The Logic of Choice

Andreas Blass*
Math. Dept.
University of Michigan
Ann Arbor, MI 48109-1109
ablass@math.lsa.umich.edu

Yuri Gurevich†‡
Microsoft Research
One Microsoft Way
Redmond, WA 98052
gurevich@microsoft.com

Abstract

The choice construct ($\texttt{choose}\ x:\ \varphi(x)$) is useful in software specifications. We study extensions of first-order logic with the choice construct. We prove some results about Hilbert's ε operator, but in the main part of the paper we consider the case when all choices are independent.

Part I
Introduction and Preliminaries

1 Introduction

This investigation was motivated by the choice construct in ASM programs, that is the programs of abstract state machines [Gurevich 1995]. The ASM choice terms have the form

$$(\texttt{choose}\ x : \varphi(x))$$

where $\varphi(x)$ is a formula (that is a Boolean-valued term[1]). Intuitively ($\texttt{choose}\ x$: $\varphi(x)$) is an element of the set $\{x : \varphi(x)\}$.

Here we are interested mainly in the extension of first-order logic with the choose construct ($\texttt{choose}\ x : \varphi(x)$). Syntactically, the desired extension is obvious. Just add the following formation rule:

- If $\varphi(x)$ is a formula, then ($\texttt{choose}\ x : \varphi(x)$) is a term. All occurrences of x in the term are bound. For any other variable y, every free (respectively bound) occurrence of y in φ remains free (respectively bound) in the term.

*Partially supported by NSF grant DMS 95-05118
†Partially supported by NSF grant CCR 95-04375
‡On leave of absence from EECS Department, University of Michigan, Ann Arbor, Michigan
[1]Boolean-valued in the sense of programming, not set theory. A Boolean-valued term is simply a term with values in $\{\texttt{true}, \texttt{false}\}$.

The semantics of the extended logic is not obvious. Evaluating a term or a formula may involve numerous actions of choice. Should they be correlated, and if yes then how? Different correlation strategies may give rise to different choice operators and thus to different logics.

We restrict attention to correlation strategies satisfying the following constraint:

- The choices from a set Z are independent of the choices from all sets Z' different from Z.

This leaves the possibility that a particular choice from a given set Z depends on other choices from Z. Two extreme cases arise naturally:

Fixed-choice strategy: For any set Z, all choices from Z give the same result.

Independent-choice strategy: For any set Z, different choices from Z are independent.

Notice that there are other natural strategies. For example, the current choice from Z can depend on the previous choices from Z but be independent from concurrent choices from Z. And concurrent choices from the same set do arise, quite naturally. Consider for example a term

$$(\texttt{choose}\ x : E(x, y))$$

where E is the edge relation of a graph. There may be $y_1 \neq y_2$ with $\{x : E(x, y_1)\} = \{x : E(x, y_2)\}$, and then we have concurrent choices from the same set.

What should one do when the choice set is empty? One may say that choosing from the empty set is impossible and thus the term $(\texttt{choose}\ y : \varphi(\bar{x}, y))$ gives in general a partial function. Traditionally first-order logic deals only with total functions. In the ASM context, structures (the states of ASMs) come equipped with default elements; choosing from the empty set results in the default element. This avoids dealing with partial functions. We borrow that solution. For most of our work, we use structures equipped with default elements and choosing from the empty set results in the default element. (Alternatively, one can stipulate that choosing from the empty set is like choosing from the whole base set: the result can be any element of the base set. We will explore this solution in Section 8.)

The fixed-choice strategy turns the choice operator into Hilbert's well known epsilon operator [Hilbert and Bernays 1939]. The extension of first-order logic with this operator will be called FO+ε. It is natural and usual to suppose in this case that every structure comes equipped with a choice function. In other words, an FO+ε structure has the form (A, F) where A is a usual first-order structure with base set $|A|$ and F is a function from the power set $2^{|A|}$ to $|A|$. In particular, $F(\emptyset)$ is the default element discussed above.

The independent-choice strategy gives another version of the choice operator which will be called δ.[2] The extension of first-order logic with δ will be denoted FO+δ. An FO+δ structure is a usual first-order structure equipped with a default element. More generally, we will consider many-sorted structures where every sort comes with its own default element.

We devote two sections to the ε operator. The main part of the paper is devoted to the investigation of FO+δ. Finally, we consider briefly an alternative independent-choice logic and the witness operator studied in [Abiteboul-Vianu 1991].

1.1 The ε Operator

The ε operator was introduced in [Hilbert and Bernays 1939] for proof-theoretic purposes and played an important role in proof theory [Leisenring 1969]. Here we are interested in model theory. The extension FO+ε of first-order logic with the ε operator is the object of Section 3 which is a kind of introduction to a general study of the ε operator. We explain the background and pose a couple of questions; no new results are proved there.

