In Abadi & Manna (1986) it is shown that for some modal logics the process of reducibility may require no, or only some, form of quantifier elimination, and that the resolution principle can be defined in terms of quantifier-free as well as quantified formulas.

4. Existence of Resolution Counterparts

In this section we show that every FFO logic has a resolution counterpart. We obtain this result by proving analogues of the Herbrand Theorem and the Lifting Lemma for classical predicate logic. Throughout this section we assume that all logics under consideration have countably infinite number of ground atomic formulas.

Let $\mathcal{P} = < \mathcal{L}, C >$ be an arbitrary but fixed disjunctive $\mathcal{G}$-valued FFO logic, where $\mathcal{G} = < \mathcal{A}, \mathcal{D} >$ is a generalized finite matrix, and for every $D \in \mathcal{D}, D \neq |\mathcal{A}|$. Let $\mathcal{L}_p$ be a propositional language with the same logical connectives as $\mathcal{P}$ and let L_p denote the set of formulas of $\mathcal{L}_p$. We call $\mathcal{P}_p = < \mathcal{L}_p, C_{\mathcal{G}} >$ the propositional logic associated with $\mathcal{P}$. Let t be a one-one function mapping the set of all atomic formulas of $\mathcal{P}$ onto the set of all propositional variables of $\mathcal{L}_p$.

If $X \subseteq L$ and $m \geq 0$, then by $H^m(X)$ we denote the set of all ground instances of formulas of X with terms of degree $> m$. For every $\alpha \in H^m(X)$, $t(\alpha)$ denotes the result of replacing every atomic formula A occurring in α with $t(A)$. Clearly, $t(H^m(X)) \subseteq L_p$.

THEOREM 4.1: (Herbrand Theorem) *Let* $m \geq 0$. *Then for every* $X \subseteq L$ *the following conditions are equivalent:*

(i) $C(X) = L$;
(ii) $C(Y) = L$, *for some finite subset* Y *of* $H^m(X)$;
(iii) $C(H^m(X)) = L$;
(iv) $C_{\mathcal{G}}(t(H^m(X))) = L_p$.

PROOF: (i)⇒(iv) Suppose that for some valuation $h \in Hom(\mathcal{L}_p, \mathcal{G})$, and some $D \in \mathcal{D}, h(t(H^m(X))) \subseteq D$. Let $< U, \pi >$ be the $< \mathcal{A}, D >$-frame consisting of the set U of all closed terms of L of degree $\geq n$, and of the mapping π defined in the following way: for every k-ary function symbol $f, \pi(f)(t_1, ..., t_k) = f(t_1, ..., t_k)$, and for every k-ary predicate $P, \pi(P)(t_1, ..., t_k) = h(t(P(t_1, ..., t_k)))$. Since for every valuation v into $< U, \pi >$ and for every $\alpha \in X$ we have that $v(\alpha) \in h(t(H^m(\{\alpha\}))) \subseteq D, C(X) \neq L$.

(iv)⇒(iii) Suppose that $\mathcal{U}$ is a model of $H^m(X)$.Let v be any valuation into $\mathcal{U}$, and let $h \in Hom(\mathcal{L}_p, \mathcal{G})$ be selected so that for every atomic formula A occurring in formulas of $H^m(X), h(t(A)) = v(A)$. Clearly, $h(t(H^m(X)))$ is a subset of a set of designated values of $\mathcal{G}$ and hence, $t(H^m(X))$ is satisfiable in $\mathcal{G}$.

(iii)⇒(ii) If (iii) is true, then $C_{\mathcal{G}}(t(H^m(X))) = L_p$ (otherwise $< U, \pi >$, defined as in the proof of (i)⇒(iv), would be a model of $H^m(X)$). Since the matrix $\mathcal{G}$ is finite, for all $Z \subseteq L_p$,

$$C_{\mathcal{G}}(Z) = \bigcup\{C_{\mathcal{G}}(Z_f) : Z_f \subseteq Z \text{ is finite}\}$$

(cf. Łoś & Suszko (1958)). Since $\mathcal{P}_p$ has a finite inconsistent set, for some $Y_f \subseteq t(H^m(X)), C_{\mathcal{G}}(Y_f) = L_p$. Let $Y = t^{-1}(Y_f)$. If $C(Y) \neq L$, then for some $< \mathcal{A}, D >$-frame $\mathcal{U}$ and for every valuation v into $\mathcal{U}, v(Y) \subseteq D$. Let h be a valuation of $\mathcal{L}_p$ into $M_{\mathcal{G}}$ defined as follows: for every variable $p, h(p) = v(t^{-1}(p))$. Clearly, $h(Y_f) \subseteq D$, which contradicts the fact that $C_{\mathcal{G}}(Y_f) = L_p$.

(ii)⇒(i): Suppose that $H^m(X)$ is unsatisfiable. Let $\mathcal{U}$ be a $< \mathcal{A}, D >$-frame where $D \in \mathcal{D}$, and let v be a valuation into $\mathcal{U}$. There is a formula $\alpha \in H^m(X)$, a formula $\beta \in X$, and a function e, mapping the set of individual variables of $\mathcal{L}$ into the set of closed terms, such that $\alpha = e(\beta)$ and $v(\alpha) \notin D$. Let v' be a valuation into $\mathcal{U}$ such that for every variable $x, v'(x) = v(e(x))$. Clearly, $v'(\beta) \notin D$, which means that X is inconsistent. □

LEMMA 4.2: (Lifting Lemma) *Suppose* $\alpha_0(t), ..., \alpha_n(t)$ *are ground instances of* $\beta_0, ..., \beta_n \in L$, *respectively, where* t *is an atomic formula. Then there is an Rs-resolvent* $\gamma_0 \vee ... \vee \gamma_n$ *of* $\beta_0, ..., \beta_n$ *such that* $\alpha_i(t/v_i)$ *is an instance of* $\gamma_i, 0 \leq i \leq n$.

