A deductive system for existential least fixpoint logic

Kevin J. Compton

INRIA

UNITE DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.: (1) 39 63 55 11

Rapports de Recherche

1992

ème anniversaire

N° 1675

Programme 2
Calcul Symbolique, Programmation
et Génie logiciel

A DEDUCTIVE SYSTEM FOR EXISTENTIAL LEAST FIXPOINT LOGIC

Kevin COMPTON

Mai 1992

RR.1675

A Deductive System for Existential Least Fixpoint Logic

Kevin Compton

Abstract. *Existential least fixpoint logic* (ELFP) is a logic with a least fixpoint operator but only existential quantification. It arises in many areas of computer science including logic programming, database theory, program verification, complexity theory, and recursion theory on abstract structures. A sequent calculus (Gentzen-style deductive system) for this logic is presented and proved to be complete. Basic model theoretic facts about ELFP are derived from the completeness theorem and the construction used in its proof. The relationship of these model theoretic facts to logic programming and database queries is explored.

Un Système Déductif pour la Logique de Plus Petit Point Fixe Existentiel

Résumé. *La logique de plus petit point fixe existentiel* (ELFP) est une logique qui possède un opérateur de plus petit point fixe, avec seulement une quantification existentielle. Elle apparaît dans de nombreux domaines de l'informatique, programmation logique, théorie des bases de données, vérification de programme, théorie de la complexité, et théorie de la récursion des structures abstraites. Nous présentons un calcul des séquents (système déductif à la Gentzen) pour cette logique et prouvons qu'il est complet. Les propriets de base du modèle théorique concernant la ELFP sont dérivée du théorème de complétude et de la construction utilisée dans sa démonstration. Nous explorons la relation entre ces propriets de theorié de modéles et la programmation logique d'une part, les requêtes de bases de données d'autre part.

A Deductive System for Existential Least Fixpoint Logic

Kevin J. Compton
INRIA, Rocquencourt
78153 Le Chesnay Cedex
FRANCE
compton@margaux.inria.fr

10 April 1992

Abstract

Existential least fixpoint logic (*ELFP*) is a logic with a least fixpoint operator but only existential quantification. It arises in many areas of computer science including logic programming, database theory, program verification, complexity theory, and recursion theory on abstract structures. A sequent calculus (Gentzen-style deductive system) for this logic is presented and proved to be complete. Basic model theoretic facts about *ELFP* are derived from the completeness theorem and the construction used in its proof. The relationship of these model theoretic facts to logic programming and database queries is explored.

1 Introduction

To express logic program queries on relational databases, Chandra and Harel [4] formulated a logic with a least fixpoint operator but only existential quantification. Blass and Gurevich [3] rediscovered the same logic, which they called existential fixpoint logic, in an effort to overcome some of the limitations of first-order logic as an assertion language for program verification. Modifying their terminology slightly, we refer to this logc as *existential least fixpoint logic* or *ELFP*. Blass and Gurevich showed that weakest preconditions and strongest postconditions for an imperative programming language with recursive procedures are expressible in this logic. They also proved that for classes of finite structures with an underlying successor relation and constants for the first and last elements, polynomial time computability is characterized by satisfaction of *ELFP* sentences. This strengthens a well known characterization of polynomial time computability due to Immerman [9] and Vardi [20]. *ELFP* is also equivalent to the systems of existential inductive definitions first studied by Aczel [1]. His motivation was to define analogues of recursively enumerable sets on abstract structures.

A logic that arises in such widely divergent areas as logic programming, database theory, program verification, complexity theory, and recursion theory on abstract structures is surely fundamental to computation. We will present a sequent calculus (Gentzen-style deductive system) for this logic and prove that it is complete. We also prove some basic model theoretic facts about *ELFP*. Most of these follow from the completeness theorem and the construction used in its proof. Finally, we will show how some of these model theoretic facts relate to logic programming and database queries.

By way of introduction, let us consider the following logic program query about membership of a an element c in the set generated from an element a by closing under a binary function f.

```
P(x) :- x=a.
P(x) :- x=f(y,z),P(y),P(z).
-?  P(c).
```

This program has two parts: an inductive definition of a relation P, and a query referring to this relation. The inductive definition says that P is the smallest unary relation containing a and such that $f(y,z)$ belongs to P whenever y and z belong to P. We may write this as

$$P(x) \equiv (x = a) \vee \exists y, z\, (x = f(y, z) \wedge P(y) \wedge P(z)).$$

Formally, the definition of P is a disjunction of formulas corresponding to the clauses in the program where P appears in the head. Each formula is formed by taking conjunctions of the literals in the body of the clause and existentially quantifying all variables not occurring the head. We form the *ELFP* sentence corresponding to the program by inserting the inductive definition before the query. We have

$$[P(x) \equiv \vartheta(P, x)]\, P(c)$$

where $\vartheta(P, x)$ is the formula $(x = a) \vee \exists y, z\, (x = f(y, z) \wedge P(y) \wedge P(z))$. (Blass and Gurevich use LET $P(x) \leftarrow \vartheta(P, x)$ THEN ψ rather than $[P(x) \equiv \vartheta(P, x)]\, \psi(P)$.)

Since P occurs only positively in $\vartheta(P, x)$, on each structure $\mathfrak{A}$ with universe A the function $F_\vartheta{:}2^A \to 2^A$ given by $F_\vartheta(R) = \{a : \mathfrak{A} \models \vartheta[R, a]\}$ is monotone. Hence, F_ϑ has a least fixpoint, which is the interpretation of P in the query $P(c)$. (See [13] for background on least fixpoint theorems.)

Thus, we build formulas of *ELFP* using conjunction, disjunction, existential quantification, and inductive definition. *ELFP* is a restricted second-order logic since relation variables (such as P in the example above) may occur in formulas. We will allow negation to be applied to formulas with no relation variables, so *ELFP* is equivalent to the logic Chandra and Harel call YE. If we did not allow negations at all, we would have the logic that Chandra and Harel call YE+.

As we shall see, *ELFP* is not compact, so no finitary deductive system for it can be complete. Some sort of infinitary rule is needed, and it is not difficult to see what kind of rule this should be when we observe that *ELFP* is really a "sublogic" of $L_{\omega_1\omega}$. This observation was made earlier by Park [17] about a more expressive logic than *ELFP*. Park called this logic the *formally continuous μ-calculus.* In another paper [6] we will consider the formal properties of this logic, which we call *stratified least fixpoint logic* because it is closely related to stratified logic programs.

$L_{\omega_1\omega}$ is possibly the most studied extension of first-order logic. It allows countably infinite conjunctions and disjunctions. At first glance, this appears to be quite different from *ELFP*, which extends the existential fragment of first-order logic by allowing inductive definitions. However, as Park and others have observed, the closure ordinal for an inductive definition occurring in an *ELFP* formula is at most ω. From this it follows that every *ELFP* sentence is equivalent to a countable disjunction of simpler *ELFP* sentences. Moreover, since negation may be applied only to formulas without quantifiers or inductive definitions, we do not need the countable conjunctions of $L_{\omega_1\omega}$ to express *ELFP* formulas. Consequently, our deductive system will require only one infinitary rule which is used to introduce inductive definitions on the left sides of sequents. This will have implications for us when return to the relationship between *ELFP* and logic programming at the end of the paper.

2 Description of the Logic

In this section we give a formal description of *ELFP*. We begin with the syntax. Fix a vocabulary V of constant, function, and relation symbols.

Definition. The symbols in *ELFP* formulas include the usual symbols from first-order logic, excluding universal quantifiers, and also *relation variables.* Element variables will be specified by lower case letters such as x, y, z, x_1, x_2, while relation variables will be specified by upper case letters such as P and Q. Each relation variable P has a specified arity and there are countably many relation variables of each finite arity. The set $\mathcal{F}$ of *ELFP* formulas φ over V is the least set satisfying the following conditions.

(i) If φ is an atomic formula over V, then φ is in $\mathcal{F}$. The logic always includes equality, so $t_1 = t_2$ is an atomic formula for every pair of terms t_1, t_2. If P is a relation symbol (in V) or a relation variable of arity k, and $\vec{x} = (x_1, \ldots, x_k)$ is a sequence of element variables, then $P(\vec{x})$ is in $\mathcal{F}$.

(ii) If ψ is a formula in $\mathcal{F}$ containing no relation variables or quantifiers, then $(\neg\psi)$ is in $\mathcal{F}$.

(iii) If ψ and ϑ are in $\mathcal{F}$, so are $(\psi \vee \vartheta)$ and $(\psi \wedge \vartheta)$.

(iv) If ψ is in $\mathcal{F}$ and x is an element variable, then $(\exists x\, \psi)$ is in $\mathcal{F}$.

(v) If ψ and ϑ are in $\mathcal{F}$, P is a relation variable of arity k, and $\vec{x} = (x_1, \ldots, x_k)$ is a sequence of distinct element variables, then $([P(\vec{x}) \equiv \vartheta]\, \psi)$ is in $\mathcal{F}$. The initial part of the formula, viz., $[P(\vec{x}) \equiv \vartheta]$, is called an *inductive definition.*

We follow the usual conventions for deleting parentheses in formulas.