Notice that FO+ε formulas are evaluated at structures equipped with a choice function. Can one use FO+ε formulas to speak about usual structures (not equipped with a choice function)? One way to do that is to restrict attention to FO+ε formulas whose meaning does not depend on the choice function; such *ε-invariant* formulas are considered in Section 3. It is easy to check (we do that in Section 3) and it is well known that every ε-invariant formula is equivalent to a first-order formula. It may seem therefore that the operator ε is benign. But this depends on the compactness theorem and therefore on the presence of infinite structures. Martin Otto found a nonelementary property of finite structures which is expressible by a sentence which is ε-invariant over finite structures [Otto 1998]. The power of ε manifests itself also in the presence of generalized quantifiers [Caicedo 1995].

Another way to utilize FO+ε formulas in order to describe ordinary structures is to quantify the choice function out. To this end, we introduce *ε-existential formulas* $(\exists\,\varepsilon)\varphi$ where φ is a formula in FO+ε. In Section 4, we compare the expressive power of ε-existential formulas with that of existential second-order formulas (that is Σ^1_1 formulas). Notice that there are more choice functions than relations of any fixed arity on a given sufficiently large set. On the other hand, the choice operator applies only to unary relations and not to relations of higher arities. It turns out nevertheless that every ε-existential formula is equivalent to a Σ^1_1 formula and the other way round. Using Fagin's characterization of NP [Fagin 1974], we conclude that, on finite structures, ε-existential formulas capture NP.

We prove also that every ε-existential formula $(\exists\,\varepsilon)\varphi$ is equivalent to an ε-existential formula $(\exists\,\varepsilon)\psi$ such that, in ψ, every application of ε has the form

$$\varepsilon x(x = t_1 \vee \cdots \vee x = t_j)$$

where the terms t_i are first-order and x is not free in any term t_i. Furthermore, if

[2] δ is the first letter of the classical Greek word $\delta\iota\alpha\lambda\varepsilon\gamma\omega$ which means *choose*. There is another classical Greek word $\varepsilon\pi\iota\lambda\varepsilon\gamma\omega$ with a close meaning which could have inspired Hilbert's notation and which is used to translate *axiom of choice*: $\alpha\xi\iota\omega\mu\alpha$ $\varepsilon\pi\iota\lambda o\gamma\eta\sigma$. (Thanks to Naomi Gurevich and Phokion Kolaitis for the help with Greek.)

r is the maximal arity of quantified relation symbols in φ and $s = \max\{r, 3\}$ then $j \leq 2s + 1$.

1.2 The δ Operator

The semantics of the *independent choice operator* δ is tricky and may be confusing. For example, the formula

$$\Big(\delta x(x = x)\Big) = \Big(\delta x(x = x)\Big) \tag{1}$$

is not valid because the two choices are independent and do not necessarily produce the same result. (In contrast, the formula $(\varepsilon x(x = x)) = (\varepsilon x(x = x))$ is valid of course.)

Given that δ is intended to represent independent choices, exactly how should a term of the form $\delta x\, \varphi(x)$ be evaluated? Ultimately, we should choose one of the elements a making $\varphi(a)$ true (or default if there is no such a), but this presupposes that we know which a's these are, which may in turn depend on the (arbitrary) evaluation of δ terms within φ. "Independent choice" means that all the choices involved in the evaluations of the various $\varphi(a)$ are done completely arbitrarily, and so is the final choice of an a from among those for which $\varphi(a)$ was evaluated as true. If the evaluation process involves choosing several times from the same set, then the chosen element may be different each time. Thus, the semantics of δ is as follows; for more details, see Sections 5 and 6.

To evaluate a term $\delta x \varphi(x)$ in a structure with fixed values for the free variables of the term, evaluate $\varphi(a)$ for every value a of x. If there are elements a for which $\varphi(a)$ evaluates to `true`, then choose one of those elements to be the value of $\delta x \varphi(x)$; otherwise the value of $\delta x \varphi(x)$ is the default element (of the appropriate sort). Because of the arbitrary choices the process of evaluating terms and formulas is not deterministic. We approach the task of defining this nondeterministic semantics for FO+δ systematically. In Section 5 we introduce nondeterministic propositional logic. In Section 6, we introduce nondeterministic first-order logic. In Section 7, we finally define the semantics of FO+δ. The extensive preparatory work makes this task easier.

Initially we wanted to enrich FO+δ with another ASM construct:

$$(\texttt{let } x \texttt{ be } s \texttt{ in } \tau(x))$$

where x is a variable, s is a term without free occurrences of x, and τ is term or formula [Gurevich 1995, Gurevich 1997]. The purpose was to allow one to reuse the same choice; for example,

$$(\texttt{let } x \texttt{ be } \delta y(y = y) \texttt{ in } (x = x))$$

is valid. But then we noticed that the *let* construct is expressible in FO+δ. We show (in the same Section 7) that FO+δ admits quantifier elimination in the sense that the quantifiers can be expressed by means of the construct δ. We show also that every FO+δ formula is equivalent, in a suitable sense, to a first-order formula whether or not infinite structures are allowed. In that sense, δ (unlike ε) is a benign choice operator. The power of δ becomes apparent in dynamic situations; see an example in Section 7.

