EXISTENTIAL FIXED-POINT LOGIC

Andreas Blass and Yuri Gurevich

Mathematics Department and Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109, U.S.A.

Abstract The purpose of this paper is to draw attention to existential fixed-point logic. Among other things, we show that: (1) If a structure $\mathcal{A}$ satisfies an existential fixed-point formula φ, then $\mathcal{A}$ has a finite subset F such that every structure $\mathcal{B}$ with $\mathcal{A}|F = \mathcal{B}|F$ satisfies φ. (2) Using existential fixed-point logic instead of first-order logic removes the expressivity hypothesis in Cook's completeness theorem for Hoare logic. (3) In the presence of a successor relation, existential fixed-point logic captures polynomial time.

Introduction

The purpose of this paper is to draw attention to a logic, a fragment of second-order logic, that enjoys a particularly close connection with certain aspects of the theory of computation. This logic, which we call *existential fixed-point logic,* is stronger than first-order logic in some ways but weaker in other ways. It goes beyond first-order logic in that it has the "fixed point" operator. On the other hand, it has only the existential quantifier, not the universal one, and it restricts the use of propositional connectives in a way that precludes defining ∀ from ∃.

After defining existential fixed-point logic and some related logics in Section 1, we devote the next two sections to the connections between this logic and Hoare's logic of asserted programs and polynomial time computability. In particular, we show in Section 2 that the use of existential fixed-point logic instead of first-order logic removes the need for an expressivity hypothesis in Cook's completeness theorem for Hoare logic. In Section 3, we show that, when we consider only finite structures equipped with a successor relation, then existential fixed-point logic captures polynomial time computability. Then we present a model-theoretic property of existential fixed-point logic that helps to clarify its relationship with first-order logic. In the final section, we show that the validity problem and the satisfiability problem for existential fixed-point logic are both complete recursively enumerable problems. Along the way to these results about the decision problem, we also find some more model-theoretic properties of the logic, including that, whenever an existential fixed-point formula is true in a structure, then its truth depends on only a finite subset of the structure and that the iterations that define the fixed points always terminate at or before the first transfinite ordinal number.

In our discussion of Hoare logic in Section 2, we introduce the profile of a program. Though closely related to the strongest postcondition and weakest precondition commonly used in connection with Hoare logic, profiles seem more intuitive and easier to use, partly because they are associated simply with a program (rather than a program and a given precondition or postcondition) and partly because they do not require the use of the same variables in connection with the program, its input, and its output. We believe that profiles will find further uses in the logic of programs.

Because of constraints of time and space, this paper does not even approach completeness. We hope to convince the reader that existential fixed-point logic is an interesting and useful logic, and we plan to return to it in future publications.

1. Definitions and conventions

As indicated in the introduction, existential fixed-point logic differs from first-order logic in two respects, the absence of the universal quantifier and the presence of the fixed-point operator. Both of these deserve some clarification.

Mere removal of the universal quantifier ($\forall$) has no real effect on first-order logic, since $\forall x\varphi$ can be expressed in terms of negation and existential quantification as $\neg\exists x\neg\varphi$. To correctly define the existential fragment **E** of first-order logic, one must prevent such surreptitious reintroduction of the universal quantifier, for example by insisting that all formulas have prenex forms with no universal quantifiers. For our purposes, however, the convenient way to formalize **E** is to allow as prepositional connectives only conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$) and to apply negation only to atomic formulas. Every quantifier-free formula is equivalent to one satisfying these restrictions, but the restrictions ensure that all quantifiers in our formulas occur only positively; in particular, existential quantifiers will remain existential when moved to prenex form .

We extend first-order logic by adding the *iterative fixed point* construction, defined as follows. Let S be any set, $\mathcal{P}(S)$ its power set, and Γ an operator on $\mathcal{P}(S)$, i.e., a function from $\mathcal{P}(S)$ to itself. Define a (possibly transfinite) sequence of subsets $\Gamma^\alpha \subseteq S$ by the recursion

$$\Gamma^0 = \emptyset,$$

$$\Gamma^{\alpha+1} = \Gamma^\alpha \cup \Gamma(\Gamma^\alpha),$$

$$\Gamma^\lambda = \bigcup_{\alpha<\lambda} \Gamma^\alpha \qquad \text{for limit ordinals } \lambda.$$

The sequence Γ^α is obviously weakly increasing with respect to C, so (as S is a set and the ordinals form a proper class) it must be constant from some point on. The common value of Γ^α for all sufficiently large α is written Γ^∞ and called the iterative fixed point of Γ. In general, Γ^∞ need not be fixed by Γ; we have only $\Gamma(\Gamma^\infty) \subseteq \Gamma^\infty$. However, if Γ is either inflationary ($X \subseteq \Gamma(X)$ for all $X \subseteq S$) or monotone ($\Gamma(X) \subseteq \Gamma(Y)$ whenever $X \subseteq Y \subseteq S$), then $\Gamma^{\alpha+1} = \Gamma(\Gamma^\alpha)$ for all α (by induction on α in the monotone case, and trivially in the inflationary case), so $\Gamma^\infty = \Gamma(\Gamma^\infty)$; hence the terminology "fixed point".

For monotone Γ, the iterative fixed point is also the *least fixed point*, a subset of every $X \subseteq S$ such that $\Gamma(X) = X$. Indeed, an equivalent (non-inductive) definition of Γ^∞ is that it is the intersection of all $X \subseteq S$ such that $\Gamma(X) \subseteq X$.