PROOF: Let $\lambda_i, 0 \leq i \leq n$, be substitutions which standardize variables of $\beta_0, ..., \beta_n$ apart and let θ be a substitution such that $\theta(\lambda_i(\beta_i)) = \alpha_i$. Moreover, let W be the set of all atomic subformulas of $\lambda_i(\beta_i)$ such that $\theta(A) = t$ and let σ be a most general unifier of W. Since θ is also a unifier of W, $\theta = \sigma \circ e$, for some substitution e. Hence, if $z \in W$, then $\alpha_i(t/v_i)$ is an instance of $\sigma(\lambda_i(\beta_i))(\sigma(z)/v_i)$. □

Let Rs_p be a resolution counterpart of $\mathcal{P}_p$ and let Rs be obtained from Rs_p by replacing every variable p occurring in verifiers of Rs_p by $t^{-1}(p)$. We assume that for every variable p occurring in verifiers of $Rs_p, t^{-1}(p)$ is a ground atomic formula (otherwise the variable of Rs_p can be renamed to satisfy this requirement). The Herbrand Theorem and the Lifting Lemma are instrumental in proving the following result.

THEOREM 4.3: *Rs is a resolution counterpart of $\mathcal{P}$.*

PROOF: Let m denote the maximal degree among terms occurring in verifiers of Rs. Since the proof of (r3) is straightforward we concentrate on (r1) and (r2).

(r2) Let $w_0 \Rightarrow w_1$ be a transformation rule of Rs and let$w_0^p \Rightarrow w_1^p$ be the corresponding rule of Rs_p. Moreover, suppose that for some $\alpha(A) \in L$, $\alpha(A/w_1) \notin C(\alpha(A/w_0))$, i.e. there exists a set $D \in \mathcal{D}$ and a $< \mathcal{A}, D >$-frame $\mathcal{U}$ such that for every valuation $v, v(\alpha(A/w_0)) \in D$ while for some valuation $v_0, v_0(\alpha(A/w_1)) \notin D$. Let $t(A) = p$ and let $\beta(p) \in L_p$ be the formula obtained from $\alpha(A)$ by renaming every atomic subformula B of $\alpha(A)$ as $t(B)$. Finally, let $h \in Hom(\mathcal{L}_p, \mathcal{G})$ be such that $h(p) = v_0(t^{-1}(p))$. Then, $h(\beta(p/w_0^p)) = v_0(\alpha(A/w_0)) \in D$ but $h(\beta(p/w_1^p)) = v_0(\alpha(A/w_1)) \notin D$ which contradicts the assumption that $w_0^p \Rightarrow w_1^p$ satisfies (r2). Therefore, $\alpha(A/w_1) \in C(\alpha(A/w_0))$. The same argument can be used to show that $\alpha(A/w_0) \in C(\alpha(A/w_1))$, and hence that Rs satisfies (r2).

(r1) Suppose X is a finite inconsistent set of formulas clean in Rs. By the Herbrand Theorem there is a finite inconsistent subset Y of $H^m(X)$. Without any loss of generality we can assume that Y contains an instance of every formula of X and that $t(Y)$ is clean in Rs_p. Let

(T_p) $\alpha_0^p, ..., \alpha_k^p$

be a refutation of $t(Y)$ in Rs_p. We shall simulate T_p to construct a refutation

(T) $\alpha_0, ..., \alpha_k$

of X in Rs such that for every $0 \leq i < k$,

(a) $t^{-1}(\alpha_i^p) \in H^m(\{\alpha_i\})$.

If $\alpha_i^p \in t(Y)$, then let $\alpha_i \in X$ be any formula such that for some ground instance $\beta_i \in Y$ of $\alpha_i, \alpha_i^p = t(\beta_i)$. Suppose that α_i^p is obtained from α_j^p by the application of a reduction rule $f(v_0, ..., v_l) \Rightarrow v$. Since for every $i \leq l$, the degree of $t^{-1}(v_i)$ is $\leq m, f(t^{-1}(v_0), ..., t^{-1}(v_l))$ occurs in α_j. Let α_i be the result of the application of the reduction $f(t^{-1}(v_0), ..., t^{-1}(v_l)) \Rightarrow t^{-1}(v)$ to α_j. Let us note that (a) is preserved. If α_i^p is obtained from $\alpha_{i_0}^p, ..., \alpha_{i_n}^p$ using R_{Ver_p}, then let α_i be obtained from $\alpha_{i_0}, ..., \alpha_{i_n}$ using R_{Ver}. The application of the resolution rule is possible by the Lifting Lemma and (a). Finally, since $\alpha_k^p = \Box$, there are $i_0, ..., i_l < k$ such that $\{\alpha_{i_0}^p, ..., \alpha_{i_l}^p\} \Rightarrow \Box$ is a $\Box$-rule. By (a), $\alpha_{i_0}, ..., \alpha_{i_l} \in Ver$ and $\{\alpha_{i_0}, ..., \alpha_{i_l}\} \Rightarrow \Box$ is in $R_\Box$. To conclude, $\alpha_k = \Box$ and (T) is a refutation of X. This completes the first half of the proof of (r1).