One result of relevance to ASM applications is that, in FO+δ, every term is equivalent to a term $\delta x \varphi(x)$ where $\varphi(x)$ is first-order.

FO+δ can be extended in many ways. Many appetizing questions arise. At the end of Section 7, we give a few initial remarks on $L^{\omega}_{\infty,\omega}$+$\delta$, on the extension FO+DTC+δ of FO+δ with the deterministic transitive closure operator, and on the extension FO+IFP+δ of FO+δ with the inflationary fixed-point operator.

1.3 Additional Logics

The semantics of δ described above is entirely reasonable from the point of view of a specific evaluation. In order to evaluate $\delta x \varphi(x)$, you compute the set $Z = \{x : \varphi(x)\}$ and choose an element a (if Z is nonempty). Notice, however, that Z itself may depend on choices made earlier during the evaluation. Consider now the global picture of all evaluations. The element a chosen in a particular evaluation may not even belong to the set Z in another evaluation. On the other hand, there might be a "better" element b that belongs to the set Z in every evaluation. Shouldn't we choose b instead of a? In Section 8, we present another semantics for δ that reflects this idea. We discuss also the relative advantages of the two semantics of δ.

In [Abiteboul and Vianu 1991], a nondeterministic witness-choosing operator was introduced in the context of relational algebra. The semantics is described informally there. We suggest a formalization and make a couple of observations in Appendix A.

Acknowledgment

We thank Janos Makowsky for commenting on a draft of the paper.

2 Relational and Functional First-Order Logics

We discuss two versions of first-order logic with equality: relational first-order logic (RFO) and functional first-order logic (FFO) where formulas are Boolean-valued terms.

In the two sections devoted to the ε operator, we use RFO because most of the papers on the ε operator use RFO, because theorems we cite use RFO, and because using FFO would not buy us any significant advantage. On the other hand, FFO will be used in sections devoted to the δ operator. There are three reasons for the use of FFO in that part of the paper. First, the introduction of the δ operator is motivated by research on abstract state machines [ASM] where FFO is used. Second, many of our inductive definitions are simpler in FFO: there is no need to give a definition twice, first for terms and then for formulas. Third, the presence of defaults is handy.

Of course the difference between the two versions of first-logic is inessential and usually one can easily translate results from one version to another.

2.1 Relational First-Order Logic (RFO)

RFO is classical first-order logic. We give a couple of definitions to establish our terminology.

2.1.1 Pebbled Structures

A structure A together with a variable assignment ξ over A will be called a *pebbled structure*. If V is the domain of ξ, we say that (A, ξ) is a V-pebbled structure. The term "pebbled structure" is motivated by Ehrenfeucht-Fraïsse games [Ebbinghaus and Flum 1995, page 21]. Suppose that B is a V-pebbled structure (A, ξ). The base set of B is the base set of A, and the vocabulary of B is that of A. If $x \in V$, we write $B(x)$ instead of $\xi(x)$.

Pebbled structures are used to give semantics to RFO. A formula φ and a V-pebbled structure B are *appropriate* for each other if the vocabulary of B includes that of φ and V includes all free variables of φ. By induction on the formula φ, one defines the truth value $Val^B(\varphi)$ of φ in a pebbled structure B appropriate for φ.

2.1.2 Global Relations

A j-ary *global relation* γ of vocabulary Υ is a function that assigns to every Υ-structure A a (local) j-ary relation γ^A on A; it is required that γ be *abstract* in the following sense: every isomorphism between Υ-structures A and A' is also an isomorphism between the expanded structures (A, γ^A) and $(A', \gamma^{A'})$ [Gurevich 1984, 1988].

It is often convenient to endow γ with particular variables $x_1, \ldots, x_j$ so that the local relations are $\gamma^A(x_1, \ldots, x_j)$. For example, the reachability relation

$$Reach(x, y) \; :\Longleftrightarrow \; \text{there is a path from } x \text{ to } y$$

on directed graphs with edge relation E is a global relation with vocabulary $\{E\}$ and variables x, y.

To formalize properly the notion of global relation with variables, let V be a set of variables and A^V be the set of mappings from V to the base set $|A|$ of a structure A. Think about each member $\bar{a}$ of A^V as a set of elements of A indexed with V; if V is ordered then $\bar{a}$ can be viewed as a tuple. A relation R over $|A|$ with variables V can be viewed as a subset of A^V. An *isomorphism* η from (A, R) to (A', R') is an isomorphism from A to A' such that $R^A = \eta^{-1} R^{A'}$.