Before we can give the semantics for *ELFP*, we need to define the notion of a free variable and of free occurrence of a variable in an *ELFP* formula.

Definition. For each *ELFP* formula φ define *free*(φ), the set of free variables in φ, and the *free occurrences* of variables in φ. When φ is atomic, *free*(φ) is the set of element and relation variables in φ; all occurrences of variables in φ are free. Free variables in formulas constructed using logical connectives and quantifiers are handled in the usual way. Also,

$$free([P(\vec{x}) \equiv \vartheta]\, \psi) = \Big((free(\vartheta) - \{x_1, \ldots, x_j\}) \cup free(\psi)\Big) - \{P\}.$$

The free occurrences of variables in $([P(\vec{x}) \equiv \vartheta]\, \psi)$ are the free occurrences of variables of $free(\vartheta) - \{P, x_1, \ldots, x_j\}$ in ϑ and the free occurrences of variables from $free(\psi) - \{P\}$ in ψ. A *sentence* is a formula with no free variables.

When we write $\varphi(x/t)$ we mean that term t has been substituted for all free occurrences of the element variable x in φ. All uses of this notation are subject to the proviso that occurrences of variables in t are free wherever t is substituted. In the case where t is just a single variable y we write $\varphi(y)$ rather than $\varphi(x/y)$. The notation $\neg\varphi$ is defined only when φ is quantifier free and contains no relation variables. The notation $\varphi(P/\rho)$ means that all subformulas of φ containing free occurrences of the relation variable P are replaced by formula ρ. To be precise, we should specify a sequence of k distinct element variables in ρ, where k is the arity of P; the correspondence between element variables of P and element variables of ρ will always be clear from context. All uses of this notation are subject to the proviso that free occurrences of variables in ρ remain free wherever ρ is substituted.

We now give the semantics of *ELFP* formulas. As usual, we define by induction on φ the relation $\mathfrak{A} \models \varphi[\alpha]$ ($\mathfrak{A}$ *satisfies* φ *at* α), where α is an assignment in $\mathfrak{A}$. More precisely, suppose φ has free relation variables $P_1, \ldots, P_k$, with respective arities $j_1, \ldots, j_k$, and free element variables $x_1, \ldots, x_l$. Fix a structure $\mathfrak{A}$. An assignment α for φ can be represented as a sequence $(R_1, \ldots, R_k, a_1, \ldots, a_l)$, where each R_i is a j_i-ary relation on $\mathfrak{A}$ and each a_i is an element of $\mathfrak{A}$. With φ we will asscciate a function F_φ mapping sequences $(R_1, \ldots, R_k)$ to l-ary relations on $\mathfrak{A}$:

$$F_\varphi(R_1, \ldots, R_k) = \{(a_1, \ldots, a_l) \mid \mathfrak{A} \models \varphi[R_1, \ldots, R_k, a_1, \ldots, a_l]\}.$$

Simultaneously with our inductive definition of satisfaction, we also show that F_φ is *continuous*; i.e., that

$$\bigcup_{\alpha<\lambda} F_\varphi(R_{1\alpha}, \ldots, R_{k\alpha}) = F_\varphi(\bigcup_{\alpha<\lambda} R_{1\alpha}, \ldots, \bigcup_{\alpha<\lambda} R_{k\alpha})$$

for all chains $(R_{i\alpha} \mid \alpha < \lambda)$ of j_i-ary relations. Notice that if F_φ is continuous, it is *monotone* as well; i.e., $F_\varphi(R_1, \ldots, R_k) \subseteq F_\varphi(R'_1, \ldots, R'_k)$ whenever $R_1 \subseteq R'_1, \ldots, R_k \subseteq R'_k$. By a continuous (or monotone) formula, we mean a formula φ such that F_φ is continuous (or monotone).

If φ is atomic, $\mathfrak{A} \models \varphi[\alpha]$ is defined in the usual way and F_φ is clearly continuous. Also, if φ is a disjunction, conjunction, negation, or (existentially) quantified formula, $\mathfrak{A} \models \varphi[\alpha]$ is again defined in the usual way, and it is not difficult to see that φ is continuous. (Notice, however, that it is crucial that negations may not be applied to formulas with free relation variables.)

Let us define $\mathfrak{A} \models \varphi[\alpha]$ when φ is of the form $[P(\vec{x}) \equiv \vartheta]\,\psi$ assuming that ϑ and ψ are continuous and their truth values have been defined for the assignment α. Let the assignments to variables other than P and $\vec{x} = (x_1, \ldots, x_k)$ be given by the assignment α. We thereby obtain from F_ϑ a continuous mapping G from k-ary relations to k-ary relations. G is monotone and hence has a least fixed-point by the Least Fixpoint Theorem (or at least by one of the theorems that go by this name; see Lassez, Nguyen, and Sonenberg [13]).

The well known construction of the least fixed-point of a monotone function is as follows. Let $G^0(R) = R$, $G^{\beta+1}(R) = G(G^\beta(R))$ and if β is a limit ordinal, $G^\beta(R) = \bigcup_{\gamma<\beta} G^\gamma(R)$. By induction $G^\beta(\emptyset) \subseteq G^\gamma(\emptyset)$ whenever $\beta < \gamma$. There is a smallest ordinal κ (called the *closure ordinal* of the inductive definition $[P(\vec{x}) \equiv \vartheta]$) such that $G^\beta(\emptyset) = G^\kappa(\emptyset)$ whenever $\beta \geq \kappa$. $G^\kappa(\emptyset)$ is the least fixed-point of G. Since G is continuous, it follows that $\kappa \leq \omega$ (see [13]). Thus, $\mathfrak{A} \models \varphi[\alpha]$ holds in case $\mathfrak{A} \models \psi[\alpha']$, where α' is identical to α except that it assigns $G^\omega(\emptyset)$ to P.

It will be useful to define, for each nonnegative integer m, the formula

$$[P(\vec{x}) \equiv \vartheta]_m\, \psi.$$

$\mathfrak{A} \models [P(\vec{x}) \equiv \vartheta]_m \psi[\alpha]$ holds just in case $\mathfrak{A} \models \psi[\alpha'']$, where α'' is identical to α except that it assigns $G^m(\emptyset)$ to P. We regard this formula as an abbreviation. Construct a sequence of formulas $\rho_0, \rho_1, \rho_2, \ldots$, where ρ_0 is the formula $\exists x\, (\neg x{=}x)$ and ρ_{m+1} is the formula $\vartheta(P/\rho_m)$. Then $[P(\vec{x}) \equiv \vartheta]_m \psi$ is an abbreviation for $\psi(P/\rho_m)$. Call this formula φ_m.

Since continuity is preserved by composition, each of the functions F_{φ_m} is continuous. Moreover, the sequence $F_{\varphi_0}, F_{\varphi_1}, F_{\varphi_2}, \ldots$ is a chain (in the partial order of function dominance) with supremum F_φ. Since the supremum of a chain of continuous functions is continuous, F_φ is continuous. (See Theorem 4.18 of Loeckx and Sieber [14].) It follows that $[P(\vec{x}) \equiv \vartheta]\, \psi$ is equivalent to the infinite disjunction

$$\bigvee_{m \in \omega} [P(\vec{x}) \equiv \vartheta]_m \psi.$$

We summarize our observations in the following theorem.

Theorem 2.1 *The following hold for* ELFP.

(i) All formulas are continuous (and hence monotone).

(ii) The closure ordinal of any inductive definition is at most ω.

(iii) $[P(\vec{x}) \equiv \vartheta]\, \psi$ is equivalent to $\bigvee_{m\in\omega}[P(\vec{x}) \equiv \vartheta]_m \psi$. Thus every sentence is equivalent to a sentence of $L_{\omega_1\omega}$.

As we noted in the introduction, this theorem was first proved by Park [17] for a more general logic, the formally continuous μ-calculus. De Roever described a similar logic around the same time and made the observation that the sentences of his logic were "syntactically continuous" (see [7]). Part (ii) of the theorem was observed by Aczel [1] for systems of existential inductive definitions. Blass and Gurevich observed (ii).

In the following section there is a somewhat delicate induction on formula complexity. Accordingly, we assign an ordinal to each *ELFP* formula.

Definition. With every *ELFP* formula φ we associate an ordinal $\lambda(\varphi)$ as follows. If φ is atomic, $\lambda(\varphi) = 0$. Also, $\lambda(\neg\psi) = \lambda(\exists y\, \psi) = \lambda(\forall y\, \psi)\lambda(\psi) + 1$ and $\lambda(\psi \vee \vartheta) = \lambda(\psi \wedge \vartheta) = \max(\lambda(\psi), \lambda(\vartheta)) + 1$. Finally, $\lambda([P(\vec{x}) \equiv \vartheta]\, \psi) = \sup_{m\in\omega} \lambda([P(\vec{x}) \equiv \vartheta]_m \psi + 1)$.

Remark. Our definition of *ELFP* differs from the one given by Blass and Gurevich [3] in two nonessential ways.