For the second half, let $c_0, ..., c_n$ be new symbols which do not occur in formulas of $\mathcal{L}$ and let $\mathcal{L}(c_0, ..., c_n)$ be the extension of $\mathcal{L}$ obtained by the addition of $c_0, ..., c_n$ as 0-argument connectives (logical constants). Let $\mathcal{M}_{Rs} =< \mathcal{V}, \mathcal{D} >$ be the generalized matrix induced by Rs and let $\mathcal{M}^* =< \mathcal{V}^*, \mathcal{D}^* >$ be the matrix for $\mathcal{L}(c_0, ..., c_n)$ defined as follows. The algebra $\mathcal{V}^*$ is obtained from $\mathcal{V}$ be renaming every verifier v_i as c_i and by adding $c_0, ..., c_n$ as 0-argument operations (constants). The family $\mathcal{D}^*$ is obtained from $\mathcal{D}$ by renaming every verifier v_i in every set of $\mathcal{D}$ as c_i. Finally, let C^* denote the consequence operation on $\mathcal{L}(c_0, ..., c_n)$ defined by $\mathcal{M}^*$. An immediate consequence of the above definitions is the following equivalence:

(b) $C(Y) = L$ *iff* $C^*(Y) = L(c_0, ..., c_n)$, for every $Y \subseteq L$.

Next, let Rs^* be the resolution proof system on $\mathcal{L}(c_0, ..., c_n)$ obtained from Rs by renaming every verifier v_i as c_i. Clearly, for every set $Y \subseteq L$ clean in Rs:

(c) $Y \overset{Rs}{\Rightarrow} \Box$ *iff* $Y \overset{Rs^*}{\Rightarrow} \Box$.

We claim that in addition:

(d) $C^*(Y) \neq L(c_0, ..., c_n)$ *implies* $Y \overset{Rs^*}{\not\Rightarrow} \Box$.

To prove the claim, first let us note that every transformation rule of Rs^* is sound with respect to C^*, since every such rule corresponds to an equation true in the algebra V^*. To show the soundness of the $R_{\mathcal{V}_{er}}$ let us suppose that $d \in \mathcal{D}^*$ and that $\mathcal{U}$ is a $< \mathcal{V}^*, d >$-valued model of $Z = Y \cup \{\alpha_0(A), ..., \alpha_n(A)\}$. Then for every valuation $v, v(Z) \subseteq d$. Let v be an arbitrary valuation. We claim that $v(\alpha_0(A/c_0) \vee ... \vee \alpha_n(A/c_n)) \in d$. Since for some formula $\alpha_l(A), v_0(\alpha_l(A/c_l)) = v(\alpha_l(A)) \in d, v_0(\alpha_0(A/c_0) \vee ... \vee \alpha_n(A/c_n)) \in d$. Now, (d) follows from (r2) and (r3). The proof that $C(X) \neq L$ implies $X \overset{Rs}{\not\Rightarrow} \Box$ follows from (b), (d), and (c). $\Box$

From Theorem 4.3 we conclude that there is an effective method for the construction of resolution counterparts of disjunctive FFO logics. Namely, by Theorem 3.1 we can effectively construct a disjunctive resolution counterpart Rs_p of $\mathcal{P}_p$ which, by Theorem 4.3, can be transformed into a resolution counterpart of $\mathcal{P}$. The resolution framework presented in this paper can also be applied to non-FFO logics. The definition of a resolution counterpart requires no changes to work in the context of first-order and not just FFO logics. Clearly, the conditions (r1) - (r3) do not require a logic under consideration to be FFO.

5. Theorem Proving Strategies

It is an opinion expressed by many researchers that for a theorem proving system to be effective, it must employ strategies to direct and restrict the application of the inference rules. In Wos (1988) we read that 'Even for the simplest of problems, an uncontrolled use of some chosen inference rule will produce far too much information.' In this section we show that two of the more powerful strategies for automated proof systems available for classical logic, namely the set of support strategy (Wos, 1988; Wos *et. al.*, 1965) and the polarity strategy (Manna & Waldinger, 1986; Murray, 1982), can be generalized and applied to resolution counterparts of FFO logics preserving their refutational completeness.

5.1 Set of Support Strategy

The set of support strategy, one of the more powerful restriction strategies for theorem proving in classical logic, was proposed in Wos *et al.* (1965). It suggests that, when proving inconsistency of a set X of formulas containing a consistent subset T, we should avoid resolving formulas in T. First we show that the set of support strategy preserves completeness of resolution counterparts of propositional logics, and then lift this result to FFO logics.

Throughout this section we let $\mathcal{P}$ be a FFO logic, $\mathcal{P}_p$ be the corresponding propositional logic, and $Rs_p =< \mathcal{A}, \mathcal{F} >$ be a resolution counterpart of $\mathcal{P}_p$ with Ver_p as its set of verifiers.

A subset Y of a set X is called a *set of support* of X if $X - Y$ is consistent. An instance

$$\frac{\alpha_0(p),...,\alpha_n(p)}{\alpha_0(p/v_0)\vee...\vee\alpha_n(p/v_n)}$$

of the resolution rule of Rs_p is said to be (X, Y)-supported if Y is a set of support of X and $\alpha_i(p) \notin X - Y$, for some $0 \le i \le n$. We say that a refutation in Rs_p is (X, Y)-supported if every application of the resolution rule uses an (X, Y)-supported instance of this rule.