Let σ be a signature (sometimes called a first-order language), P an r-place predicate symbol not in σ, $\mathbf{x}$ an r-tuple of distinct variables, and δ a formula of the signature $\sigma \cup \{P\}$. Let $\mathcal{A}$ be any structure of signature σ, and let all free variables of δ other than $\mathbf{x}$ be assigned specific values in $\mathcal{A}$. Then δ defines an operator Δ on the set $P(\mathcal{A}^r)$ of r-place relations on $\mathcal{A}$ by

$$\Delta(X) = \{a \mid (\mathcal{A}, X) \vDash \delta[\mathbf{a/x}]\};$$

here $(\mathcal{A}, X)$ is the $\sigma \cup \{P\}$-structure obtained from $\mathcal{A}$ by interpreting P as X, and the notation $[\mathbf{a/x}]$ means that the variables $\mathbf{x}$ are assigned the values $\mathbf{a}$. Let Δ^∞ be the iterative fixed point of this operator Δ. Any $\sigma \cup \{P\}$-formula φ can be interpreted in $\mathcal{A}$ by letting P denote Δ^∞; we write this interpretation as

$$\text{LET } P(x) \leftarrow \delta \text{ THEN } \varphi .$$

In other words, this new formula holds in $\mathcal{A}$ if and only if φ holds in $(\mathcal{A}, \Delta^\infty)$. More generally, we shall permit simultaneous inductive definitions of several relations. The general form of the LET. ..THEN... construction is

(*) $\qquad \text{LET } P_1(\mathbf{x}^1) \leftarrow \delta_1, \ldots, P_n(\mathbf{x}^n) \leftarrow \delta_n \text{ THEN } \varphi,$

where $\delta_1, \ldots, \delta_n$, and φ are $\sigma \cup \{P_1, \ldots, P_n\}$-formulas; $P_1, \ldots, P_n$ are distinct predicate symbols not in σ; and each $\mathbf{x}^i$ is a list of distinct variables of the right length to serve as arguments of P_i. A variable is free in (*) if and only if either it is free in φ or, for some i, it is free in δ_i and not in the list $\mathbf{x}^i$. The semantics of (*) is defined as follows. Let a σ-structure $\mathcal{A}$ and values for the free variables of (*) be given. Define transfinite sequences Δ_i^α for $1 \leq i \leq n$ by

$$\Delta_i^0 = \varnothing;$$

$$\Delta_i^{\alpha+1} = \Delta_i^\alpha \cup \{ \mathbf{a} \mid (\mathcal{A}, \Delta_1^\alpha, \ldots, \Delta_n^\alpha) \vDash \delta_i [\mathbf{a}/\mathbf{x}^{\mathbf{i}}]\},$$

|

$$\Delta_i^\lambda = \bigcup_{\alpha<\lambda} \Delta_i^\alpha \quad \text{for limit ordinals } \lambda.$$

For sufficiently large α, Δ_i^α is independent of α, and we call it Δ_i^∞. Then (*) is defined to be true in $\mathcal{A}$ if and only if φ is true in $(\mathcal{A}, \Delta_1^\infty, \ldots, \Delta_n^\infty)$.

We use the notations **FO**, **E**, **U**, and **QF** for first-order logic and its existential, universal, and quantifier-free fragments, respectively. We write **FO+IFP** for the logic obtained by adding the LET. . .THEN. . . construction to **FO**; similarly for **QF+IFP**.

We could similarly define **E+IFP** or **U+IFP**, but this would make little sense, since it would surreptitiously introduce the "unwanted" quantifiers. For example, LET $P(x) \leftarrow \exists y\, \neg Q(x,y)$ THEN $\neg P(x)$ is equivalent to $\forall y\, Q(x,y)$. In connection with **E** and **U**, therefore, it is reasonable to require the formulas $\delta_1, \ldots, \delta_n$, and φ in (*) to contain $P_1, \ldots, P_n$ only positively. We shall verify below that this requirement makes the operator Δ on *n*-tuples of relations monotone, so the Δ_i^∞ constitute the *least fixed point* of Δ. Accordingly, we refer to this restricted version of **IFP** as **LFP**.

Since we shall need to keep track of positivity requirements for various predicate symbols, it will be convenient to build such requirements into the signatures. Thus, we adopt the convention that a signature consists not only of a set of predicate symbols and a set of function symbols, each set being equipped with a number-of-places function, but also of a partition of the set of predicate symbols (including equality) into two subsets, the *negatable* and the *positive* predicate symbols. With this convention, the various logics described above are formally defined as follows. The **QF** formulas of signature *a* are given by the inductive clauses

(Atomic) Every atomic formula is a formula.

(¬) If φ is an atomic formula whose predicate is negatable,
then ¬φ is a formula.

(∧, ∨) If φ and ψ are formulas, then so are (φ ∧ ψ) and (φ ∨ ψ).

The formulas of **E** are produced by adding the clause

(∃) If φ is a formula and *x* is any variable,

then ∃*x*φ is a formula.

The formulas of **U** are produced by adding to the three clauses defining **QF** the clause

($\forall$) If φ is a formula and x is any variable,

then $\forall x\varphi$ is a formula.

All five of these clauses together produce the formulas of **FO.** In each case, **+IFP** indicates addition of the clause

(IFP) If $\sigma' = \sigma \cup \{P_1, \ldots, P_n\}$,
where the P_i are distinct negatable predicate symbols not in σ,
and each P_i has r_i argument places,
if $\mathbf{x}^{\mathbf{i}}$ is an r_i-tuple of distinct variables ($1 \leq i \leq n$),
and if $\delta_1, \ldots, \delta_n$, and φ are σ'-formulas,
then

$$\text{LET } P_1(\mathbf{x}^1) \leftarrow \delta_1, \ldots, P_n(\mathbf{x}^n) \leftarrow \delta_n \text{ THEN } \varphi$$

is a σ-formula.

(Note that the definition of formulas is done for all signatures simultaneously.) Finally, **+LFP** indicates the addition of the clause (LFP) that is exactly like (IFP) except for having "positive" in place of "negatable" .

2. Preconditions, postconditions, and profiles

In this section, we present the line of thought concerning Hoare's logic of asserted programs (Hoare 1969) that first led us to investigate existential fixed-point logic, **E+LFP.** We assume familiarity with Hoare's logic for while-programs with recursively defined procedures, as presented in (Apt 1981). We use the partial correctness interpretation of asserted programs unless the contrary is explicitly stated. We consider any procedure declarations used in a program to be part of the program; in particular, when we refer to the variables in a program, this includes the variables used in these declarations.

We believe that the need for an expressivity hypothesis in Cook's completeness theorem indicates that first-order logic is not the best logic for dealing with programs. We set out to find a reasonable logic in which the strongest postconditions (or weakest preconditions) in Hoare's logic could be expressed naturally, without special hypotheses.