First, they allow negation to be applied only to atomic formulas, whereas we allow negation to be applied to quantifier-free formulas with no relation variables. This does not change the expressive power of the logic since any quantifier-free formulas with no relation variables can be easily transformed into an equivalent formula in which only atomic formulas are negated.

Second, they allow simultaneous inductive definitions. That is, rather than a single relation variable P and formula ϑ, they allow multiple relation variables and formulas in inductive definitions. Thus, they allow formulas of the form

$$[P_1(\vec{x}_1) \equiv \vartheta_1; \cdots; P_k(\vec{x}_k) \equiv \vartheta_k]\, \psi.$$

Again, this does not change the expressive power of the logic. Blass and Gurevich observe that a formula with simultaneous inductive definitions may always be transformed into an equivalent formula with only inductive definitions (as defined here). This was first proved by Chandra and Harel [4]; their proof was based on a similar result of Moschovakis [16].

3 The Deductive System

In this section we present a sequent calculus for the logic *ELFP*. We call this calculus **LE**.

In **LE** we inductively define a binary relation $\vdash$ holding between finite *sets* of *ELFP* formulas. In Gentzen's sequent calculus **LK** for first-order logic, the relation $\vdash$ holds between *sequences* of formulas. By working with sets we may ignore two of the so-called "weak" rules of inference, viz., the rules of contraction and exchange. (See Takeuti [19].)

We observe the following conventions. Lower case Greek letters denote *ELFP* formulas. Upper case Greek letters denote sets of *ELFP* formulas. Γ, Δ denotes $\Gamma \cup \Delta$. Γ, φ denotes $\Gamma \cup \{\varphi\}$. A *sequent* is an expression of the form $\Gamma \vdash \Delta$. (The semantics for this expression is given below.) In general, a formula φ occurring as part of a sequent denotes the set $\{\varphi\}$. Finally, $t_1 \doteq t_2$ indicates that either $t_1 = t_2$ or $t_2 = t_1$ may be used.

In Gentzen's calculus **LK** the relation $\vdash$ is defined inductively by rules of inference of the form

$$\frac{\Gamma_1 \vdash \Delta_1}{\Gamma \vdash \Delta} \quad \text{or} \quad \frac{\Gamma_1 \vdash \Delta_1 \quad \Gamma_2 \vdash \Delta_2}{\Gamma \vdash \Delta}$$

The first asserts that $\Gamma \vdash \Delta$ holds whenever $\Gamma_1 \vdash \Delta_1$ holds; the second asserts that $\Gamma \vdash \Delta$ holds whenever $\Gamma_1 \vdash \Delta_1$ and $\Gamma_2 \vdash \Delta_2$ hold. In these rules, $\Gamma \vdash \Delta$ is called the *lower sequent* and $\Gamma_1 \vdash \Delta_1$ and $\Gamma_2 \vdash \Delta_2$ are called *upper sequents.*

As usual, $\Gamma \models \varphi$ will mean that every model of Γ satisfies φ and $\Gamma \models \Delta$ will mean that every model of Γ satisfies some formula in Δ. (When Δ is empty, this is interpreted to mean that Γ has no models). We would like to have a sound and complete sequent calculus for *ELFP*— one in which $\Gamma \vdash \Delta$ if and only if $\Gamma \models \Delta$ — with all rules of the forms above. This is not to be. It would imply that *ELFP* is compact, i.e., whenever $\Gamma \models \Delta$, there would be finite sets $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ such that $\Gamma' \models \Delta'$. We will see in the next section that is not true.

We cannot hope to have a complete sequent calculus if lower sequents are inferred from just finitely many upper sequents. Thus, in one of our rules, there will be a countably infinite number of upper sequents. These infinitary rules are expressed in the following form:

$$\frac{\Gamma_m \vdash \Delta_m \quad (m \in \omega)}{\Gamma \vdash \Delta}$$

Definition. The *axioms* of **LE** are the sequents of the form

$$\varphi \vdash \varphi,$$

where φ is a formula of *ELFP*, and

$$\emptyset \vdash t=t,$$

where t is a term.

Definition. The *rules of inference* for **LE** are as follows.

$$(* \vdash) \quad \frac{\Gamma \vdash \Delta}{\Gamma, \Sigma \vdash \Delta} \qquad\qquad (\vdash *) \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \Sigma}$$

$$(S \vdash) \quad \frac{\Gamma, \varphi(x/t_1) \vdash \Delta}{\Gamma, t_1 \doteq t_2, \varphi(x/t_2) \vdash \Delta} \qquad (\vdash S) \quad \frac{\Gamma \vdash \Delta, \varphi(x/t_1)}{\Gamma, t_1 \doteq t_2 \vdash \Delta, \varphi(x/t_2)}$$

$$(\neg \vdash) \quad \frac{\Gamma \vdash \Delta, \psi}{\Gamma, \neg\psi \vdash \Delta} \qquad (\vdash \neg) \quad \frac{\Gamma, \psi \vdash \Delta}{\Gamma \vdash \Delta, \neg\psi}$$

$$(\vee \vdash) \quad \frac{\Gamma, \psi \vdash \Delta \quad \Gamma, \vartheta \vdash \Delta}{\Gamma, \psi \vee \vartheta \vdash \Delta} \qquad (\vdash \vee) \quad \frac{\Gamma \vdash \Delta, \psi, \vartheta}{\Gamma \vdash \Delta, \psi \vee \vartheta}$$

$$(\wedge \vdash) \quad \frac{\Gamma, \psi, \vartheta \vdash \Delta}{\Gamma, \psi \wedge \vartheta \vdash \Delta} \qquad (\vdash \wedge) \quad \frac{\Gamma \vdash \Delta, \psi \quad \Gamma \vdash \Delta, \vartheta}{\Gamma \vdash \Delta, \psi \wedge \vartheta}$$

$$(\exists \vdash) \quad \frac{\Gamma, \psi(x) \vdash \Delta}{\Gamma, \exists y\, \psi(y) \vdash \Delta} \quad x \notin \mathit{free}(\Gamma \cup \Delta) \qquad (\vdash \exists) \quad \frac{\Gamma \vdash \Delta, \psi(x/t)}{\Gamma \vdash \Delta, \exists y\, \psi(y)}$$

$$([\,] \vdash) \quad \frac{\Gamma, [P(\vec{x}) \equiv \vartheta]_m\, \psi \vdash \Delta \quad (m \in \omega)}{\Gamma, [P(\vec{x}) \equiv \vartheta]\, \psi \vdash \Delta} \qquad (\vdash [\,]) \quad \frac{\Gamma \vdash \Delta, [P(\vec{x}) \equiv \vartheta]_m\, \psi}{\Gamma \vdash \Delta, [P(\vec{x}) \equiv \vartheta]\, \psi}$$

Rules $(* \vdash)$ and $(\vdash *)$ are, respectively, the left and right *weakening rules.* Rules $(S \vdash)$ and $(\vdash S)$ are the left and right *substitution rules.* The other rules introduce the various operations on formulas on the left and right sides of sequents. Notice that in the rule $(\vdash [\,])$ there is just one upper sequent: m is a fixed nonnegative integer.

We have not included the familiar *cut rule*

$$\frac{\Gamma, \varphi \vdash \Delta \quad \Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta.}$$

When we show that **LE** is complete we will not use this rule so we will not need to explicitly prove a cut-elimination theorem. This is a matter of taste. We could have presented a somewhat simpler proof of completeness assuming a cut rule, then proved a cut-elimination theorem using techniques similar to those in Tait [18].

Definition. The set of *theorems* of **LE** is the least set of *ELFP* sequents containing the axioms and closed under the rules of inference of **LE**.

Theorem 3.1 (Soundness Theorem) *If* $\Gamma \vdash \Delta$ *is a theorem of* **LE**, *then* $\Gamma \models \Delta$.

Proof. Simply verify the axioms and rules above. This is trivial in all cases except $([\,] \vdash)$ and $(\vdash [\,])$. These follow from Proposition 2.1. □

Definition. A *proof* in **LE** is a tree in which every branch is finite and each node is labeled by a sequent. Moreover, each leaf is labeled by an axiom and each interior node is labeled by the lower sequent of some rule of inference while the labels of its children are precisely the upper sequents of the rule of inference. Thus, nodes may have 0, 1, 2, or ω children.

It is immediate that S is a theorem of **LE** if and only if S is the label on the root of some **LE** proof.

4 Main Result

In this section we will prove that the deductive system **LE** is complete. As we saw earlier, *ELFP* is equivalent to a fragment of $L_{\omega_1\omega}$. C. Karp [11] was the first to prove the completeness of a deductive system for $L_{\omega_1\omega}$. Our system is based on a sequent calculus for $L_{\omega_1\omega}$ due to Lopez-Escobar [15]. One notable feature of this calculus is a proof rule for equality introduction that circumvents some of the usual problems with equality in cut-free sequent calculi. Lopez-Escobar attributes this rule to Maehara and Takeuti.