The following theorem shows that by restricting refutations to be (X, Y)-supported, the system Rs_p retains its refutational completeness with respect to $\mathcal{P}_p$. This theorem improves the result on set of support strategy reported in Stachniak & O'Hearn (1990).

THEOREM 5.1: *The set of support strategy is complete. That is, if Y is a set of support of a finite inconsistent set X clean in Rs_p, then there is an (X, Y)-supported refutation of X in Rs_p.*

PROOF: We adopt the notions of a weighted formula and of a semantic tree from Stachniak (1991) (see also O'Hearn & Stachniak (1989)), where the reader may refer for formal definitions and basic properties. Let X and Y be as required and let $\mathcal{M}_p =< \mathcal{A}, \mathcal{D} >$ be the matrix induced by Rs_p. Since $\mathcal{P}_p$ is disjunctive and the family $\mathcal{D}$ consists of maximal consistent sets, $\mathcal{M}_p$ is *well-connected*, i.e. for all $D \in \mathcal{D}$, and all $a_i, a_j \in Ver_p$,

$$a_i \vee a_j \in D \text{ iff } a_i \in D \text{ or } a_j \in D.$$

Since X is inconsistent in $\mathcal{P}$, by Theorem 3.2, $C_{\mathcal{M}_p}(X) = L$. Throughout the proof of this theorem we shall identify a weighted formula $< \alpha(q_1, ..., q_k), f >$ with the formula $\alpha(q_1/f(q_1), ..., q_k/f(q_k))$, when $\{q_1, ..., q_k\}$ constitutes the domain of f. We construct a sequence $X_0, ..., X_t$ of sets in which every set X_k is associated with a sequence $S_1^k, ..., S_m^k$ of trees such that:

(a) for every $1 \le j \le m$, S_j^k is a closed semantic tree of X_k with respect to $< \mathcal{A}, \{D_j\} >$.

We start the construction of $X_0, ..., X_t$ by letting $X_0 = X$. For each $1 \le j \le m$, let S_j^0 be any closed semantic tree of X with respect to $< \mathcal{A}, \{D_j\} >$. In Stachniak (1991) it is shown that such trees can always be selected since

(b) $C_{\mathcal{M}_p}(X) = \bigcap\{C_{<\mathcal{A},\{D\}>}(X) : D \in \mathcal{D}\}$.

So (a) holds for $k = 0$.

Suppose that the sequence has been constructed until $k \geq 0$ and let $i = k + 1$. Let $S_1^k, ..., S_m^k$ be the closed semantic trees associated with X_k. If all the trees $S_1^k, ..., S_m^k$ are one-element, then we stop the construction. Otherwise, there is a tree S_j^k and a node K of S_j^k such that:

(c) some child K_j of K falsifies a formula $\alpha_j(p_i) \notin X - Y$ in $< \mathcal{A}, \{D_j\} >$.

Let us note that if for every tree S_j^k (c) were false, then $X - Y$ would not be satisfiable in any of the matrices $< \mathcal{A}, \{D_j\} >$ and, by (b), $X - Y$ would be inconsistent in $\mathcal{P}_p$. Let $K_0, ..., K_n$ be the children of K and let $\alpha_0, ..., \alpha_n$ be formulas of X_k such that α_q is falsified by $K_q, 0 \leq q \leq n$ (note that α_j satisfies (c)). We apply the resolution rule to $\alpha_0(p_i), ..., \alpha_n(p_i)$ and form the set

$$X_i = X_k \cup \{\alpha_0(p_i/v_0) \vee ... \vee \alpha_n(p_i/v_n)\}.$$

We can then associate $\alpha_0(p_i/v_0) \vee ... \vee \alpha_n(p_i/v_n)$ with a weighted formula $< \beta(r_1, ..., r_m), g >$ so that

$$\beta(r_1/g(r_1)...r_m/g(r_m)) = \alpha_0(p_i/v_0) \vee ... \vee \alpha_n(p_i/v_n).$$

Let us note that since $\mathcal{M}_{Rs}$ is well-connected, $< \beta, g >$ is falsified by K. Using this fact we can form trees $S_1^i, ..., S_m^i$ which satisfy (a) and have fewer inference nodes than $S_1^k, ..., S_m^k$. Clearly, after a finite number of steps we will obtain a set X_t such that k has a sequence of one-element semantic trees associated with it. From here, using (r2) and (r3) it is not difficult to show that $\Box$ can be deduced from X_t. $\Box$

The notion of a set of support generalizes to FFO logics in the obvious way. Let Rs be the resolution counterpart of $\mathcal{P}$ constructed from Rs_p as in Section 4.

THEOREM 5.2: *Let $X \subseteq L$ be a finite inconsistent set of formulas clean in Rs. If Y is a set of support of X, then there is an (X, Y)-supported refutation of X in Rs.*

PROOF: Let X and Y be as required and let m be the maximal degree among terms occurring in verifiers of Rs. By the Herbrand Theorem, $H^m(X - Y)$ is consistent and hence $Z = H^m(X) - H^m(X - Y)$ is a set of support of $H^m(X)$. Let us note that

(1) if $\alpha \in H^m(\{\beta\})$ and $\alpha \notin H(X) - Z$,then $\beta \notin X - Y$.