Thus, a global relation γ with vocabulary Υ and variables V can be defined as a function that assigns to every Υ-structure A a local relation $\gamma^A \subseteq A^V$. The abstractness requirement is as above: every isomorphism between Υ-structures A and A' is also an isomorphism between the expanded structures (A, γ^A) and $(A', \gamma^{A'})$.

A global relation γ with vocabulary Υ and variables V can be seen as a Boolean-valued function on V-pebbled Υ-structures; it also can be seen as a Boolean-valued function on V'-pebbled Υ-structures B where $V \subseteq V'$. Let φ be a formula of vocabulary Υ, and V be the set of free variables of φ. We define a global relation GR_φ (or

$GR(\varphi)$) with vocabulary Υ and variables V. If B is a V'-pebbled Υ-structure and $V \subseteq V'$, then

$$GR^B(\varphi) :\iff B \models \varphi$$

2.2 Functional First-Order Logic (FFO)

FFO is a version of first-order logic where formulas are special terms [Gurevich 1991, 1995]. RFO may be many-sorted but usually it is one-sorted. In contrast, FFO has at least two sorts (or types) because the two truth values form a separate type `Boole`. In applications, it is convenient to have many types; accordingly our FFO is in general multi-typed. Thus our RFO and FFO differ in two essentially orthogonal respects: RFO is untyped relational logic, and FFO is multi-typed functional logic. We proceed to define FFO more formally.

2.2.1 Syntax

A *vocabulary* consists of the following symbols:

- A collection of *types* (or, more exactly, *type symbols*).
- A collection of function symbols. Each function symbol comes with a *profile*

$$f : S_1 \times \cdots \times S_j \to T$$

  where $S_1, \ldots, S_j, T$ are type symbols. If $j = 0$, we write simply $f : T$.

The only obligatory type is `Boole`; a function symbol f with profile of the form $S_1 \times \cdots \times S_j \to$ `Boole` is a *relation symbol.* Other types could be for example `Integer, String, Vertex`.

To simplify things, we avoid polymorphism. So we required above that each function symbol has only one profile. Accordingly, in the case of equality, we assume that, for every type T, there is a special equality symbol $=_T$: $T \times T \to$ `Boole` and a special nullary symbol $\texttt{default}_T : T$; the subscript will be usually omitted. The other obligatory function symbols are nullary function symbols `true` and `false` of type `Boole` and the usual propositional connectives with the obvious profiles. The choice of "usual" connectives is arbitrary, but $\neg, \wedge, \vee$ should be included.

The version of FFO closest to RFO will be called the *minimal FFO.* This is the version of FFO where

- there is only one non-Boolean type (called `Domain`), and
- there are no Boolean variables and `Boole` does not appear in the left part of the profile $S_1 \times \cdots S_j \to T$ of any non-obligatory function symbol.

Remark In the case of minimal FFO, we need only one equality symbol and only one default symbol (usually called `undef`) because the Boolean equality and default will be definable as $\leftrightarrow$ and `false`, respectively. Here $\leftrightarrow$ can be a connective; alternatively $p \leftrightarrow q$ may abbreviate e.g. $(p \wedge q) \vee (\neg p \wedge \neg q)$. $\square$

There is an infinite supply of variables of every type, except that the minimal FFO has no Boolean variables. *Terms* are defined inductively. Every term is assigned a type. Terms of type `Boole` are called *formulas.*

- A variable of type T is a term of type T.
- If f is function symbol with profile $S_1 \times \cdots \times S_j \rightarrow T$ and $s_1, \ldots, s_j$ are terms of types $S_1, \ldots, S_j$ respectively, then $f(s_1, \ldots, s_j)$ is a term of type T.
- If $\varphi(x)$ is a formula and x is a variable, then $\exists x\, \varphi(x)$ and $\forall x\, \varphi(x)$ are formulas.

Remark This rudimentary type system differs from the type system used in the ASM literature (e.g. [Del Castillo, Gurevich and Strötmann 1998]) because, in the spirit of first-order logic, we don't use type constructors of positive arity here. □

2.2.2 Pebbled Structure Semantics

The notion of *structure* is generalized in the following way. A structure A is given by a set (the base set of A), interpretations of type symbols and interpretations of function symbols. It is assumed that the base set contains the symbols `true` and `false`. A type symbol T is interpreted as a nonempty subset T^A of the base set. A function symbol $f : S_1 \times \cdots \times S_j \rightarrow T$ is interpreted as a function from $S_1^A \times \cdots \times S_j^S$ to T^A. `Boole` is interpreted as the set that consists of the two truth values. The equality symbols, `true`, `false` and the propositional connectives are interpreted in the obvious way; `false` is the Boolean default.

In the obvious way, generalize the notion of pebbled structures and the notion that a term and a pebbled structure are appropriate for each other. By the obvious induction, a term τ of type T is given a value $Val^B(\tau) \in T^B$ in every pebbled structure B appropriate for τ.