Instead of working directly with preconditions or postconditions, we find it more natural to work with what we call the *profile* of a program. This is essentially a description of the program's input-output behavior. More precisely, let S be a program, let $\mathbf{v}$ be a list of distinct variables that contains (at least) all the variables in S, and let $\mathbf{x}$ and $\mathbf{y}$ be two lists of variables that are all distinct, each of the same length as $\mathbf{v}$. Then by the *profile* of S with respect to the lists of variables $\mathbf{v}, \mathbf{x}, \mathbf{y}$, we mean the set of all states μ such that, if S is started in a state where the values of its variables $\mathbf{v}$ are $\mu(\mathbf{x})$, then the run of S terminates, and the final values of $\mathbf{v}$ are $\mu(\mathbf{y})$. If this profile is expressible by a formula, i.e., if there is a formula satisfied by exactly those states μ that belong to the profile, then we write $\mathit{Profile}(S,\mathbf{v};\, \mathbf{x},\mathbf{y})$ for such a formula. Notice that every occurrence of a variable in S or $\mathbf{v}$ is bound, and every occurrence of a variable in $\mathbf{x}$ or $\mathbf{y}$ is free, in $\mathit{Profile}(S,\mathbf{v};\, \mathbf{x},\mathbf{y})$. Profiles are related to strongest postconditions (for which we use the notation $\mathit{Post}(\{\varphi\}, S)$) by the equivalences

$$\mathit{Profile}(S,\mathbf{v};\, \mathbf{x},\mathbf{y}) \Leftrightarrow \mathit{Post}(\{\mathbf{x} = \mathbf{y}\}, S[\mathbf{y}/\mathbf{v}]),$$

where $[\mathbf{y}/\mathbf{v}]$ means substitution of $\mathbf{y}$ for $\mathbf{v}$, and

$$\mathit{Post}(\{\varphi\}, S) \Leftrightarrow \exists \mathbf{x}\, (\varphi\, [\mathbf{x}/\mathbf{v}] \wedge \mathit{Profile}(S,\mathbf{v};\, \mathbf{x},\mathbf{v}))$$

Profiles are also closely related to weakest preconditions, but it will be convenient for us to formulate this relationship in terms of the dual of the weakest precondition, i.e., the set of states μ such that S, started in μ, terminates and the final state satisfies φ. This would be the weakest precondition for a total correctness interpretation of asserted programs, and it is the complement of the weakest precondition (in the usual, partial correctness sense) for $\neg\varphi$. If it is expressible by a formula, then we write this formula as $\mathit{Pre}(S, \{\varphi\})$. The connection with profiles is given by the equivalences

$$\mathit{Profile}(S,\mathbf{v};\, \mathbf{x},\mathbf{y}) \Leftrightarrow \mathit{Pre}(S\, [\mathbf{x}/\mathbf{v}], \{\mathbf{x} = \mathbf{y}\})$$

and

$$\mathit{Pre}(S, \{\varphi\}) \Leftrightarrow \exists \mathbf{y}\, (\varphi\, [\mathbf{y}/\mathbf{x}] \wedge \mathit{Profile}(S,\mathbf{v};\, \mathbf{v},\mathbf{y})).$$

Notice that the only logical operations used on the right sides of these equivalences are conjunction and existential quantification. Thus, in any logic having these operations, all of *Post, Pre,* and *Profile* will be expressible if one of them is.

Theorem 1. *For any while-program S with recursively defined procedures, the profile is expressible by a formula of* **E + LFP** *(the same formula for all structures). If* φ *is an* **E + LFP** *formula, then so are* $\mathit{Post}(\{\varphi\}, S)$ *and* $\mathit{Pre}(S, \{\varphi\})$.

Proof. In view of the remarks preceding the theorem, it suffices to prove the first assertion of the theorem. We do this by induction on programs. We need not consider the *while* construction since it is subsumed by recursion; *while* φ *do S od* is equivalent to the procedure P recursively declared by

$$P \Leftarrow \mathit{if}\ \varphi\ \mathit{then}\ (S; P)\ \mathit{else}\ x := x\ \mathit{fi}.$$

(If *skip* were available in the programming language, we would prefer it to $x := x$.) The other types of programs have profiles defined as follows. (Recall that the *if* φ *then*. . . construction is allowed only if φ is quantifier-free.)

$$Profile(v_i := t,; \mathbf{v};\ \mathbf{x, y}) \Leftrightarrow y_i = t[\mathbf{x}/\mathbf{v}] \wedge \bigwedge_{j \neq i} y_j = x_j\ ,$$

$$Profile((S_1; S_2),\mathbf{v};\ \mathbf{x, y}) \Leftrightarrow \exists z\ (Profile(S_1,\mathbf{v};\ \mathbf{x}, \mathbf{z}) \wedge Profile(S_2,\mathbf{v};\ \mathbf{z}, \mathbf{y})),$$

$$Profile(\ if\ \varphi\ \ then\ \ S_1\ \ else\ S_2\ \ f\,i,\ \mathbf{v};\ \mathbf{x, y}) \Leftrightarrow$$

$$(\varphi[\mathbf{x}/\mathbf{v}] \wedge Profile(S_1,\mathbf{v};\ \mathbf{x}, \mathbf{y})) \vee (\neg[\mathbf{x}/\mathbf{v}] \wedge Profile(S_2,\mathbf{v};\ \mathbf{x}, \mathbf{y})),$$

and, if P has the recursive declaration $P \Leftarrow S$, then

$$Profile(P,\mathbf{v};\ \mathbf{x}, \mathbf{y}) \Leftrightarrow \text{LET } R(\mathbf{x,y}) \leftarrow \pi \text{ THEN } \mathcal{R}\,(\mathbf{x,y}),$$

where π is the formula $Profile(S,\mathbf{v};\ \mathbf{x}, \mathbf{y})$ built by means of the preceding equivalences but using $\mathcal{R}\,(\mathbf{x,y})$ for $Profile(P,\mathbf{v};\ \mathbf{x}, \mathbf{y})$ wherever P occurs in S. It is easy to verify that all occurrences of $\mathcal{R}$ in π are positive, so the formulas defining profiles are all in **E+LFP.** This completes the proof of the theorem. ⊣

Recall that the *Pre* referred to in the theorem is the weakest precondition in the total correctness interpretation. The theorem immediately implies that the usual weakest precondition (for the partial correctness interpretation) satisfies a similar theorem with respect to the language whose formulas are the negations of those of **E+LFP,** a language that has universal (but not existential) quantification and the greatest (rather than least) fixed-point operator.