Since $t(H^m(X))$ is $C_{\mathcal{M}}$-inconsistent, by Theorem 5.1, there is a refutation of $t(H^m(X))$ restricted by (s-of-s) with $t(Z)$ as the set of support. As in the proof of Theorem 4.3 we can use the Lifting Lemma to construct a refutation of X in Rs. By (1), this refutation preserves (s-of-s). $\Box$

As an immediate consequence of Theorem 5.2 we obtain the completeness of set of support for non-clausal resolution for classical predicate logic.

5.2 Polarity Strategy

The second speed-up technique discussed in this paper is the polarity strategy (cf. Stachniak & O'Hearn (1990), Manna & Waldinger (1986), and Murray (1982)). The central idea behind this technique is to use the polarity status of subexpressions of formulas to guide the application of the resolution rule. This strategy has been used in non-clausal resolution counterparts of classical predicate logic studied in Manna & Waldinger (1986) and Murray (1982).

Let $\mathcal{P} =< \mathcal{L}, C >$ be an arbitrary but fixed FFO logic and let $\mathcal{P}_p =< \mathcal{L}_p, C_p >$ be the propositional logic associated with $\mathcal{P}$ (cf. Section 4). Let $<$ be a binary relation on L satisfying the following condition:

(p) $\alpha < \beta$ implies $\beta \in C_p(\{\alpha\})$.

We adopt the notion of polarity of a subformula with respect to $<$ from Manna & Waldinger (1986), where the reader may refer for a formal definition and detailed discussion of this notion (see also Stachniak & O'Hearn (1990)). Intuitively, we are effectively assigning to some subformulas α of a formula β the polarity value '+' (positive) or '−' (negative). This assignment is defined so that, if α is positive, then replacing α with a larger expression (with respect to $<$) will result in a larger formula. Similarly, if α is negative, then its replacement with a smaller expression (with respect to $<$) will result in a larger formula. We define α to be of no polarity, if it is neither positive nor negative.

Let $<$ be an arbitrary but fixed relation satisfying (p). Suppose that Rs_p is a resolution counterpart of $\mathcal{P}_p$ and let us assume that there are two verifiers v_F, v_T of Rs such that for every verifier $v, v_F < v < v_T$. For instance, if $<_p$ is the relation defined by (p) where 'implies' is replaced with 'iff', then we can interpret v_F and v_T as a contradictory and a tautological verifier, respectively, i.e., $v_T \in C_p(\emptyset)$ and $C_p(\{v_F\}) = L_p$.

LEMMA 5.3 (Stachniak & O'Hearn, 1990): *Let $\alpha(p) \in L_p$. Then:*

(i) *If every occurrence of p in $\alpha(p)$ is positive, then*

$$\alpha(p/v_T) \in C_p(\alpha(p/v)) \subseteq C_p(\{\alpha(p/v_F)\}).$$

(ii) *If every occurrence of p in $\alpha(p)$ is negative, then*

$$\alpha(p/v_F) \in C_p(\alpha(p/v)) \subseteq C_p(\{\alpha(p/v_T)\}).$$

This notion of polarity can be used to restrict the application of the resolution rule in various ways. We shall consider two completeness-preserving restrictions. We start with (pol1) introduced in Stachniak & O'Hearn (1990).

Let us fix an enumeration $v_0, ..., v_n$ of Ver_p so that $v_0 = v_F$ and $v_n = v_T$.

(pol1) *An instance*

$$\frac{\beta_0(p)...\beta_n(p)}{\beta_0(p/v_0) \vee ... \vee \beta_n(p/v_n)}$$

of Rv_{er_p} can be used during the deductive process only if the following condition is satisfied:

- *every occurrence of p in* β_0 (β_n) *is positive (negative), if for some* $0 \leq i \leq n$, *every occurrence of p in* β_i *is positive (negative).*

We say that Rs_p admits (pol1), if for every finite set X of formulas, $X \overset{Rs_p}{\Rightarrow} \square$ *iff* $\square$ can be deduced from X using the inference rules of Rs_p, and the resolution rule of Rs_p is restricted by (pol1).

THEOREM 5.4: *Rs_p admits (pol1).*

PROOF: We repeat the argument presented in the proof of Theorem 5.1 with one modification: when turning an inference node into a failure node we must ensure that the application of the resolution rule satisfies (pol1). To this end, suppose K is an inference node in a semantic tree, whose children $K_0, ..., K_n$ falsify $\alpha_0(p), ..., \alpha_n(p)$, respectively in the matrix $< \mathcal{A}, \{D_j\} >$. Suppose that for some $0 \leq l \leq n$, every occurrence of p in α_l is positive. By Theorem 3.2(i), $f[p/v_l](\alpha_l(p)) \notin D_j$, where for every $v, f[p/v]$ is defined as follows: for every variable q,

$$f[p/v](q) = \begin{cases} f(q) & \text{if } p \neq q \\ v & \text{otherwise.} \end{cases}$$

Moreover, by Lemma 5.3, $\alpha_l(p/v_l) \in C_{<\mathcal{A},\{D_j\}>}(\alpha_l(p/v_F))$. This means that $f(\alpha_l(p/v_F)) = f[p/v_F](\alpha_l(p)) \notin D_j$, and we can assume that $\alpha_0 = \alpha_l$. By a similar argument, we can assume that $\alpha_n = \alpha_l$, if every occurrence of p in α_l is negative. Hence, K is a failure node. $\square$

We can further restrict the application of the resolution rule by requiring that:

(pol2) *An instance*

$$\frac{\beta_0(p)...\beta_n(p)}{\beta_0(p/v_0) \vee ... \vee \beta_n(p/v_n)}$$

of Rv_{er_p} can be used during the deductive process only if it satisfies (pol1) as well as the following condition:

there is $0 \leq i \leq n$ *and an occurrence of p in* β_i *of no polarity, or there are* $0 \leq i, j \leq n$, *such that p occurs positively in* β_i *and negatively in* β_j.