2.2.3 Global Function Semantics

The notion of global relation generalizes obviously to the notion of global function (and to other global objects [Gurevich 1984, 1988]). A *global function* γ with vocabulary Υ, variables V and type T is a function that assigns to every Υ-structure A a local function γ^A with variables V and type T in such a way that every isomorphism between Υ-structures A and A' is also an isomorphism between the expanded structures (A, γ^A) and $(A', \gamma^{A'})$. For example, the reachability function

$$ReachFun(x, y) \;:=\; \begin{cases} \texttt{true} & \text{if there is a path from } x \text{ to } y \\ \texttt{false} & \text{otherwise} \end{cases}$$

on directed graphs with edge relation E is a global function with vocabulary $\{E\}$, type `Boole`, and variables x, y of type `Domain`

A global function γ with vocabulary Υ, type T and variables V can be seen as a function on V'-pebbled Υ-structures B where $V \subseteq V'$; γ^B is an element of T^B. Let τ be a term of vocabulary Υ and type T, and let V be the set of free variables of τ. The global function GF_τ (or $GF(\tau)$) is the global function computed by τ; its vocabulary is Υ, its type is T and its variables are V.

Part II
The Fixed-Choice Operator

3 Fixed-Choice Logic FO+ε

In this and the next section, FO is RFO. The base set of a structure A is denoted by $|A|$.

3.1 Syntax and Semantics

Syntax FO+ε is obtained from FO by adding the following term-formation rule:

- If $\varphi(x)$ is a formula, then $(\varepsilon x \varphi(x))$ is a term, called an ε-term.

The variable x is bound in $(\varepsilon x \varphi(x))$. The type of the ε-term $(\varepsilon x \varphi(x))$ is that of the variable x.

Semantics A *choice function* for a nonempty set S is a function F from 2^S (the power set of S) to S such that $F(X) \in X$ for all nonempty $X \subseteq S$. A *choice function* for a structure A is a choice function for $|A|$. An *ε-structure* is a pair (A, F) where A is a structure and F is a *choice function* for $|A|$. The *vocabulary* of (A, F) is that of A. Although RFO structures don't have default elements, an ε-structure effectively has the default element $F(\emptyset)$.

A *pebbled ε-structure* is a triple (A, F, ξ) where (A, F) is an ε-structure and ξ is a variable assignment over A; it is a V-pebbled ε-structure if V is the domain of ξ. A V-pebbled ε-structure provides values for terms with free variables in V. The definition of these values follows the standard inductive definition but has one additional clause:

- If t is an ε-term $(\varepsilon x \varphi(x))$, then

$$\begin{aligned} Val^{(A,F,\xi)}(t) &:= F\{a \in |A| : (A, F, \xi) \models \varphi(a)\} \\ &= F\{a \in |A| : (A, F, \xi(x \mapsto a)) \models \varphi(x)\} \end{aligned}$$

Question 3.1 Notice that an ε-structure (A, F) determines a unique well-ordered enumeration of $|A|$ with the following property: if α is an ordinal and $\{x_\beta : \beta < \alpha\}$ is a proper subset of $|A|$, then $x_\alpha := F(|A| - \{x_\beta : \beta < \alpha\})$. Call this ordering standard. Is the standard ordering uniformly definable on arbitrary structures? Is the standard ordering uniformly definable on finite structures? Is the last element of the standard ordering uniformly definable on finite structures? Is the existence of the last element in the standard ordering uniformly definable on arbitrary structures? Is any total ordering uniformly definable on arbitrary structures? Is any total ordering uniformly definable on finite structures? We expect all the answers to be negative, but the questions are open. By the way, the standard ordering is easily definable if FO+ε is augmented with the unary inflationary fixed-point operator [Ebbinghaus and Flum 1995, page 121].

3.2 ε-Invariant Sentences

To compare the expressive power of FO+ε with that of first-order logic, consider FO+ε sentences φ which are *ε-invariant* (or *deterministic*) in the following sense: if A is a structure of the vocabulary of φ and F_1, F_2 are two choice functions for A, then

$$(A, F_1) \models \varphi \iff (A, F_2) \models \varphi$$

Proposition 3.2 *Every invariant FO+ε sentence φ is equivalent to some first-order sentence ψ in the following sense: for every structure A of the vocabulary of φ and every choice function F for A, we have*

$$(A, F) \models \varphi \iff A \models \psi$$

The Proposition seems to be folklore; it is mentioned in [Caicedo 1995] without a reference. For reader's convenience, we provide a proof.

Proof Let Υ be the vocabulary of φ and let $<$ be a fresh binary relation. Construct first-order formulas $\alpha(<), \beta(<), \gamma(<)$ in vocabulary $\Upsilon \cup \{<\}$ such that

- $\alpha(<)$ asserts that $<$ is a linear order with a first element;
- $\beta(<)$ asserts that φ holds under the interpretation

$$\varepsilon(X) = \begin{cases} \min(X) & \text{if } X \neq \emptyset \text{ and } \min(X) \text{ exists,} \\ \min\{x : x = x\} & \text{otherwise;} \end{cases}$$

- $\gamma(<)$ asserts that every nonempty definable set X that occurs in the evaluation of φ according to the above interpretation has a minimal element.