In the formulas used in the proof of Theorem 1 to express profiles, all the syntactic apparatus of **E+LFP** was used − connectives, the existential quantifier, and the least fixed point operator. But disjunction and existential quantification were used only in a restricted way. Disjunction occurred only with mutually exclusive disjuncts. Similarly, $\exists x\varphi(x)$ occurred only in contexts where at most one x can satisfy $\varphi(x)$. It might be interesting to weaken **E+LFP** by building such restrictions into its syntax. Since two disjuncts might be mutually exclusive in some structures but not in others, this restriction would, if taken literally, make the notion of formula dependent on structures. Such a dependence of syntax on semantics can be avoided by requiring disjuncts to be provably (in some formal system) incompatible, and similarly for existential quantification, but the resulting logic would still involve an unorthodox dependence of the notion of formula on the notion of proof.

Another aspect of the same observations is that **E+LFP** (without the restrictions discussed in the last paragraph) can express profiles not only for while-programs with recursive procedures but also for certain sorts of non-deterministic computations. For example, we could allow atomic programs of the form *guess* v_i, whose effect is to non-deterministically assign a (possibly) new value to the variable v_i, and whose profile with respect to $\mathbf{v}$, $\mathbf{x}$, and $\mathbf{y}$ is $\bigwedge_{j \neq i} y_j = x_j$. Then the existential quantifier in the profile of $S_1; S_2$ would no longer have the uniqueness property discussed above. As another example, Dijkstra's (1975) "alternative command" built from guarded commands $\varphi_1 \rightarrow S_1, \ldots, \varphi_n \rightarrow S_n$, with all φ_i quantifier-free, has a profile defined by $\bigvee_{i=1}^{n} (\varphi_i [\mathbf{x}/\mathbf{v}] \wedge \mathit{Profile}(S_i, \mathbf{v}; \mathbf{x}, \mathbf{y}))$.

3. Capturing polynomial time

One says that a logic *captures* a complexity class $\mathcal{C}$ for a collection $\mathcal{K}$ of finite structures if the subcollections of $\mathcal{K}$ definable in the logic are precisely those that belong to $\mathcal{C}$. That is, a subcollection $\mathcal{X}$ of $\mathcal{K}$ has the form $\{\mathcal{A} \in \mathcal{K} \mid \mathcal{A} \vDash \varphi\}$ for some sentence φ of the given logic if and only if the problem "Given a member of $\mathcal{K}$, decide whether it belongs to $\mathcal{X}$" is in the complexity class $\mathcal{C}$.

It is known, by work of Immerman (1982) and Vardi (1982) that **FO+LFP** captures PTIME, the class of problems solvable in polynomial time, for the class of structures of the form $(\{0, 1, \ldots, n\}, <, \ldots)$, where $<$ is the linear ordering $0 < 1 < \ldots < n$. We shall deal instead with the class $\mathcal{K}$ of structures of the form $(\{0, 1, \ldots, n\}, 0, n, S, \ldots)$, where the first and last elements, 0 and n, are the values of constant symbols 0 and *End*, and where S is the successor relation, $S(x, y) \Leftrightarrow x + 1 = y$.

Immerman's and Vardi's result remains true for the class $\mathcal{K}$, since 0, *End*, and S are easily definable in **FO** from $<$ and since $<$ is definable in **E+LFP** from S by

$$z < y \Leftrightarrow \text{LET } P(x, y) \leftarrow (S(x, y) \vee \exists z(P(x, z) \wedge S(z, y))) \text{ THEN } P(x, y).$$

In fact, Immerman's proof uses $<$ primarily to define 0, *End*, and S.

When we deal with systems lacking universal quantification, however, the equivalence between $<$ on the one hand and 0, *End*, and S on the other breaks down. The latter is the more natural set of primitives for describing computations of Turing machines, since the computation mechanism directly refers to the next moment of time and adjacent squares on the tape. Thus, it is not surprising that $\mathcal{K}$. is the appropriate class for extending Immerman's and Vardi's result to logics without $\forall$.

Theorem 2. E+LFP *captures PTIME for $\mathcal{K}$. If $\mathcal{K}$ is modified by taking the successor as a unary function S (with $S(\mathit{End}) = \mathit{End}$) rather than a binary relation, then in fact* QF+LFP *captures PTIME.*

It would be possible to prove this by going through Immerman's proof and verifying that $\forall$ is not needed for $\mathcal{K}$ and $\exists$ is not needed if the successor is available as a function. We shall, however, use a different approach. In view of Immerman's and Vardi's theorem, Theorem 2 is an immediate consequence of the following result.

Theorem 3. *On structures in* $\mathcal{K}$, E+LFP *can define* $\forall$ *and is therefore equivalent to* **FO+LFP.** *When the successor is available as a function, then* **QF+LFP** *can define* $\exists$.

Proof. Both parts are proved by using the least fixed point operator to search through the structure, thereby simulating a quantifier. For the first part, we have that $\forall x\varphi(x)$ is equivalent to

$$\text{LET } P(x) \leftarrow \varphi(x) \vee (x = 0 \vee \exists y(S(y, x) \wedge P(y))) \text{ THEN } P\ (End).$$

For the second part, we must, for technical reasons, search through the structure backward; $\exists x\varphi(x)$ is equivalent to

$$\text{LET } P(x) \leftarrow \varphi(x) \vee P(S(x)) \text{ THEN } P(0). \qquad \dashv$$

We shall see in the next section that the use of S rather than $<$ was essential for the results of this section; S is not E+LFP-definable in $(\{0, 1, . . . , n\}, <)$.