Our completeness proof is somewhat simpler than Lopez-Escobar's. The definitions and lemmas that follow will be used in our proof. We note that the main difficulty in the proof is in handling equality, not infinite disjunctions. Our approach, which is based on the method of consistency properties, clearly works also to give a completenes proof for a cut-free sequent calculus for first-order logic with equality. The resulting consistency property differes in certain fundamental respects from the consistency properties generally used for logics with equality (see, e.g., Fitting [8] and Keisler [12]).

Assume that V is countable. At the end of the section we sketch the modifications needed to make our results carry over to uncountable vocabularies.

Definition. An equivalence relation $\sim$ on the universe of a structure $\mathfrak{A}$ is a *congruence* on $\mathfrak{A}$ if the following two conditions hold.

(i) If R is a relation symbol in V of arity k and $a_1 \sim b_1, \ldots, a_k \sim b_k$, then $(a_1, \ldots, a_k)$ is in $R^{\mathfrak{A}}$ if and only if $(b_1, \ldots, b_k)$ is in $R^{\mathfrak{A}}$.

(ii) If f is a function symbol in V of arity k and $a_1 \sim b_1, \ldots, a_k \sim b_k$, then $f^{\mathfrak{A}}(a_1, \ldots, a_k) \sim f^{\mathfrak{A}}(b_1, \ldots, b_k)$.

For a given congruence relation $\sim$, let $[a]$ denote the congruence class containing the element a.

Whenever $\sim$ is a congruence on a structure $\mathfrak{A}$, we can form a new structure $\mathfrak{A}/\sim$ called a *quotient structure*. The elements of $\mathfrak{A}/\sim$ are the congruence classes $[a]$. In $\mathfrak{A}/\sim$ define $f^{\mathfrak{A}/\sim}([a_1], \ldots, [a_k]) = [f^{\mathfrak{A}}(a_1, \ldots, a_k)]$ for each function symbol f of arity k in V, and stipulate that $\mathfrak{A} \models R([a_1], \ldots, [a_k])$ if and only if $\mathfrak{A} \models R(a_1, \ldots, a_k)$ for each relation symbol in V. (For these purposes, constant symbols are considered 0-ary function symbols.) It follows from the definition of congruence that relation and function symbols in V have well defined interpretations in $\mathfrak{A}/\sim$.

Definition. An *Herbrand structure* for a variable set X is a structure $\mathfrak{T}(X)$ whose universe consists of terms with variables in X, such that for each function symbol f, $f^{\mathfrak{T}(X)}(t_1, \ldots, t_k)$ is the term $f(t_1, \ldots, t_k)$. Since V may also contain relation symbols, Herbrand structures are not uniquely determined for a given X.

In the completeness proof for **LE** below, we construct a model by taking the quotient of an Herbrand structure. This is a standard technique in foundations of logic programming (see Apt [2], but difficulties arise when we attempt to use this technique to prove completeness theorems for logics with equality. The following definition and technical lemma are to address these difficulties.

Definition. A set Γ of *ELFP* formulas is closed under *substitution* if whenever $t_1 \doteq t_2$ and $\varphi(x/t_1)$ belong to Γ, then so does $\varphi(x/t_2)$.

Note that closure of Γ under substitution implies that $t = t' \in \Gamma$ if and only if $t' = t \in \Gamma$.

Lemma 4.1 *Let Γ be a set of atomic formulas with variables in X and $\mathfrak{T}(X)$ be the Herbrand structure where the interpretation of each relation symbol R is $\{(t_1, \ldots, t_k) \mid R(t_1, \ldots, t_k) \in \Gamma\}$. Suppose Γ is closed under substitution. Define a binary relation $\sim$ on the universe of $\mathfrak{T}(X)$ by*

$$t(x_1/t_1, \ldots, x_k/t_k) \sim t(x_1/t'_1, \ldots, x_k/t'_k)$$

whenever t is a term and $t_1 = t'_1, \ldots, t_k = t'_k \in \Gamma$. Then $\sim$ is a congruence relation on $\mathfrak{T}(X)$. Moreover, for each relation symbol R, $R(t_1, \ldots, t_l) \in \Gamma$ if and only if $\mathfrak{T}(X)/\!\sim\, \models R([t_1], \ldots, [t_l])$.

Proof. We first show that $\sim$ is an equivalence relation. Reflexivity and symmetry are obvious.

Let us prove transitivity. Suppose that t is a term and $t_1 = t'_1, \ldots, t_k = t'_k \in \Gamma$ so that

$$t(x_1/t_1, \ldots, x_k/t_k) \sim t(x_1/t'_1, \ldots, x_k/t'_k).$$

Suppose also that u is a term and $u_1 = u'_1, \ldots, u_l = u'_l \in \Gamma$ so

$$u(x_1/u_1, \ldots, x_l/u_l) \sim u(x_1/u'_1, \ldots, x_l/u'_l).$$

We prove transitivity by showing that if $t(x_1/t'_1, \ldots, x_k/t'_k)$ and $u(x_1/u_1, \ldots, x_l/u_l)$ are the same term, then

$$t(x_1/t_1, \ldots, x_k/t_k) \sim u(x_1/u'_1, \ldots, x_l/u'_l).$$

To do this we must show that there is a term v and $v_1 = v'_1, \ldots, v_m = v'_m \in \Gamma$ such that $t(x_1/t_1, \ldots, x_k/t_k)$ is the same term as $v(x_1/v_1, \ldots, x_m/v_m)$, and $u(x_1/u'_1, \ldots, x_l/u'_l)$ is the same term as $v(x_1/v'_1, \ldots, x_m/v'_m)$. We prove this by induction on the combined depth of t and u.

We may suppose that the variables x_i, where $1 \leq i \leq m$, do not occur in the terms $t_1, \ldots, t_k, t'_1, \ldots, t'_k, u_1, \ldots, u_l, u'_1, \ldots, u'_l$. We may also suppose that at least one of the variables x_i occurs in t, for otherwise $t(x_1/t_1, \ldots, x_k/t_k)$ and $t(x_1/t'_1, \ldots, x_k/t'_k)$ are the same term, and transitivity is immediate. For the same reason we may suppose that at least one of the variables x_i occurs in u.

Consider the case where t has depth 0, i.e., t is a variable x_i. Then $t(x_1/t_1, \ldots, x_l/t_l)$ is just t_i and $t(x_1/t'_1, \ldots, x_l/t'_l)$ is just t'_i. We know that t'_i is the same term as $u(x_1/u_1, \ldots, x_k/u_k)$ so, since $t_i = t'_i \in \Gamma$, we have that $t_i = u(x_1/u_1, \ldots, x_k/u_k) \in \Gamma$. By substitution, $t_i = u(x_1/u'_1, \ldots, x_k/u'_k)) \in \Gamma$ so

$$t(x_1/t_1, \ldots, x_k/t_k) \sim u(x_1/u'_1, \ldots, x_l/u'_l)$$

and transitivity holds in this case.

The case where u has depth 0 is analogous.

Consider the case where neither t nor u has depth 0. Then the outermost function symbol in t must also be the outermost function symbol in u; let this symbol be f. Hence, t is of the form $f(r_1, \ldots, r_n)$ and u is of the form $f(s_1, \ldots, s_n)$. Since $t(x_1/t'_1, \ldots, x_k/t'_k)$ and $u(x_1/u_1, \ldots, x_l/u_l)$

are the same term, $r_i(x_1/t'_1,\dots,x_k/t'_k)$ and $s_i(x_1/u_1,\dots,x_l/u_l)$ are the same term for $1 \leq i \leq n$. By the induction hypothesis, there are terms v_i and equations $v_{i1} = v'_{i1},\dots,v_{im(i)} = v'_{im(i)} \in \Gamma$ such that $r_i(x_1/t_1,\dots,x_k/t_k)$ is the same term as $v_i(x_{i1}/v_{i1},\dots,x_{im(i)}/v_{im(i)})$, and $s_i(x_1/u'_1,\dots,x_l/u'_l)$ is the same term as $v_i(x_{i1}/v'_{i1},\dots,x_{im(i)}/v'_{im(i)})$. We may assume that the variables x_{ij} are distinct for $1 \leq i \leq n$ and $1 \leq j \leq m(i)$. Thus, $t(x_1/t_1,\dots,x_k/t_k)$ is the same term as $f(v_1,\dots,v_n)(x_{11}/v_{11},\dots,x_{nm(n)}/v_{nm(n)})$, and $u(x_1/u'_1,\dots,x_l/u'_l)$ is the same term as $f(v_1,\dots,v_n)(x'_{11}/v_{11},\dots,x'_{nm(n)}/v_{nm(n)})$. It follows that $t(x_1/t_1,\dots,x_k/t_k) \sim u(x_1/u'_1,\dots,x_l/u'_l)$. This concludes the proof of transitivity.

It follows easily from the substitution property of Γ that $\sim$ is a congruence. It is now straightforward to verify that $R(t_1,\dots,t_l) \in \Gamma$ if and only if $\mathfrak{T}(X)/\!\sim\, \models R([t_1],\dots,[t_l])$. □

Definition. Let X be a countable set of element variables. It will be convenient to assume that in the *ELFP* formulas we consider all free variables are in X and all bound variables are not in X. A *consistency property* $\mathcal{P}$ is a nonempty set of pairs $(\Gamma;\Delta)$ where Γ and Δ are sets of *ELFP* formulas such that the following hold for all $(\Gamma;\Delta)$ in $\mathcal{P}$.