Let $<'$ be another fresh binary symbol. The invariance of φ implies that the implication

$$\big(\alpha(<) \wedge \beta(<) \wedge \gamma(<)\big) \longrightarrow \big(\alpha(<') \wedge \gamma(<') \longrightarrow \beta(<')\big)$$

is valid. By the Craig Interpolation Theorem, there exists a first-order Υ-formula ψ such that the implications

$$\begin{aligned} \big(\alpha(<) \wedge \beta(<) \wedge \gamma(<)\big) \;&\longrightarrow\; \psi, \\ \psi \;&\longrightarrow\; \big(\alpha(<') \wedge \gamma(<') \longrightarrow \beta(<')\big) \end{aligned}$$

are valid. It is easy to see that ψ is equivalent to φ. $\square$

In the rest of this section, we restrict attention to finite structures. We will say that an FO+ε formula is ε-invariant if it is ε-invariant over finite structures. Similarly, equivalence of formulas will mean equivalence over finite structures. The proof of Proposition 3.2 does not apply in this situation because the interpolation theorem is not available [Ebbinghaus and Flum 1995, page 64].

Martin Otto exhibited an ε-invariant sentence that is not equivalent to any first-order sentence [Otto 1998]. Notice that every ε-invariant formula φ of any vocabulary Υ gives rise to a first-order formula φ^* of vocabulary $\Upsilon \cup \{<\}$ (where $<$ is a fresh binary relational symbol) which is order-invariant over finite structures in the following sense: if an Υ-structure A with a linear order $<_1$ satisfies φ^* then A with any other linear order $<_2$ satisfies φ^*. The desired φ^* asserts that φ holds under the interpretation

$$\varepsilon(X) = \begin{cases} \min(X) & \text{if } X \neq \emptyset \\ \min\{x : x = x\} & \text{otherwise} \end{cases}$$

Earlier, Gurevich found a nonelementary property of finite Boolean algebras (namely, "the number of atoms is even") that is expressible by an order-independent elementary formula γ; see [Ebbinghaus and Flum 1995, Proposition 2.5.6(a)].[3]

Question 3.3 Does there exist an order-invariant elementary formula that is not equivalent to any ε-invariant formula? In particular, is the property "the number of atoms is even" of finite Boolean algebras expressible by an ε-invariant formula?

We conjecture the negative answer for the second question (which implies the positive answer for the first question).

Proposition 3.4
1. The decision problem whether a given FO+ε formula φ is ε-invariant is undecidable. 2. There exists an FO+ε formula φ such that the following decision problem **INV**(φ) *is co-NP hard: Given a structure A of the vocabulary of φ, decide whether φ is ε-invariant over A.*

Parallel results for order-invariant sentences have been proved in [Gurevich 1988, pages 29–30]. The same proofs can be adapted here.

Proof
1. The decision problem whether a given first-order sentence α is true in all finite structures is undecidable [Trakhtenbrot 1950; Börger, Grädel and Gurevich 1996]. Therefore the decision problem whether a given first-order sentence α is true on all finite structures of cardinality ≥ 2 is undecidable. We reduce the latter problem to the problem in the proposition.

[3] Actually, Gurevich's original example was a collection of structures A with a subset *Atoms* of even cardinality and a binary relation E such that, for every $X \subseteq Atoms$ there exists a unique $y \in |A| - Atoms$ with $X = \{x \in Atoms : x\,E\,y\}$. The Boolean algebra reformulation is due to Ebbinghaus and Flum.

Let P be a unary relation symbol that does not occur in α. Set

$$\beta := P\Big(\varepsilon x\big(P(x) \vee \neg P(x)\big)\Big)$$

It is easy to see that $\alpha \vee \beta$ is ε-invariant over finite structures if and only if α is valid on finite structures of cardinality ≥ 2.

2. The problem whether a given graph has a Hamiltonian path is NP complete [Garey and Johnson 1979]. Therefore the problem whether a given graph of cardinality > 2 has no Hamiltonian path is co-NP complete. We reduce the latter problem to the problem in the proposition.