4. A preservation theorem

A result of classical model theory (see (Chang &; Keisler 1973) page 34) characterizes the formulas of **E** up to logical equivalence as those first-order formulas φ whose truth is preserved by extensions, i.e., if $\mathcal{A}$ is a substructure of $\mathcal{B}$ (written $\mathcal{A} \subseteq \mathcal{B}$) and $\mathcal{A} \vDash \varphi[\mu]$ (where μ maps variables to values in $\mathcal{A}$), then $\mathcal{B} \vDash \varphi[\mu]$. We shall show that formulas of **E+LFP** have the same preservation property. We shall also prove that truth of **E+LFP**-formulas is preserved when the relations that interpret positive predicate symbols are increased. It is convenient to prove these results (and somewhat more) simultaneously; to do this we introduce the notion of homomorphism that is appropriate for our signatures that specify negatable and positive predicate symbols.

Let $\mathcal{A}$ and $\mathcal{B}$ be two structures for the same signature σ. A function h: $\mathcal{A} \rightarrow \mathcal{B}$ is a *homomorphism* if
(a) h commutes with functions:

$$h(f_A(a_1, . . . , a_n)) = f_B\,(h(a_1), . . . , h(a_n))$$

for all function symbols f of σ and all $a_1, . . . , a_n \in \mathcal{A}$.
(b) h preserves positive relations:

$$\mathcal{R}_A(a_1, . . . , a_n) \Rightarrow \mathcal{R}_B(h(a_1), . . . , h(a_n))$$

for all positive $\mathcal{R}$ and all $a_1, . . . , a_n \in \mathcal{A}$.
(c) h preserves and reflects negatable relations:

$$\mathcal{R}_A(a_1, . . . , a_n) \Leftrightarrow \mathcal{R}_B(h(a_1), . . . , h(a_n))$$

for all negatable $\mathcal{R}$ and all $a_1, . . . , a_n \in \mathcal{A}$.

Two types of homomorphisms will be of particular importance to us. First, if $\mathcal{A} \subseteq \mathcal{B}$, then the identity function on $\mathcal{A}$ is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$. Second, if $\mathcal{A}$ and $\mathcal{B}$ are identical except that, for some positive predicate symbols, the interpretations in $\mathcal{B}$ are supersets of those in $\mathcal{A}$, then again the identity function is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$. If equality is negatable in σ, then clause (c) requires homomorphisms to be one-to-one, and it follows easily that every homomorphism is a composite of homomorphisms of these special types and an isomorphism. If, on the other hand, equality is positive in σ, then our definition admits homomorphisms that are not one-to-one.

Theorem 4. *Let h: $\mathcal{A} \rightarrow \mathcal{B}$ be a homomorphism,* μ *a function assigning values in $\mathcal{A}$ to variables, and* φ *a formula of* E+LFP. *If* $\mathcal{A} \models \varphi[\mu]$, *then* $\mathcal{B} \models \varphi[h \circ \mu]$

Proof. We first observe that, by induction on terms using clause (a) of the definition, $h(t_{\mathcal{A}}[\mu]) = t_{\mathcal{B}}[h \circ \mu]$ for all terms t. We then prove the theorem by induction on φ. Clause (b) and the $\Rightarrow$ half of clause (c) give the result for atomic φ, and the other half of (c) takes care of negations. The cases of conjunction, disjunction, and existential quantification are easy (just as in first-order logic). It remains to prove the theorem when φ is obtained by LET…THEN... from formulas for which the theorem is true. To simplify notation, we suppose that φ is

$$\text{LET } P(x) \leftarrow \delta \text{ THEN } \psi$$

where P is unary. (The general case is handled the same way.) Of course, δ and ψ are formulas of the signature $\sigma \cup \{P\}$ where P is positive.

Let $\Delta^{\alpha}(\mathcal{A}) \subseteq \mathcal{A}$ and $\Delta^{\alpha}(\mathcal{B}) \subseteq \mathcal{B}$ be as in the definition of the semantics of LET…THEN... in Section 1. We shall show, by induction on the ordinal a, that h is a homomorphism from $\mathcal{A}^{\alpha} = (\mathcal{A}, \Delta^{\alpha}(\mathcal{A}))$ to $\mathcal{B}^{\alpha} = (\mathcal{B}, \Delta^{\alpha}(\mathcal{B}))$. In view of the assumption that h is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$, what we must prove is (since P is positive) that $h(\Delta^{\alpha}(\mathcal{A})) \subseteq \Delta^{\alpha}(\mathcal{B})$. The cases of 0 and limit ordinals are trivial, so suppose $\alpha = \beta + 1$ and h is a homomorphism from $\mathcal{A}^{\beta}$ to $\mathcal{B}^{\beta}$. Then, for any $a \in \mathcal{A}$,

$$\begin{aligned} a \in \Delta^{\alpha}(\mathcal{A}) &\Leftrightarrow \mathcal{A}^{\beta} \models \delta[a/x;\ \textit{otherwise}\ \mu] \\ &\Rightarrow \mathcal{B}^{\beta} \models \delta[h(a)/x;\ \textit{otherwise}\ h \circ \mu] \\ &\Leftrightarrow h(a) \in \Delta^{\alpha}(\mathcal{B}), \end{aligned}$$

where the two $\Leftrightarrow$'s are from the definition of Δ^{α} and the $\Rightarrow$ is from the hypotheses that the theorem holds for δ and that h is a homomorphism from $\mathcal{A}^{\beta}$ to $\mathcal{B}^{\beta}$. This completes the proof that h is a homomorphism from $\mathcal{A}^{\alpha}$ to $\mathcal{B}^{\alpha}$.

In particular, taking α large enough, we have that h is a homomorphism from $\mathcal{A}^{\infty}$ to $\mathcal{B}^{\infty}$. Thus, applying the induction hypothesis that the theorem holds for ψ, we have

$$\mathcal{A}^{\infty} \models \psi[\mu] \ \Leftrightarrow\ \mathcal{B}^{\infty} \models \psi[h \circ \mu],$$

i.e.,

$$\mathcal{A} \models \varphi[\mu] \Leftrightarrow \mathcal{B} \models \varphi[h \circ \mu],$$

as desired. This completes the proof of the theorem. ⊣

We obtain two corollaries by applying the theorem to homomorphisms of the special types described above.