(i) Γ and Δ are disjoint.

(ii) If Γ contains formulas $t_1 = t'_1, t_2 = t'_2,\dots,t_k = t'_k$, where $k \geq 0$, then Δ contains no formulas of the form $t(x_1/t_1,\dots,x_k/t_k) = t(x_1/t'_1,\dots,x_k/t'_k)$.

(iii) If $t \doteq t'$ and $\psi(x/t)$ are in Γ, where ψ is an atomic formula, then $(\Gamma,\psi(x/t');\Delta)$ is in $\mathcal{P}$.

(iv) If $\neg\psi$ is in Γ, then $(\Gamma;\Delta,\psi)$ is in $\mathcal{P}$. If $\neg\psi$ is in Δ, then $(\Gamma,\psi;\Delta)$ is in $\mathcal{P}$.

(v) If $\psi \vee \vartheta$ is in Γ, then either $(\Gamma,\psi;\Delta)$ or $(\Gamma,\vartheta;\Delta)$is in $\mathcal{P}$. If $\psi \vee \vartheta$ is in Δ, then $(\Gamma;\Delta,\psi,\vartheta)$ is in $\mathcal{P}$.

(vi) If $\psi \wedge \vartheta$ is in Γ, then $(\Gamma,\psi,\vartheta;\Delta)$ is in $\mathcal{P}$. If $\psi \wedge \vartheta$ is in Δ, then either $(\Gamma;\Delta,\psi)$ or $(\Gamma;\Delta,\vartheta)$ is in $\mathcal{P}$.

(vii) If $\exists y\,\psi(y)$ is in Γ, then $(\Gamma,\psi(x);\Delta)$ is in $\mathcal{P}$ for some $x \in X$. If $\exists y\,\psi(y)$ is in Δ, then $(\Gamma;\Delta,\psi(t))$ is in $\mathcal{P}$ for all terms t.

(viii) If $[P(\vec{x}) \equiv \vartheta]\,\psi$ is in Γ, then $(\Gamma,[P(\vec{x}) \equiv \vartheta]_m\,\psi;\Delta)$ is in $\mathcal{P}$ for some m. If $[P(\vec{x}) \equiv \vartheta]\,\psi$ is in Δ, then $(\Gamma;\Delta,[P(\vec{x}) \equiv \vartheta]_m\,\psi)$ is in $\mathcal{P}$ for all m.

Theorem 4.2 (*ELFP* Model Existence Theorem) *If Γ_0 and Δ_0 are sets of* ELFP *sentences and $(\Gamma_0;\Delta_0)$ is an element of a consistency property $\mathcal{P}$, then there is a structure $\mathfrak{A}$ and assignment α such that $\mathfrak{A} \models \varphi[\alpha]$ for every φ in Γ and $\mathfrak{A} \not\models \psi[\alpha]$ for every ψ in Δ.*

Proof. Let $\varphi_0,\varphi_1,\varphi_2,\dots$ be a sequence of *ELFP* formulas, all of whose variables are in X, and with every *ELFP* formula occurring infinitely often in the sequence. Let $t_0,t_1,t_2,\dots$ be a sequence of all terms whose variables are all in X. Let $(\psi_0,y_0),(\psi_1,y_1),(\psi_2,y_2)\dots$ be a sequence listing all pairs where ψ_n is an atomic formula and y_n is a free variable in ψ_n.

We construct sequences $\Gamma_0 \subseteq \Gamma_1 \subseteq \Gamma_2 \subseteq \cdots$ and $\Delta_0 \subseteq \Delta_1 \subseteq \Delta_2 \subseteq \cdots$ where $(\Gamma_n;\Delta_n) \in \mathcal{P}$ and the following hold.

(i) Suppose φ_n is of the form $t = t'$. If φ_n and $\psi_m(y_m/t)$ are in Γ_n, but $\psi_m(y_m/t') \notin \Gamma_n$ for some m, then take the least such m and let $\psi_m(y_m/t') \in \Gamma_{n+1}$. If φ_n, and $\psi_m(y_m/t')$ are in Γ_n but $\psi_{m'}(y_m/t) \notin \Gamma_n$ for some m', then take the least such m' and let $\psi_m(y_{m'}/t) \in \Gamma_{n+1}$.

(ii) Suppose φ_n is of the form $\neg\psi$. If $\varphi_n \in \Gamma_n$, then $\psi \in \Delta_{n+1}$. If $\varphi_n \in \Delta_n$, then $\psi \in \Gamma_{n+1}$.

(iii) Suppose φ_n is of the form $\psi \vee \vartheta$. If $\varphi_n \in \Gamma_n$, then either $\psi \in \Gamma_{n+1}$ or $\vartheta \in \Gamma_{n+1}$. If $\varphi_n \in \Delta_n$, then $\psi \in \Delta_{n+1}$ and $\vartheta \in \Delta_{n+1}$.

(iv) Suppose φ_n is of the form $\psi \wedge \vartheta$. If $\varphi_n \in \Gamma_n$, then $\psi \in \Gamma_{n+1}$ and $\vartheta \in \Gamma_{n+1}$. If $\varphi_n \in \Delta_n$, then either $\psi \in \Delta_{n+1}$ or $\vartheta \in \Delta_{n+1}$.

(v) Suppose φ_n is of the form $\exists y\, \psi(y)$. (Here $\psi(y)$ may have free variables other than y.) If $\varphi_n \in \Gamma_n$, then $\psi(x) \in \Gamma_{n+1}$ for some $x \in X$. If $\varphi_n \in \Delta_n$ and $\psi(t_m) \notin \Delta_n$ for some m, then take the least such m and put $\psi(t_m) \in \Delta_{n+1}$.

(vi) Suppose φ_n is of the form $[P(\vec{x}) \equiv \vartheta]\, \psi$. If $\varphi_n \in \Gamma_n$, then $[P(\vec{x}) \equiv \vartheta]_m\, \psi \in \Gamma_{n+1}$ for some m. If $\varphi_n \in \Delta_n$ and $[P(\vec{x}) \equiv \vartheta]_m\, \psi \notin \Delta_n$ for some m, then take the least such m and let $[P(\vec{x}) \equiv \vartheta]_m\, \psi \in \Delta_{n+1}$.

It follows immediately from the definition of consistency property that there exist sequences $\Gamma_0 \subseteq \Gamma_1 \subseteq \Gamma_2 \subseteq \cdots$ and $\Delta_0 \subseteq \Delta_1 \subseteq \Delta_2 \subseteq \cdots$ as described above. Let $\Gamma_\omega = \bigcup_{n\in\omega} \Gamma_n$ and $\Delta_\omega = \bigcup_{n\in\omega} \Delta_n$. Note that by part (i) of the definition of consistency property, Γ_ω and Δ_ω are disjoint.

Let Γ be the set of atomic formulas in Γ_ω and $\mathfrak{T}(X)$ be the Herbrand structure described in the statement of Lemma 4.1. It follows from part (i) of the construction above that Γ is closed under substitution. We show by induction on $\lambda(\varphi)$ that if $\varphi(t_1, \ldots, t_l) \in \Gamma_\omega$, then $\mathfrak{T}(X)/\!\sim\, \models \varphi([t_1], \ldots, [t_l])$ and if $\varphi(t_1, \ldots, t_l) \in \Delta_\omega$, then $\mathfrak{T}(X)/\!\sim\, \not\models \varphi([t_1], \ldots, [t_l])$.

The basis of the induction for formulas of the form $R(t_1, \ldots, t_l)$ follows by Lemma 4.1 and the disjointness of Γ_ω and Δ_ω. Suppose that φ is of the form $t = t'$. If $\varphi \in \Gamma_omega$, then clearly $t \sim t'$ so $\mathfrak{T}(X)/\!\sim\, \models [t] = [t']$. If $t = t' \in \Delta_omega$, we wish to show that it is not the case that $t \sim t'$. This is a consequence of condition (ii) in the definition of consistency property.

The rest of the proof follows from parts (ii)–(vi) of the construction. We will consider the two most difficult cases of the induction: existential quantification and inductive definition.

Suppose $\exists y\, \psi(y, t_1, \ldots, t_l)$ is in Γ_ω. Then by part (v) of the construction, $\psi(x, t_1, \ldots, t_l)$ is in Γ_ω for some $x \in X$. By the induction hypothesis, $\mathfrak{T}(X)/\!\sim\, \models \psi([x], [t_1], \ldots, [t_l])$ and hence $\mathfrak{T}(X)/\!\sim\, \models \exists y\, \psi(y, [t_1], \ldots, [t_l])$.

Suppose $\exists y\, \psi(y, t_1, \ldots, t_l)$ is in Δ_ω. Then by part (v) of the construction, $\psi(t, t_1, \ldots, t_l)$ is in Δ_ω for every term t. By the induction hypothesis, $\mathfrak{T}(X)/\!\sim\, \not\models \psi([t], [t_1], \ldots, [t_l])$ for every term t, so $\mathfrak{T}(X)/\!\sim\, \not\models \exists y\, \psi(y, [t_1], \ldots, [t_l])$.