Construct an FO+ε sentence φ in the language of graphs that asserts that the binary relation

$$x \leq y :\iff x = \varepsilon z(x = z \vee y = z)$$

is a linear order and that every pair of neighbors in that order is adjacent. A graph G has no Hamiltonian path if and only if φ is ε-invariant over G. □

Remark Of course $(P(x) \vee \neg P(x))$ can be replaced with $x = x$. But if one is willing to use equality then one can get rid of P altogether. For example, β may assert the existence of an x such that $x = \varepsilon z(z = x \vee z = y)$ for all $y \neq x$. (There is a small price to pay: Replace ≥ 2 with ≥ 3 everywhere in the proof.) □

The first decision problem remains undecidable in the case of arbitrary (not only finite structures); the same proof is valid except that the reference to [Trakhtenbrot 1950] should be replaced with a reference to [Church 1936, Turing 1936]. However, in that case, there is a recursive set R of ε-invariant formulas such that every ε-invariant formula is equivalent (over all structures) to some formula in R. The desired R consists of formulas that do not mention ε at all; see Proposition 3.2. We return to finite structures. Does there exist a recursive set R of ε-invariant formulas such that every ε-invariant formula is equivalent to some formula in R? More generally:

Question 3.5 Does there exists a recursive syntax for properties expressible by ε-invariant formulas?

Question 3.5 requires explanations. Recall that we restrict attention to finite structures. We follow [Gurevich 1988]. A *logic* L is given by a pair of functions (Sen, Sat) satisfying the following conditions. Sen associates with every vocabulary Υ a recursive set whose elements are called *L-sentences of vocabulary* Υ. Sat associates with every vocabulary Υ a recursive relation $A \models \varphi$ where A ranges over Υ-structures and φ over L-sentences of vocabulary Υ; it is assumed of course that $A \models \varphi \iff B \models \varphi$ if A and B are isomorphic. Now we can formulate Question 3.5 precisely: Does there exist a logic L such that the properties expressible by L-sentences are exactly the properties expressible by ε-invariant sentences?

Remark Since the canonical order, derived from the ε-operator in Question 3.1, can be defined in the extension $FO+\varepsilon+IFP$ of FO+ε with the inflationary fixed-point operator, the following properties are equivalent, for any global relation:

- Ptime,
- Order-invariant $FO+<+IFP$ definable,
- ε-invariant $FO+\varepsilon+IFP$ definable.

Concerning the equivalence of the first two properties, see [Ebbinghaus and Flum 1995, page 150]. □

4 ε-Existential Formulas

In this section, FO still is relational first-order logic RFO.

Syntax An *ε-existential formula* Φ has the form $(\exists\,\varepsilon)\varphi$ where φ is a formula in FO+ε. The vocabulary $Voc(\Phi)$ of Φ is that of φ. The free individual variables of Φ are those of φ; the set of the free individual variables of Φ will be denoted $Var(\Phi)$.

Semantics Let Φ be an ε-existential formula $(\exists\,\varepsilon)\varphi$. Recall that models of φ have the form (A, F, ζ) where (A, ζ) is a pebbled structure and F is a choice function for A. Our intention is to quantify the choice function out. A pebbled structure (A, ζ) *models* φ if there exists a choice function F for A such that (A, F, ζ) models φ. $Mod(\Phi)$ is the collection of all models of Φ.

Σ^1_1 Formulas Recall that a Σ^1_1 formula Ψ is an existential second-order formula. The vocabulary $Voc(\Psi)$ of Ψ is the collection of free relation and function symbols of Ψ. $Mod(\Psi)$ is the collection of pebbled structures (A, ζ) such that $(A, \zeta) \models \Psi$.

We say that a Σ^1_1 formula Ψ is *equivalent* to an ε-existential formula Φ if $Mod(\Psi) = Mod(\Phi)$.

Theorem 4.1 *Every ε-existential formula $(\exists\,\varepsilon)\varphi(\bar{x})$ is equivalent to some Σ^1_1 formula.*

Notice that the relations that are existentially quantified in a Σ^1_1 formula are considerably smaller (in large structures) than the choice functions quantified in the semantics of an ε-existential formula. The point of the theorem is that, in the semantics of any particular ε-existential formula φ, only a small part of the choice function is used, and this part can be expressed with relations that a Σ^1_1 formula can quantify.

Proof For simplicity of exposition, we assume that φ has no free individual variables. List all ε-terms

$$\varepsilon x_0\chi_0(x_0, \bar{y}_0), \ldots, \varepsilon x_{r-1}\chi_{r-1}(x_{r-1}, \bar{y}_{r-1})$$

in φ in such a way that, for every $j < r$, all ε-subterms of $\varepsilon x_j \chi_j(x_j, \bar{y}_j)$ occur earlier in the list. Let $\{f_0, \ldots, f_{r-1}\}$ be fresh function symbols such that $arity(f_j) = length(\bar{y}_j)$ for all $j < r$.

Construct formulas

$$\begin{array}{lll} \varphi^i & \text{where} & 0 \le i \le r \\ \chi^i_j & \text{where} & 0 \le i \le j < r \end{array}$$

as follows.

$\varphi^0 = \varphi$ and every $\chi^0_j = \chi_j$. Notice that χ^0_0 is ε-free (and contains no f_j).

φ^1 (respectively χ^1_j where $j \ge 1$) is obtained from φ^0 (respectively χ^0_j) by replacing $\varepsilon x_0 \chi^0_0(x_0, \bar{y}_0)$ with $f_0(\bar{y}_0)$. Notice that χ^1_1 is ε-free (and contains no f_j with $j \ge 1$).