Corollary. *With* $\mathcal{A}$, μ, *and* φ *as in the theorem, if* $\mathcal{A} \models \varphi[\mu]$ *and* $\mathcal{A} \subseteq \mathcal{B}$, *then* $\mathcal{B} \models \varphi[\mu]$.

Corollary. *With* $\mathcal{A}$, μ, *and* φ *as in the theorem, if* $\mathcal{B}$ *is identical with* $\mathcal{A}$ *except that some positive relations have been enlarged, and if* $\mathcal{A} \models \varphi[\mu]$, *then* $\mathcal{B} \models \varphi[\mu]$.

The last corollary says that **E+LFP**-formulas are monotone with respect to positive predicate symbols. It justifies the terminology "LFP".

The first corollary shows that, as we claimed earlier, **E+LFP** cannot in general express universal quantification. Indeed, in the signature having just one unary predicate symbol P, the **FO**-formula $\forall x P(x)$ is not preserved by extensions, hence is not expressible in **E+LFP**. The theorem similarly implies that (in the notation of Section 3) 0, *End*, and S are not **E+LFP**-definable on structures of the form $(\{0, 1, \ldots, n\}, <)$, for there are homomorphisms between such structures that fail to preserve 0, *End*, and S; the simplest example is the function from {0, 1} to {0, 1, 2, 3, 4} that sends 0 to 1 and 1 to 3. Thus, **E+LFP** can define $<$ from S but cannot define S from $<$, the exact opposite of the situation for **FO**.

The fact that **E+LFP** cannot in general express universal quantification should be contrasted with Theorem 3. The argument for the undefinability of $\forall$ cannot be applied within the class $\mathcal{K}$ considered in Theorem 3 because there are no non-trivial homomorphisms between structures in *K*.

Corollary. *If a* **FO**-*formula and an* **E+LFP**-*formula are logically equivalent, then they are equivalent to an* **E**-*formula*.

Proof. Such an **FO**-formula is, by the first corollary above, preserved by extensions. By the theorem quoted from (Chang &; Keisler 1973) at the beginning of this section, it is equivalent to an **E**-formula. ⊣

5. Decision problems and finiteness

In this section, we treat the decision problems for satisfiability and validity of **E+LFP**-sentences. In contrast to languages like **FO** that are closed under negation, **E+LFP** does not have its validity and satisfiability problems dual to each other. In fact, we shall see that for **E+LFP** both of these problems are complete recursively enumerable sets.

The proofs of these facts involve two other results that are of some independent interest. One gives a connection between **E+LFP** and a certain fragment of second-order logic. The other shows that, when an **E+LFP**-formula is satisfied in a structure, this fact depends on only a finite part of the structure. This also implies that the iterations involved in the semantics of LET…THEN... always stabilize at or before stage ω, the first transfinite stage.

We begin by defining the relevant fragment of second-order logic. In analogy with the terminology "strict Π_1^1} " used in admissibility theory (see (Barwise 1975)), we define a second-order formula to be *strict* $\forall_1^1$ if it consists of a string of universal (second-order) quantifiers over relations, followed by a string of existential first-order quantifiers, followed by a quantifier-free formula. We write such a formula as $\forall\mathbf{X}\exists\mathbf{y}\,\varphi$. The word "strict" refers to the restrictions that the second-order quantifiers are over relations, not functions, and that the first-order quantifiers are all existential. If either of these restrictions were removed, the other would become pointless. (With universal quantifiers over functions, one could use Skolem functions to achieve the effect of universal first-order quantification; with universal first-order quantifiers, one could restrict relation variables to range over functions only.)

Theorem 5. *Every formula of* **E+LFP** *is logically equivalent to a strict* $\forall_1^1$ *formula.*

Proof. We proceed by induction on **E+LFP**-formulas. The atomic and negation cases are trivial, and the conjunction and disjunction cases are handled by the familiar prenexing operations. The existential quantifier is handled by a standard technique for moving second-order quantifiers to the left past first-order ones; specifically, $\exists z\forall X\theta$ is equivalent to $\forall X'\exists z\theta'$, where X' has one more argument place than X, and where θ' is obtained from θ by replacing X with X' and inserting z as the extra argument.

Finally, consider (for notational convenience, as in the proof of Theorem 4) LET $P(x) \leftarrow \delta$ THEN φ, and let Δ be as in the definition of the semantics of this formula. Although the definition says that LET $P(x) \leftarrow \delta$ THEN φ is true in $\mathcal{A}$ if and only if φ is true in $(\mathcal{A}, \Delta^\infty)$, where Δ^∞ is the least fixed point of Δ, the monotonicity result proved in the last section allows us to equivalently formulate this condition as: φ is true in $(\mathcal{A}, X)$ for all X such that $\Delta(X) \subseteq X$, since Δ^∞ is the smallest such X. Therefore, LET $P(x) \leftarrow \delta$ THEN φ is equivalent to

$$\forall P\,(\forall x(\delta \Rightarrow P(x)) \Rightarrow \varphi).$$

If we insert into this formula strict $\forall_1^1$ equivalents for δ and φ, and apply prenexing operations (including moving second-order quantifiers past first-order ones as above), we obtain a formula in strict $\forall_1^1$ form. ⊣

The converse of Theorem 5 is false, at least if P$\neq$NP. To see this, recall that any class of finite structures definable in **E+LFP** (or even **FO+IFP**) is PTIME recognizable. On the other hand, it is easy to define the class of non-3-colorable graphs, a co-NP complete class, by a strict $\forall_1^1$ formula.

As our first application of Theorem 5, we characterize the recursion-theoretic complexity of the decision problem for validity in **E+LFP**.

Theorem 6. *The set of logically valid* **E+LFP**-*sentences is a complete recursively enumerable set. The set of* **E+LFP**-*sentences true in all finite structures is a complete co-r.e. set.*

Proof. Notice first that the proof of Theorem 5 gives a recursive translation from **E+LFP**-sentences to equivalent strict $\forall_1^1$ sentences. The validity of $\forall \mathbf{X} \exists \mathbf{y}\ \varphi$ is equivalent to the validity of $\exists \mathbf{y}\ \varphi$. Thus, we have a recursive many-one reduction of the validity problem for **E+LFP** to the validity problem for first-order logic (in fact for **E**). As the latter is recursively enumerable, by Gödel's completeness theorem, so is the former.