The following two rules are needed for completeness:

$$R^+ : \frac{\alpha(p^+)}{\alpha(p/v_T)} \qquad\qquad R^- : \frac{\alpha(p^-)}{\alpha(p/v_F)}.$$

('p^+' and 'p^-' denote the fact that all occurrences of p in α are positive and negative, respectively.)

We say that Rs_p admits (pol2), if for every finite set X of formulas, $X \overset{Rs_p}{\Rightarrow} \square$ *iff* $\square$ can be deduced from X using the inference rules of Rs_p, the rules R^+ and R^-, and the resolution rule of Rs_p is restricted by (pol2).

THEOREM 5.5: *Rs_p admits (pol2).*

PROOF: Let X be clean in Rs. In light of Lemma 5.3, if all occurrences of a variable p in a formula $\alpha(p)$ is positive (is negative), then $\alpha(p/v_T) \in C_p(\alpha(p))$ $(\alpha(p/v_F) \in C_p(\alpha(p)))$. Hence, if $C_p(X) \neq L$, then $\Box$ cannot be derived from X in Rs restricted by (pol2). Namely, the two new rules preserve consistency of X.

Now, suppose that $C_p(X) = L$. We repeat the argument presented in the proof of Theorem 5.4 with the following modification. Let K be an inference node of a semantic tree T associated with a set X_k. Every child K_l of $K, 0 \leq l \leq n$, falsifies some formula $\alpha_l(p_i) \in X_k$ in the matrix $< \mathcal{A}, \{D_j\} >$. In particular, K_n falsifies $\alpha_n(p_i)$, where p_i is interpreted as v_T. Suppose that all occurrences of p_i in every α_l are positive. We can apply the rule R^+ to α_n to form $X \cup \{\alpha_n(p_i/v_T)\}$, turning K into a failure node. Similarly, if all occurrences of p_i in every α_l are negative, then we use the rule R^- to turn K into a failure node. If neither of these cases apply for K, (pol2) will be satisfied and we can use an argument similar to that used in the proof of Theorem 3.3 to show that the resolution rule can be applied in a way satisfying (pol1). $\Box$

It is an easy exercise to verify that the combination of (pol1) and set of support, or (pol2) and set of support, preserves refutational completeness of resolution counterparts of structural propositional logics.

To generalize the results of Theorems 5.4 and 5.5 to FFO logics, we only need to define the notion of polarity in a way that does not interfere with the Lifting Lemma. Let t be a 1-1 function mapping the set of atomic formulas of $\mathcal{P}$ onto the set of all propositional variables of $\mathcal{L}_p$. The following definition suffices:

An occurrence of β in a formula α of $\mathcal{P}$ has positive (negative) polarity iff the corresponding occurrence of $t(\beta)$ in $t(\alpha)$ has positive (negative) polarity.

In the case of classical logic, this definition agrees with the classical definition of polarity (cf. Murray (1982)). We can now define the polarity strategies (pol1) and (pol2) similarly to the propositional case:

(pol1) *An instance*

$$\frac{\beta_0(\phi_0)...\beta_n(\phi_n)}{\theta(\beta_0)(\theta(\phi_0)/v_0) \vee ... \vee \theta(\beta_n)(\theta(\phi_n)/v_n)}$$

of Rv_{er} can be used during the deduction process only if the following condition is satisfied:

- *every occurrence of $\theta(\phi_0)$ in $\theta(\beta_0)$ (in $\theta(\beta_n)$) is positive (negative), if for some $0 \leq i \leq n$, every occurrence of $\theta(\phi_0)$ in $\theta(\beta_i)$ is positive (negative).*

The generalization of (pol2) to FFO-logics is left to the reader.

Our definition of the polarity strategies for FFO-logics ensures that the rules R^+ and R^- of (pol2) are sound and that refutation in Rs_p satisfying (pol1) or (pol2) can be lifted to a refutation in Rs restricted by the polarity strategies. Therefore, if the resolution counterpart Rs of $\mathcal{P}$ is obtained from Rs_p in the way described in Section 4, we have:

THEOREM 5.6: *Rs admits (pol1) and (pol2).*

The reader may check that the combination of either of the polarity strategies with the set of support strategy preserves refutational completeness.

We illustrate the application of the theorem proving strategies discussed in this paper with the following example.