Suppose $[P(\vec{x}) \equiv \vartheta]\, \psi(t_1, \ldots, t_l)$ is in Γ_ω. Then by part (vi) of the construction, $[P(\vec{x}) \equiv \vartheta]_m\, \psi(t_1, \ldots, t_l)$ is in Γ_ω for some m. By the induction hypothesis, $\mathfrak{T}(X)/\!\sim\, \models [P(\vec{x}) \equiv \vartheta]_m\, \psi([t_1], \ldots, [t_l])$ and hence $\mathfrak{T}(X)/\!\sim\, \models [P(\vec{x}) \equiv \vartheta]\, \psi([t_1], \ldots, [t_l])$. Notice that it is crucial here that $\lambda([P(\vec{x}) \equiv \vartheta]_m\, \psi) < \lambda([P(\vec{x}) \equiv \vartheta]\, \psi)$.

Suppose $[P(\vec{x}) \equiv \vartheta]\, \psi(t_1, \ldots, t_l)$ is in Δ_ω. Then by part (vi) of the construction, $[P(\vec{x}) \equiv \vartheta]_m\, \psi(t_1, \ldots, t_l)$ is in Δ_ω for all m. By the induction hypothesis, $\mathfrak{T}(X)/\!\sim\, \not\models [P(\vec{x}) \equiv \vartheta]_m\, \psi([t_1], \ldots, [t_l])$ for all m and hence $\mathfrak{T}(X)/\!\sim\, \not\models [P(\vec{x}) \equiv \vartheta]\, \psi([t_1], \ldots, [t_l])$.

The other cases are similar. Thus, in $\mathfrak{T}(X)/\sim$, all sentences in Γ_ω hold and all sentences in Δ_ω fail. □

Remark. Let us continue with the notation of the preceding proof: $\Gamma_0 \nvdash \Delta_0$ and Γ is the set of atomic formulas in Γ_ω. Also, let Δ be the set of atomic formulas in Δ_ω. The crux of the proof was to construct a model in which all formulas in Γ hold and all formulas in Δ fail. $\mathfrak{T}(X)/\sim$ is one such model, but there may be others. An alternative construction is as follows. Let B' be the set of subterms appearing in formulas in Δ. Let B be $B'/\sim$ (i.e., restrict $\sim$ to B' and consider equivalence classes). Now B may not be the universe of a structure (when we take the natural interpretations of function and relation symbols) because some functions may not be defined everywhere. Therefore, take the structure $\mathfrak{A}$ with universe $B \cup \{a\}$, where a interprets all terms not in B'. It is easy to see that in $\mathfrak{A}$ the formulas in Γ hold and the formulas in Δ fail. An induction on formulas shows that in $\mathfrak{A}$, all sentences in Γ_ω hold and all sentences in Δ_ω fail.

Consequently, if Δ is finite, then there is a finite model of Γ_0 in which some sentence of Δ_0 fails. Let us examine some sufficient conditions for Δ to be finite.

Assume that the vocabulary V is finite and that Γ_0 and Δ_0 are finite. If the sentences in Δ_0 do not contain quantifiers or inductive definitions, then it is easy to verify that Δ is finite. The only parts of the construction that can add infinitely many formulas to Δ_ω are parts (v) and (vi), but they are not used when Δ_0 contains no quantifiers or inductive definitions.

Again assume that the vocabulary V is finite and that Γ_0 and Δ_0 are finite. Assume also that V contains no function symbols (but constant symbols are allowed). Since Γ_0 is finite, we can take X to be finite: in the construction, the only role for X was to provide witnesses for existential quantifiers in subformulas occurring in Γ_0. There are only finitely many atomic formulas with variables in X, so Δ is finite. Thus, we have the following theorem.

Theorem 4.3 *Let Γ_0 and Δ_0 be finite sets of* ELFP *sentences over a finite vocabulary and let $(\Gamma_0; \Delta_0)$ be an element of a consistency property $\mathcal{P}$. Suppose either that the sentences in Δ_0 contain no quantifiers or inductive definitions, or that V contains no function symbols. Then there is a finite structure $\mathfrak{A}$ and assignment α such that $\mathfrak{A} \models \varphi[\alpha]$ for every φ in Γ and $\mathfrak{A} \not\models \psi[\alpha]$ for every ψ in Δ.*

Now we state the main theorem.

Theorem 4.4 (Completeness Theorem for LE) *If $\Gamma \models \Delta$ in* ELFP, *then $\Gamma \vdash \Delta$ is a theorem of* **LE**.

Proof. We prove the contrapositive of the theorem: assuming that $\Gamma \nvdash \Delta$ in **LE** we construct a model of Γ in which Δ fails. Let $\mathcal{P}$ be the set of all pairs $(\Gamma'; \Delta')$ such that $\Gamma' \nvdash \Delta'$. It suffices by the *ELFP* Model Existence Theorem to show that $\mathcal{P}$ is a consistency property.

Suppose $(\Gamma'; \Delta')$ is in $\mathcal{P}$.

Γ' and Δ' are disjoint. If they had an element φ in common, we could apply the rules $(* \vdash)$ and $(\vdash *)$ to the axiom $\varphi \vdash \varphi$ to show $\Gamma' \vdash \Delta'$.

If Γ' contains formulas $t_1 = t'_1, t_2 = t'_2, \ldots, t_k = t'_k$, where $k \geq 0$, then Δ' contains no formulas of the form $t(x_1/t_1, \ldots, x_k/t_k) = t(x_1/t'_1, \ldots, x_k/t'_k)$. Otherwise, we could apply $(* \vdash)$ to the axiom $\emptyset \vdash t = t$ to obtain $\Gamma' \vdash t = t$. Then apply $(\vdash S)$ to obtain $\Gamma' \vdash t(x_1/t_1, \ldots, x_k/t_k) = t(x_1/t'_1, \ldots, x_k/t'_k)$. Finally, apply $(\vdash *)$ to obtain $\Gamma' \vdash \Delta'$.

If $t \doteq t'$ and $\psi(x/t)$ are in Γ', where ψ is an atomic formula, then $(\Gamma', \psi(x/t'); \Delta')$ is in $\mathcal{P}$. This follows by $(S \vdash)$.

The other conditions in the definition of consistency property follow directly by similar arguments. □

We conclude this section by sketching the modifications needed to make the proof of Theorem 4.4 work for uncountable sets of formulas. As before, we assume that $\Gamma \not\vdash \Delta$ and consider the set $\mathcal{P}$ of all pairs $(\Gamma'; \Delta')$ such that $\Gamma' \not\vdash \Delta'$. Now follow the construction in the *ELFP* Model Existence Theorem: specify a sequence of *ELFP* formulas with every formula occurring infinitely often in the sequence, a sequence of terms whose free variables are all in X, and a sequence of pairs of atomic formulas and their free variables. Since the vocabulary is no longer assumed to be countable, these sequences may not have order type ω. Thus, it becomes necessary to specify what happens at limit ordinals. We clearly want to take unions at the limit ordinals (as we did to form Γ_ω and Δ_ω in the proof of the *ELFP* Model Existence Theorem). The problem then is to show that we remain in $\mathcal{P}$ when we do this. That is, whenever β is a limit ordinal it should be the case that $\Gamma_\beta \not\vdash \Delta_\beta$. This would be immediate if *ELFP* were compact, but since it is not, we must be more devious. We show that if β is a limit ordinal then $\Gamma_\beta \not\vdash \Delta_\beta$ implies $\Gamma_{\beta+\omega} \not\vdash \Delta_{\beta+\omega}$. Assume the contrary, so that Υ is an **LE** proof of $\Gamma_{\beta+\omega} \vdash \Delta_{\beta+\omega}$. We describe intuitively how to modify Υ to give an **LE** proof of $\Gamma_\beta \vdash \Delta_\beta$, thereby obtaining a contradiction. For each occurrence of a formula φ in $(\Gamma_{\beta+\omega} - \Gamma_\beta) \cup (\Delta_{\beta+\omega} - \Delta_\beta)$ and each branch of Υ, there is node closest to the root where φ is "introduced". The idea is to modify Υ at all such nodes so that φ is eliminated. This is easy if φ is introduced using one of the weakening rules $(* \vdash)$ or $(\vdash *)$: just refrain from introducing φ at this point. If φ is introduced by one of the other rules, we must examine items (i)–(vi) in the proof of the *ELFP* Model Existence Theorem to see that φ may be replaced (possibly using the rules of inference several times) with either a formula of Γ_β in the left part of the sequent or a formula of Δ_β in the right part of the sequent. It is not difficult now to make this idea rigorous.

5 Compactness and Finite Model Properties.

In general, the compactness and the finite model properties are not true in *ELFP*. However, under certain conditions these properties hold. In this section, we examine some of these conditions.