To prove completeness, we use the well-known fact that (the duals of) Skolem normal forms provide a recursive many-one reduction of the validity problem for **FO** to that for **E**. As the former is a complete recursively enumerable set and **E+LFP** includes **E**, this suffices to finish the proof of the first assertion.

The second assertion is proved by a similar reduction to well-known facts about **FO**. $\dashv$

Before considering the satisfiability problem for **E+LFP**, it will be useful to obtain a model- theoretic finiteness result as a consequence of Theorem 5. In formulating this result, we use the notation $\mathcal{A}\,|F$, where $\mathcal{A}$ is a structure and F is a subset of $\mathcal{A}$, for the set F with the restrictions to F of the relations and functions of $\mathcal{A}$; if F is not closed under some of these functions, then we restrict their ranges to F also, so $\mathcal{A}\,|F$ involves partial functions.

Theorem 7. *Let* φ *be an* E + LFP *-formula (or just a strict* $\forall_1^1$ *formula),* $\mathcal{A}$ *a structure, and* μ *a function assigning values in* $\mathcal{A}$ *to the free variables of* φ, *such that* $\mathcal{A} \vDash \varphi\,[\mu\,]$. *Then* $\mathcal{A}$ *has a finite subset F, containing the (finitely many) values of* μ, *such that, if* $\mathcal{B}$ *is any other structure with* $F \subseteq \mathcal{B}$ *and* $\mathcal{A}\,|F = \mathcal{B}\,|F$, *then* $\mathcal{B} \vDash \varphi[\mu]$.

This result, for strict $\forall_1^1$ formulas, was known to Mal'cev (1959).

Proof. In view of Theorem 5, it suffices to prove Theorem 7 in the case that φ is a strict $\forall_1^1$ formula

$$\forall \mathbf{X}_1 \ldots \forall \mathbf{X}_k\ \exists \mathbf{y}_1 \ldots \exists \mathbf{y}_n \theta\ .$$

The assumption that $\mathcal{A} \models \varphi\ [\mu\ ]$ can be reformulated as follows. Consider instances $\theta(\mathbf{b})$ of θ, obtained by replacing the variables $\mathbf{y}$ in θ with (names for) elements $\mathbf{b}$ of $\mathcal{A}$. As θ is quantifier-free, these instances can be viewed as formulas of propositional logic, with instances of atomic subformulas of θ as the prepositional variables. For those atomic formulas that don't involve the predicate variables X_i, a truth value is determined by $\mathcal{A}$ and μ, and we think of each $\theta(\mathbf{b})$ as already having these truth values substituted in. So these instances $\theta(\mathbf{b})$ of θ are formulas of propositional logic with propositional variables of the form $X_i(\mathbf{a})$. Now the assumption that $\mathcal{A} \models \varphi\ [\mu]$ means that, no matter how we assign truth values to these "propositional variables" $X_i(\mathbf{a})$, at least one of the instances $\theta(\mathbf{b})$ is true. In other words, the negations $\neg\theta(\mathbf{b})$ are not simultaneously satisfiable. By the compactness theorem of propositional logic, finitely many of these negations, say $\neg\theta(\mathbf{b}_1), \ldots, \neg\theta(\mathbf{b}_r)$, are not simultaneously satisfiable. Let F be the finite subset of $\mathcal{A}$ obtained by evaluating each of the terms that occurs in θ, with each of the $\mathbf{b}_i$ as values of $\mathbf{y}$, and with the values of the other variables given by μ. Then, if $\mathcal{B}$ is as in the theorem, exactly the same unsatisfiable collection $\{\neg\theta(\mathbf{b}_1), \ldots, \neg\theta(\mathbf{b}_r)\}$ will occur among the negations of the instances of θ in $\mathcal{B}$. Therefore, every truth assignment must verify at least one instance of θ in $\mathcal{B}$, i.e., $\mathcal{B} \models \forall\mathbf{X}\ \exists\mathbf{y}\theta$. $\dashv$

Corollary. *A sentence of* **E+LFP** *(or a strict* $\forall_1^1$ *sentence) is satisfiable if and only if it is satisfiable in a finite structure.*

Proof. *If* $\mathcal{A} \models \varphi$ and if F is as in Theorem 7, then let $\mathcal{B}$ be the finite structure obtained from $\mathcal{A}\,|F$ by adding one new element *, setting all function values that were undefined in $\mathcal{A}\,|F$ equal to * in $\mathcal{B}$, and extending the relations of $\mathcal{A}\,|F$ arbitrarily to $\mathcal{B}$. Then $\mathcal{B}\,|F = \mathcal{A}\,|F$, so, by the theorem, $\mathcal{B} \models \varphi$. $\dashv$

The preceding corollary provides an upper bound for the complexity of the satisfiability problem.

Theorem 8. *The set of satisfiable* **E+LFP**-*sentences (or strict* $\forall_1^1$ *sentences) is a complete recursively enumerable set.*

Proof. Recursive enumerability is easy to prove, in view of the corollary to Theorem 7. If a sentence is satisfiable, then this fact can be verified by exhibiting a finite structure that satisfies it.

The proof of completeness is more difficult and requires some preliminary considerations concerning a limited sort of universal quantification that is available in **E+LFP.** The idea here is similar to that used in the proof of Theorem 3, but we can no longer restrict our attention to a class (like $\mathcal{K}$ in Theorem 3) of particularly well-behaved structures.