EXAMPLE: Let $\mathcal{L}_3$ be a first-order language with infinitely many ground atomic formulas and with the following connectives: $\vee$ and $\rightarrow$ (binary), and $\neg$ and T (unary). Let $\mathcal{P}_3 =< \mathcal{L}_3, C_3 >$ be the FFO logic defined by the matrix $\mathcal{M}_3 =< \mathcal{A}_3, \{1\} >$, where $\mathcal{A}_3$ is the algebra $< \{0, \frac{1}{2}, 1\}, I, max, min, n, t >$ with the operations defined as follows:

$$I(a,b) = min(1, 1-a+b),$$

max and min are the binary maximum and minimum operations,

$$n(a) = 1-a,$$

$$t(a) = \tfrac{1}{2}.$$

For simplicity, we identify the operation t with the constant $\frac{1}{2}$. Hence, we may assume that T is a logical constant. The connectives of $\mathcal{L}_3$ are interpreted in $\mathcal{A}_3$ as max, I, n, and t, respectively. The logic $\mathcal{P}_3$ is known as Slupecki's logic (or as Slupecki's variant of the three-valued Lukasiewicz logic, cf. Rescher (1976)). $\mathcal{P}_3$ has a resolution counterpart consisting of three verifiers:

$$v_0 = \neg(\tfrac{1}{2} \rightarrow \tfrac{1}{2}),$$
$$v_1 = \tfrac{1}{2},$$
$$v_2 = \neg(v_0).$$

The formulas v_0, v_1, and v_2 define the truth-values 0, $\frac{1}{2}$, and 1 of $\mathcal{M}_3$, respectively. Let $\mathcal{B}_3 =< \{v_0, v_1, v_2\}, \rightarrow, \vee, \neg, \frac{1}{2} >$, where each operation on $\{v_0, v_1, v_2\}$ is determined by the corresponding operation of $\mathcal{A}_3$. For example, $v_0 \vee v_2 = v_2$ since $max(0,1) = 1$. Finally, let $\mathcal{F}_3 = \{V \subseteq \{v_0, v_1, v_2\} : v_0 \in V \text{ or } v_1 \in V\}$ and let $Rs_3 =< \mathcal{B}_3, \mathcal{F}_3 >$. Clearly, Rs_3 is a resolution proof system on $\mathcal{L}_3$. We leave it to the reader to verify that Rs_3 is a resolution counterpart of $\mathcal{P}_3$.

In addition to the 'ground' transformation rules of Rs_3, it is useful to include more general transformation rules that correspond to polynomial equations true in the algebra $\mathcal{B}_3$; some sample rules are:

$$\alpha \rightarrow v_2 \Rightarrow v_2,$$
$$\alpha \rightarrow v_0 \Rightarrow \alpha,$$
$$v_0 \rightarrow \alpha \Rightarrow v_2,$$
$$v_2 \rightarrow \alpha \Rightarrow \alpha,$$
$$v_0 \vee \alpha \Rightarrow \alpha,$$
$$\alpha \vee \alpha \Rightarrow \alpha.$$

Let $<$ be the binary relation on the propositional language corresponding to $\mathcal{L}_3$, defined in the following way:

$$\alpha < \beta \text{ iff } h(\alpha \rightarrow \beta) = 1, \text{ for every valuation } h \text{ in } \mathcal{M}_3.$$

We choose $<$ as our polarity relation. Clearly, $<$ satisfies (p). In addition,

$\vee$ is positive over both arguments,

$\rightarrow$ is negative over its first argument and positive over its second,

$\neg$ is negative over its only argument.

Some care must be taken when choosing a polarity relation. For example, if we had defined $<$ as:

$\alpha < \beta$ iff $\beta \in C_{\mathcal{M}_3}(\{\alpha\})$,

then $\rightarrow$ and $\neg$ would have no polarity over their arguments.

Now, we use Rs_3 to refute the set consisting of the following formulas:

(1) $P(x) \rightarrow Q(f(x))$,

(2) $\neg Q(x) \vee R(x)$,

(3) $(P(x) \rightarrow R(x)) \dot{\rightarrow} \neg(P(x) \rightarrow R(x))$.

Initially, there are 24 possible applications of the resolution rule to these formulas. If we choose $\{(3)\}$ as the set of support, then the set of support strategy blocks 10 of these applications, many of which lead to tautologies. The (pol1) strategy blocks 8 of the remaining 14 applications while (pol2) blocks 10 of the 14 applications (the two extra applications blocked by (pol2) involve self-resolving (3) with itself). Thus the combination of set of support and (pol1) leaves us with 6 possible initial applications of resolution, and the combination with (pol2) leaves us with only 4, out of 24 initial possible applications. In the derivation presented below, we leave out obvious applications of reduction rules.

(4) $\neg(Q(x)) \vee \neg(Q(f(x))) \vee v_1$ (resolvent of 2,2,3)

(5) $P(f(x))$ (resolvent of 1,4,4)

(6) v_1 (resolvent of 3,5,5)

(7) $\square$ (using $\square$-rule $\{v_1\} \Rightarrow \square$.)

This derivation is consistent with both polarity strategies and with set of support.

While (pol2) restricts the resolution rule more severely than does (pol1), we need to deal with the two extra rules R^+ and R^-. If applied carelessly, these rules can generate tautologies, which are useless for refutation purposes. For example, R^+ can be applied to the formula $q \rightarrow p$ to obtain $q \rightarrow v_2$, which is a tautology. However, in some cases the judicious use of these rules can save space and time by leading to shorter proofs with fewer reductions. For example, if we try to refute

(8) $\neg(P(x) \rightarrow (Q(y) \rightarrow P(x))$,

then (pol2) strategy will not allow us to self resolve (8) with itself upon Q. Rather, we can use R^+ to infer

(9) $\neg(P(x) \rightarrow (v_2 \rightarrow P(x))$

which reduces to

(10) $\neg(P(x) \rightarrow P(x))$.