One way to state the compactness property of first-order logic is to say that every inconsistent theory has a finite inconsistent subtheory. That is, if $\Gamma \models \emptyset$ then there is a finite $\Gamma' \subseteq \Gamma$ such that $\Gamma' \models \emptyset$. This holds for *ELFP* in the case where the vocabulary contains no function symbols (but constant symbols are allowed). A more general form of compactness says that if $\Gamma \models \Delta$, then there are finite subsets $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ such that $\Gamma' \models \Delta'$. For first-order logic, this is equivalent to the more restricted form of compactness, but since *ELFP* is not closed under negation, it fails for *ELFP* even when V contains no function or constant symbols. This form of compactness does hold however when Γ is also first-order (i.e., none of its sentences contain inductive definitions). It is true that *ELFP* is *countably complete*: if $\Gamma \models \Delta$, then there are countable subsets $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ such that $\Gamma' \models \Delta'$. Let us prove these assertions.

Theorem 5.1 ELFP *is countably compact.*

Proof. By the Completeness Theorem for **LE** it is enough to show that if $\Gamma \vdash \Delta$ is a theorem of **LE**, then there are countable sets $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ such that $\Gamma' \vdash \Delta'$ is also a theorem of **LE**.

For **LE** proofs Υ_1 and Υ_2, write $\Upsilon_1 \prec \Upsilon_2$ if Υ_1 is a proper subproof of Υ_2. That is, Υ_2 is the labeled subtree of Υ_1 whose nodes are the descendents of some given node in Υ_2. The relation $\prec$ is well founded, i.e., there are no infinite descending chains. This is obvious since all branches of an **LE** proof are finite.

Thus, we have the following form of induction on proofs: To prove that a statement holds of all proofs Υ, it is enough to show that for every proof Υ, if the statement holds of all proofs $\Upsilon' \prec \Upsilon$, then it holds of Υ. It is an easy matter to show that for every proof Υ, if the root of Υ is labeled by $\Gamma \vdash \Delta$, then there are countable sets $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ such that $\Gamma' \vdash \Delta'$ is the label on the root of some proof Υ'. This is clear if $\Gamma \vdash \Delta$ is an axiom. Otherwise, $\Gamma \vdash \Delta$ follows from the labels of the children of the root of Υ according to one of the rules of inference. We will show how this works in the case of the rule $([\,]\vdash)$, which is the most interesting case. All the other cases follow by the same sort of argument.

Suppose that the root of Υ is labeled $\Gamma, [P(\vec{x}) \equiv \vartheta]\,\psi \vdash \Delta$ and its children are labeled $\Gamma, [P(\vec{x}) \equiv \vartheta]_m\,\psi \vdash \Delta$ for each $m \in \omega$. By the induction hypothesis there are countable sets $\Gamma'_m \subseteq \Gamma$ and $\Delta'_m \subseteq \Delta$ such that $\Gamma'_m, [P(\vec{x}) \equiv \vartheta]_m\,\psi \vdash \Delta'_m$ is label on the root of some **LE** proof for each $m \in \omega$. (It may be that for some m there are countable sets $\Gamma'_m \subseteq \Gamma$ and $\Delta'_m \subseteq \Delta$ such that $\Gamma'_m \vdash \Delta'_m$ is label on the root of some **LE** proof; if this occurs, we are done, so we assume otherwise.) Let Γ' be the union of the sets Γ'_m and Δ' be the union of the sets Δ'_m. Then by $(*\vdash)$ and $(\vdash *)$, $\Gamma', [P(\vec{x}) \equiv \vartheta]_m\,\psi \vdash \Delta'$ is label on the root of some **LE** proof for each $m \in \omega$ and hence $\Gamma', [P(\vec{x}) \equiv \vartheta]\,\psi \vdash \Delta'$ is a theorem of **LE**. □

Since **LE** has no cut rule, the only use of the infinitary rule $([\,]\vdash)$ is to introduce formulas with inductive definitions on the left side of a sequent. The following proposition follows immediately from this observation.

Proposition 5.2 *Let Γ be a set of existential first-order formulas and Δ be a set of* ELFP *formulas. If $\Gamma \vdash \Delta$ is a theorem of* LE *then it has a finite proof in* **LE**.

As a consequence we have the following.

Theorem 5.3 *Let Γ be a set of existential first-order formulas and Δ be a set of* ELFP *formulas. If $\Gamma \models \Delta$, then there are finite sets $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ such that $\Gamma' \models \Delta'$.*

Proof. The previous proposition asserts that $\Gamma \vdash \Delta$ has a finite **LE** proof. The argument in the proof of Theorem 5.1 now produces finite sets $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ such that $\Gamma' \models \Delta'$. □

Remark. Theorem 5.3 can be proved without recourse to the Completeness Theorem for **LE**. Let $\Delta^\neg = \{\neg\varphi : \varphi \in \Delta\}$. Of course, $\Delta^\neg$ will not necessarily be a set of *ELFP* sentences, but this is not a problem. We regard it as a set of second-order sentences. Clearly, $\Gamma, \Delta^\neg \models \emptyset$. We will be done if we can show that there is a finite set $\Gamma' \subseteq \Gamma \cup \Delta^\neg$ such that $\Gamma' \models \emptyset$. In other words we must show that $\Gamma \cup \Delta^\neg$ is consistent if every finite subset is consistent. Blass and Gurevich [3] show that *ELFP* sentences are equivalent to Π^1_1 sentences (in fact, they are equivalent to sentences in a subclass of Π^1_1 called strict-Π^1_1). Thus, the sentences in $\Delta^\neg$ (and hence in $\Gamma \cup \Delta^\neg$)

are equivalent to Σ^1_1 sentences. But Σ^1_1 sentences satisfy the compactness property. This follows by observing that ultraproducts preserve Σ^1_1 sentences (Theorem 4.1.14 of Chang and Keisler [5]) so the ultraproduct proof of the Compactness Theorem (Theorem 4.1.11 of Chang and Keisler) works for Σ^1_1 sentences.

Theorem 5.4 *Let* Γ *be a set of* ELFP *formulas over a vocabulary with no function symbols. If every finite subset of* Γ *is consistent, then so is* Γ.

Proof. By Theorem 5.1 we may assume that Γ is countable. Let $\Gamma = \{\varphi_0, \varphi_1, \varphi_2, \ldots\}$, $\Gamma_n = \{\varphi_0, \varphi_1, \varphi_2, \ldots, \varphi_{n-1}\}$, and V_n be the set of relation and constant symbols occurring in Γ_n. By assumption, the sets Γ_n are consistent, so for each n, there is a V_n-structure $\mathfrak{A}_n$ which is a model of Γ_n.

Let $\mathfrak{A}$ be a structure over some vocabulary containing V_n. By $h_n(\mathfrak{A})$ we will mean the V_n-structure formed by taking the reduction of $\mathfrak{A}$ to the vocabulary V_n and then taking the submodel whose universe consists of interpretations of constant symbols in V_n. (If V_n has no constant symbols, $h_n(\mathfrak{A})$ is empty; this is the only place in the paper where empty structures are allowed.)

Observe that for each n there are only finitely many structures $h_n(\mathfrak{A}_k)$, where $k \geq n$, and that each one of these structures can be extended to a model of Γ_n. Form a tree whose nodes at the n-th level are the structures $h_n(\mathfrak{A}_k)$ and where $h_n(\mathfrak{A}_k)$ is an ancestor of $h_{n'}(\mathfrak{A}_{k'})$ if $n \leq n'$ and $h_n(\mathfrak{A}_k) = h_n(h_{n'}(\mathfrak{A}_{k'}))$. This is a finitely branching infinite tree so by König's Lemma it has an infinite branch. Let $\mathfrak{A}_\omega$ be the V-structure which is the direct limit (defined in the obvious way) along some infinite branch of the tree. For each n, $h_n(\mathfrak{A}_\omega)$ can be extended to a model $\mathfrak{B}_n$ of Γ_n. Let A_ω be the universe of $\mathfrak{A}_\omega$ and B_n be the universe of $\mathfrak{B}_n$. We may assume the sets $B_n - A_\omega$ are disjoint. Let $\mathfrak{B}_\omega = \bigcup_{n \in \omega} \mathfrak{B}_n$. Then for each n, $\mathfrak{B}$ has a substructure, viz. $\mathfrak{B}_n$, which is a model of Γ_n. Since *ELFP* formulas are preserved by extensions (see Blass and Gurevich [3]) $\mathfrak{B} \models \Gamma_n$ for each n. Therefore $\mathfrak{B} \models \Gamma$. □

With regard to the previous theorem, it would be interesting to have a construction which, from an **LE** proof of $\Gamma \vdash \emptyset$, produces a finite subset $\Gamma' \subseteq \Gamma$ and an **LE** proof of $\Gamma' \vdash \emptyset$.

Let us now show that compactness fails in general for *ELFP*.

Proposition 5.5 ELFP *is not compact.*

Proof. We will give several examples of sets of *ELFP* sentences Γ and Δ where $\Gamma \vdash \Delta$ but there are no finite sets $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ where $\Gamma' \vdash \Delta'$. These examples show that the only circumstances where compactness holds in general are given by Theorems 5.3 and 5.4.