Consider a signature that contains constant symbols 0 and *End* and a unary function symbol S (and possibly other symbols). We would like to consider structures from the class $\mathcal{K}$ of Section 3, but, by Theorem 4, no **E+LFP**-sentence can characterize this class. Nevertheless, we can easily express in **E+LFP** that *End* can be reached from 0 by finitely many applications of S:

$$\text{LET } P(x) \leftarrow x = End \vee P(S(x)) \text{ THEN } P(0).$$

More importantly, we can express that a formula $\varphi(x)$ holds at 0 and *End* and all the intermediate points in this iteration of S:

$$\text{LET } P(x) \leftarrow \varphi(x) \wedge (x = End \vee P(S(x))) \text{ THEN } P(0).$$

Thus, although we cannot quantify universally over the whole universe, we can quantify universally over the (unique) finite S-chain joining 0 to *End.*

It is now a routine matter, which we omit, to use this limited sort of universal quantification to express the halting of a Turing machine computation by an **E+LFP**-sentence. That is, for any Turing machine M, there is an **E+LFP**-sentence that is satisfied in a structure if and only if the structure has an S-chain from 0 to *End* and the other relations of the structure, restricted to this S-chain, encode (in some standard fashion) a halting computation of M on a totally blank input tape. The sentence here depends recursively on M, so we have a many-one reduction of the halting problem to the satisfiability problem for **E+LFP**. The completeness of the former therefore implies the completeness of the latter. ⊣

We conclude this section with another application of Theorem 7. It provides additional support for the constructive nature of **E+LFP**. In stating it, we use the standard notation ω for the first transfinite ordinal.

Theorem 9. *Let Δ be the operator used in the definition of the semantics of an* **E+LFP**-*formula*

$$\text{LET } P_i(x^1) \leftarrow \delta_1, \ldots, P_n(\mathbf{x}^n) \leftarrow \delta_n \text{ THEN } \varphi.$$

Then $\Delta^\omega = \Delta^\infty$.

Proof. For notational simplicity, we give the proof for LET $P(x) \leftarrow \delta$ THEN φ. In view of the definitions of Δ^α and Δ^∞, it suffices to prove that $\Delta(\Delta^\omega) \subseteq \Delta^\omega$.

So suppose that $a \in \Delta(\Delta^\omega)$. By definition of Δ, we have $(\mathcal{A}, \Delta^\omega) \models \delta\,[a/x]$. Apply Theorem 7 to get a finite subset F of $\mathcal{A}$ such that, whenever $(\mathcal{B}, X) \mid F = (\mathcal{A}, \Delta^\omega) \mid F$ then $(\mathcal{B}, X) \models \delta\,[a/x]$. As Δ^ω is the union of Δ^n over all finite n, each element of $F \cap \Delta^\omega$ is in some Δ^n, and, as F is finite, one n works for all elements of $F \cap \Delta^\omega$ simultaneously. Fix such an n. Then $(\mathcal{A}, \Delta^n) \mid F = (\mathcal{A}, \Delta^\omega) \mid F$. So $(\mathcal{A}, \Delta^n) \models \delta\,[a/x]$. But this means that

$$a \in \Delta(\Delta^n) = \Delta^{n+1} \subseteq \Delta^\omega$$

as desired. ⊣

Appendix. **U+LFP**

For the sake of completeness, we briefly discuss fixed-point logic with the universal, rather than the existential, quantifier. Notice that, by simply negating formulas of **E+LFP**, we get the logic with $\forall$ and the *greatest* fixed-point operator. We are interested rather in the logic **U+LFP** with $\forall$ and the *least* fixed-point operator.

By a *weak substructure* of a structure $\mathcal{A}$, we mean the result of taking a substructure of $\mathcal{A}$ and then possibly enlarging the relations that interpret some positive predicate symbols. Then one can show, by induction on **U+LFP**-formulas, that any such formula true in $\mathcal{A}$ remains true in any weak substructure that contains the values assigned to the variables. In particular, truth of **U+LFP**-formulas is preserved by substructures. It follows, for example, that $\exists x P(x)$ is not equivalent to any **U+LFP**-formula. In fact, if a **FO**-formula and a **U+LFP**-formula are equivalent, then they are also equivalent to a **U**-formula.

By Theorem 3, **U+LFP**-formulas can define $\exists$ for structures in the class $\mathcal{K}$, provided that S is available as a function. In fact, it suffices to have S and equality as negatable predicate symbols, since $\exists x \varphi(x)$ is equivalent to

$$\text{LET } P(x) \leftarrow \varphi(x) \ \vee \ (x \neq End \wedge \forall y\, (P(y) \vee \neg S(x, y)) \text{ THEN } P(0).$$

(We could also replace the constants 0 and *End* with negatable predicate symbols.) Thus, on structures of this sort, **U+LFP** captures PTIME.

The finiteness properties proved in Section 5 for **E+LFP** fail badly for **U+LFP**. For example, if the structure $\mathcal{A}$ is a linearly ordered set, then the formula

$$\text{LET } P(x) \leftarrow \forall y\, (P(y) \vee \neg y < x) \text{ THEN } P(x)$$

defines the largest well-ordered initial segment of $\mathcal{A}$, and the iteration occurring in the definition of the meaning of this formula requires an ordinal number of steps equal to the length of this segment. The same idea shows that, in the standard model of arithmetic with suitable primitive recursive predicates taken as atomic formulas, **U+LFP** can define certain complete Π_1^1 sets, the well-founded parts of recursively enumerable relations. In contrast, by the results of Section 5, **E+LFP** can define only recursively enumerable sets.

Bibliography

Apt, K. (1981) Ten years of Hoare's logic: A survey - Part I. *ACM Trans. Program. Lang. Syst.* 3 pp. 431-483.

Barwise, K. J. (1975) *Admissible Sets and Structures.* Springer-Verlag.

Chang, C. C. and Keisler, H. J. (1973) *Model Theory.* North-Holland.

Cook, S. A. (1978) Soundness and completeness of an axiom system for program verification. *SIAM J. Computing 7* pp.70-90.

Dijkstra, E. W. (1975) Guarded commands, nondeterminacy, and the formal derivation of programs. *Cotnm. ACM* 18 pp.453-457.

Hoare, C. A. R. (1969) An axiomatic basis for computer programming. *Comm. ACM* 12 pp.576-580, 583.

Immerman, N. (1982) Relational queries computable in polynomial time. *14th ACM STOC pp.U7-152.*

Mal’cev, A. I. (1959) Modelniye sootvetstviya (Model correspondences). *Izv. Akad. Nauk SSSR Ser. Mat.* 23 pp.313-336 (Russian).

Vardi, M. (1982) Complexity of relational query languages *14th ACM STOC* pp. 137-146.