(10) can be self resolved to obtain a contradiction. By using R^+ we avoid one resolution step and a number of reductions which would be forced by this additional application of the resolution rule. □

Although the above example illustrates the power of the set of support and polarity strategies in restricting the search space, it should be pointed out that even with their use resolution based reasoning systems may deduce far too many formulas. Hence, further speed-up techniques are needed for more effective direction of the deductive process. Among the strategies formulated for classical logic that can be almost immediately adopted to FFO logics are the well-known *weighting strategy* (McCharen *et. al.*, 1976) and *unit preference strategy* (Wos *et. al.*, 1964) whose non-clausal counterpart for FFO logics can be formulated, roughly speaking, by allowing the number of atoms occurring in a formula of the deduced set, but not occurring in verifiers, to play the role of the length of a clause.

Acknowledgments

We thank the anonymous referees for their helpful comments and suggestions that led to many improvements in the presentation of the paper. O'Hearn was supported in part by a Natural Sciences and Engineering Research Council of Canada Postgraduate Scholarship. Stachniak's research was supported by the Natural Sciences and Engineering Research Council of Canada.

References

Abadi, M., Manna, Z. (1986). Modal theorem proving. *Proceedings of the 8th International Conference on Automated Deduction, LNCS* **230**, 172-189.

Belnap, N. (1977). A useful four-valued logic. In: (Dunn, M., Epstein, G, eds.) *Modern Uses of Multiple-Valued Logic*. D. Reidel Publishing Company.

Chan, M. (1987). The recursive resolution method for modal logic. *New Generation Computing* **5**, 155-183.

Current, K., ed. (1991). *Proceedings of the 21th International Symposium on Multiple-Valued Logic*, Victoria.

Fariñas del Cerro, L., Herzig, A. (1988). Linear modal deductions. *Proceedings of the 9th International Conference on Automated Deduction, LNCS* **310**, 487-499.

Fitting, M. (1988). Logic programming on a topological bilattice. *Fundamenta Informaticae* **11**, 209-218.

Hawranek, J., Tokarz, M. (1977). Matrices for predicate logics. *Bulletin of the Section of Logic PAN* **6**, 27-31.

Ginsberg, M. (1986). Multi-valued logics. *AAAI-86, Fifth National Conference on Artificial Intelligence*, Los Altos, 243-247.

Łoś, J., Suszko, R. (1958). Remarks on sentential logics. *Indagationes Mathematicae* **20**, 177-183.

Manna, Z., Waldinger, R. (1986). Special relations in automated deduction. *JACM* **33**, 1-59.

McCharen, J., Overbeek, R., Wos, L. (1976). Complexity and related enhancements for automated theorem-proving programs. *Computers and Mathematisc with Applications* **2**, 1-16.

Morgan, C. (1976) A resolution principle for a class of many-valued logics. *Logique et Analyse* **74-76**, 311-339.

Murray, N. (1982). Completely non-clausal theorem proving. *Artificial Intelligence* **18**, 67-85.

Mycroft, A. (1988). Multivalued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence* **4**, 265-316.

O'Hearn, P., Stachniak, Z. (1989a). A resolution framework for finitely-valued first-order logics. *Proceedings of the 2nd Scandinavian Conference on Artificial Intelligence*, Tampere, 78-90.

O'Hearn, P., Stachniak, Z. (1989b). Note on theorem proving strategies for resolution counterparts of finitely-valued logic. *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, Portland, 364-372.

Ohlbach, H. (1988). A resolution calculus for modal logics. *LNCS* **310**, 500-516.

Orlowska, E. (1985). Mechanical theorem proving for post logics. *Logique et Analyse* **110-111**, 173-192.

Przymusinski, T. (1989). Three-valued non-monotonic formalisms and logic programming. *Proceedings of the First International Conference on Principles of Knowledge Representation*, Toronto, 341-348.

Rescher, N. (1969). *Many-Valued Logics*. McGraw-Hill.

Schmitt, P. (1986). Computational aspects of three-valued logic. *Proceedings of the 8th International Conference on Automated Deduction, LNCS* **230**, 190-198.

Silio, C., ed. (1990). *Proceedings of the 20th International Symposium on Multiple-Valued Logic*, Charlotte.

Stachniak, Z. (1991). Minimization of resolution proof systems. *Fundamenta Informaticaei* **14**, 129-146.

Stachniak, Z. (1988). Resolution rule: an algebraic perspective. *Proceedings of Algebraic Logic and Universal Algebra in Computer Science Conference*, Ames, *LNCS* **425**, 227-242.

Stachniak, Z., O'Hearn, P. (1990). Resolution in the domain of strongly finite logics. *Fundamenta Informaticae* **13**, 333-351.

Waszkiewicz, J., Weglorz, B. (1969). On products of structures for generalized logics. *Studia Logica* **25**, 7-15.

Wos, L. (1988). *Automated Reasoning. 33 Basic Research Problems*. Prentice Hall.

Wos, L., Robinson, G., Carson, D. (1965). Efficiency and completeness of the set of support strategy in theorem proving. *Journal of ACM* **12**, 536-541.

Wos, L., Robinson, G., Carson, D. (1964). The unit preference strategy in theorem proving. *Proceedings of the AFIPS Conference* **26**, Spartan Books, Washington, DC, 615-621.

Wójcicki, R. (1988). *Theory of Logical Calculi: Basic Theory of Consequence Operations*. Kluwe Academic Publishers.