The first example is suggested immediately by Theorem 2.1(iii). The only element of Γ is the sentence

$$[P(x,y) \equiv E(x,y) \vee \exists z\,(P(x,z) \wedge E(z,y))]\, P(c,d).$$

The elements of Δ are the sentences

$$[P(x,y) \equiv E(x,y) \vee \exists z\,(P(x,z) \wedge E(z,y))]_m\, P(c,d).$$

Clearly, $\Gamma \models \Delta$ but there is no finite $\Delta' \subseteq \Delta$ such that $\Gamma \models \Delta'$. In this example we have not used function symbols or equality.

In the next example Γ contains the sentences $\neg f(d) = d$, $f(f(d)) = f(d)$, and

$$[P(x,y) \equiv f^m(x) = y \vee \exists z\,(f(x) = z \wedge P(z,y))]\, P(c,d).$$

Clearly, $\Gamma \models \emptyset$ but there is no finite $\Gamma' \subseteq \Gamma$ such that $\Gamma' \models \emptyset$.

Notice that this example would work just as well if f were a partial function. In Γ replace the formula $f(x) = y$ with $R(x,y)$ and replace the formulas $f^m(x) = y$ and $f^m(x) = f(y)$ with formulas that use R instead of f. (It is necessary to use existential quantifiers here.) Let Δ contain the sentence $\exists x, y, z\,(R(x,y) \wedge R(x,z) \wedge \neg y = z)$. This is equivalent to the negation of a sentence asserting that $R(x,y)$ represents a partial unary function. Thus, $\Gamma \models \Delta$, but there is no finite $\Gamma' \subseteq \Gamma$ such that $\Gamma' \models \Delta$. In this example the vocabulary contains no function symbols. With a little more work we could replace equality with an equivalence relation. $\square$

Blass and Gurevich proved that *ELFP* has the finite model property: every finite consistent set of *ELFP* sentences has a finite model. We are interested in a generalization of this property. Given finite sets Γ and Δ such that $\Gamma \not\models \Delta$, is there a finite model of Γ in which some sentence in Δ fails? This property does not hold in general, even for the set of existential first-order sentences. For example, let $\Gamma = \emptyset$ and Δ contain existential sentences that say f is not one-to-one and there is an element mapped onto c by f. However, there are two cases where this generalized finite model property holds. The first is just a modest generalization of the result of Blass and Gurevich.

Theorem 5.6 *Let Γ and Δ be finite sets of* ELFP *sentences such that $\Gamma \not\models \Delta$. Suppose either that Δ is a set of quantifier free sentences, or that the vocabulary contains no function symbols. Then Γ has a finite model in which some sentence in Δ fails.*

This follows by Theorem 4.3.

We conclude the section with two immediate corollaries of this theorem.

Corollary 5.7 *Let Γ and Δ be finite sets of* ELFP *sentences such that either Δ is a set of quantifier free sentences or the vocabulary contains no function symbols. Then every finite model of Γ satisfies some sentence in Δ if and only if $\Gamma \vdash \Delta$.*

Corollary 5.8 *Fix a vocabulary with no function symbols. It is decidable whether $\Gamma \models \Delta$ when Γ is a finite set of existential first-order sentences, and Δ is a finite set of* ELFP *sentences.*

Proof. If $\Gamma \models \Delta$, there is a finite proof of $\Gamma \vdash \Delta$ in **LE** because the rule $([\,]\vdash]$ is never used. If $\Gamma \not\models \Delta$ there is a finite model of Γ in which some sentence in Δ fails. The search for the finite proof and the finite model can be done in tandem. $\square$

6 Conclusions.

We close with a brief discussion of the relationship between the model theoretic results presented in this paper, and logic programming and query languages.

From our discussion in the introduction, it is easy to see that we can translate any pure Prolog program into an *ELFP* formula $\varphi(\vec{y})$ of the form

$$[P_1(\vec{x}_1) \equiv \vartheta_1; \cdots; P_k(\vec{x}_k) \equiv \vartheta_k]\, \psi(\vec{y}),$$

where the simultaneous inductive definition is derived from the program clauses and $\psi(\vec{y})$ corresponds to the goal. (See Apt [2] for terminology.) Program execution is intimately connected to proving the sequent $\emptyset \vdash \exists \vec{y} \varphi(\vec{y})$. Since **LE** has no cut rule, a sequent of this form must be derived using the rule $(\vdash \exists)$. Thus, we must be able to prove a sequent of the form $\emptyset \vdash \varphi(\vec{t})$. A Prolog interpreter finds the term sequences $\vec{t}$ using SLD resolution. Proposition 5.8 is important here because we should not need the infinitary rule $([\,]\vdash)$ in this circumstance. (A Prolog interpreter may fail to find a term sequence $\vec{t}$ because of the search strategy used by its theorem prover, but this is a different issue.) Proposition 5.8 applies more generally to sequents of the form $\Gamma \vdash \varphi$, where Γ is a set of existential first-order sentences. It is not clear what sort of computation corresponds to proofs of such sequents.

Our results also show that pure Prolog is logically different from Datalog and Datalog$^\neg$, which have received much attention as database query languages. From a logical point of view, Datalog$^\neg$ is pure Prolog with no function symbols. (Datalog is a further restriction with no negations). In practice, Datalog$^\neg$ is used to make queries on finite structures (relational databases), not to perform computations on possibly infinite structures as pure Prolog does (see [10]). The results of the last section show that there are good reasons, apart from the tradition of relational databases, to prohibit function symbols when working on finite structures. Corollary 5.7 shows that in the absence of function symbols, **LE** is sound and complete when restricted to finite structures. Moreover, Theorem 5.3 shows that a form of compactness holds. Finally, Corollary 5.8 shows that if a database is not given explicitly, but instead a set of existential first-order sentences holding in the database are presented, queries are still decidable.

We hope that model theoretic foundations of *ELFP* presented here will suggest interesting research directions in the many areas of computer science where it arises.

References

[1] P. ACZEL, *Introduction to inductive definitions*, in Handbook of Mathematical Logic, J. Barwise, ed., North-Holland, Amsterdam, 1977, pp. 739–782.

[2] K. R. APT, *Logic programming*, in Handbook of Theoretical Computer Science, J. van Leeuwen, ed., vol. B, North-Holland, Amsterdam, 1990, pp. 493–574.

[3] A. BLASS AND Y. GUREVICH, *Existential fixed-point logic*, in Computation Theory and Logic, E. Börger, ed., vol. 270 of Lecture Notes in Computer Science, New York, 1987, Springer-Verlag, pp. 20–36.

[4] A. CHANDRA AND D. HAREL, *Horn clause queries and generalizations*, J. Logic Programming, 1 (1985), pp. 1–15.

[5] C. C. CHANG AND H. J. KEISLER, *Model Theory*, North-Holland, Amsterdam, third ed., 1990.

[6] K. J. COMPTON, *Stratified least fixpoint logic*. Submitted.

[7] W. P. DE ROEVER, *Recursive Program Schemes: Semantics and Proof Theory*, vol. 70 of Mathematical Centre Tracts, Mathematische Centrum, Amsterdam, 1976.

[8] M. FITTING, *First-Order Logic and Automated Theorem Proving*, Springer-Verlag, New York, 1990.

[9] N. IMMERMAN, *Relational queries computable in polynomial time*, Inform. and Control, 68 (1986), pp. 86–104.

[10] P. C. KANELLAKIS, *Elements of relational database theory*, in Handbook of Theoretical Computer Science, J. van Leeuwen, ed., vol. B, North-Holland, Amsterdam, 1990, pp. 1073–1156.

[11] C. KARP, *Languages with Expressions of Infinite Length*, North-Holland, Amsterdam, 1964.

[12] H. J. KEISLER, *Model Theory for Infinitary Logic*, North-Holland, Amsterdam, 1971.

[13] J.-L. LASSEZ, V. NGUYEN, AND E. SONENBERG, *Fixed point theorems and semantics: A folk tale*, Information Processing Lett., 14 (1982), pp. 112–116.

[14] J. LOECKX AND K. SIEBER, *Foundations of Program Verification*, Wiley, New York, 1984.

[15] E. G. K. LOPEZ-ESCOBAR, *An interpolation theorem for denumerably long sentences*, Fund. Math., 57 (1965), pp. 253–272.

[16] Y. N. MOSCHOVAKIS, *Elementary Induction on Abstract Structures*, North-Holland, Amsterdam, 1974.

[17] D. PARK, *Finiteness is μ-ineffable*, Theoret. Comput. Sci., 3 (1976), pp. 173–181.

[18] W. W. TAIT, *Normal derivability in classical logic*, in Syntax and Semantics of Infinitary Languages, J. Barwise, ed., vol. 72 of Lecture Notes in Mathematics, Berlin, 1968, Springer-Verlag, pp. 204–236.

[19] G. TAKEUTI, *Proof Theory*, North-Holland, Amsterdam, second ed., 1987.

[20] M. VARDI, *Complexity of relational query languages*, in Proc. 14th ACM Symp. on Theory of Computing, New York, 1982, Association for Computing Machinery, pp. 137–146.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399