φ^2 (respectively χ^2_j where $j \ge 2$) is obtained from φ^1 (respectively χ^1_j) by replacing $\varepsilon x_1 \chi^1_1(x_1, \bar{y}_1)$ with $f_1(\bar{y}_1)$. Notice that χ^2_2 is ε-free (and contains no f_j with $j \ge 2$).

And so on.

φ^{r-1} (respectively χ^{r-1}_{r-1}) is obtained from φ^{r-2} (respectively χ^{r-2}_{r-1}) by replacing $\varepsilon x_{r-1} \chi^{r-2}_{r-2}(x_{r-2}, \bar{y}_{r-2})$ with $f_{r-2}(\bar{y}_{r-2})$. Notice that χ^{r-1}_{r-1} is ε-free (and does not contain f_{r-1}).

φ^r is obtained from φ^{r-1} by replacing $\varepsilon x_{r-1} \chi^{r-1}_{r-1}(x_{r-1}, \bar{y}_{r-1})$ with $f_{r-1}(\bar{y}_{r-1})$. Notice that φ^r is ε-free.

Now form the conjunction ψ of the following formulas:

- φ^r,
- **(Correlation)** If $i \le j < r$, and $\bar{y}$ is a tuple of fresh variables whose length equals the arity of f_i, and $\bar{z}$ is a tuple of fresh variables whose length equals the arity of f_j, then

$$\forall \bar{y} \forall \bar{z} \Big[\forall x \Big(\chi^i_i(x, \bar{y}) \leftrightarrow \chi^j_j(x, \bar{z})\Big) \to f_i(\bar{y}) = f_j(\bar{z})\Big]$$

- **(Witnessing)** If $i < r$ and $k = \mathit{arity}(f_i)$ and $\bar{y}$ is a k-tuple of fresh variables, then

$$\forall \bar{y} \Big[\exists x \chi^i_i(x, \bar{y}) \to \chi^i_i(f_i(\bar{y}), \bar{y})\Big]$$

We check that $(\exists\, \varepsilon)\psi$ is equivalent to $\exists f_0 \ldots \exists f_{r-1} \psi$.

Claim 4.2 *Suppose that A is a structure of the vocabulary of φ and F a choice function for A. Expand A by interpreting functions f_j as follows:*

$$f_j(\bar{y}) := \varepsilon x \chi_j(x, \bar{y})$$

for all $j < r$ and all tuples $\bar{y}$ of appropriate length. Let B be the expanded structure. Then

$$(A, F) \models \varphi \iff B \models \psi$$

The proof is obvious.

Claim 4.3 *Suppose that A is a structure of the vocabulary of φ and B is an expansion of A to the vocabulary of ψ such that B satisfies all the correlation and witnessing conditions. There exists a choice function F for A such that*

$$(A,F) \models \varphi \iff B \models \psi$$

Indeed, pick any choice function F for A satisfying the following conditions:

$$F(\{x : \chi_i^i(x,\bar{y})\}) = f_i(\bar{y})$$

for all $i < r$ and all tuples $\bar{y}$ of the appropriate length. The correlation conjuncts of ψ guarantee that the conditions do not contradict each other. The witnessing conjuncts of ψ guarantee that the conditions do not contradict the requirement that F be a choice function. Now it is easy to see that $(A,F) \models \varphi \iff B \models \psi$.

That concludes the proof of Theorem 4.1. □

Remark. The proof above may introduce more second-order quantifiers than necessary. If $j > i$ and $\chi_j(x_j, \bar{y}_j)$ is obtained from $\chi_i(x_i, \bar{y}_i)$ by renaming variables, the function symbol f_i can be used instead of f_j.

Recall that, for every FO+ε formula φ, **INV**(φ) is the following decision problem: Given a finite structure A of the vocabulary of φ, decide whether φ is ε-invariant over A.

Corollary 4.4 *Each* **INV**(φ) *is co-NP.*

Proof We use the results and notation of the proof of Theorem 4.1. Let A be a structure of the vocabulary of φ and let B range over expansions of A to the vocabulary of ψ. Call B *relevant* if it satisfies all the correlation and witnessing conditions. Relevance is a first-order condition and therefore can be checked in polynomial time.

By Claims 4.2 and 4.3, φ is not ε-invariant over A if and only if there exist two relevant expansions B_1, B_2 of A such that $B_1 \models \psi$ and $B_2 \models \neg\psi$. Thus the complementary decision problem is NP. □

This corollary and Proposition 3.4 imply that, for a certain φ, **INV**(φ) is co-NP complete.

Theorem 4.5 *Every Σ_1^1 formula is equivalent to some ε-existential formula.*

The idea here is that the relations quantified in a Σ_1^1 formula can be encoded in a choice function and decoded in a FO+ε definable way. The coding task is non-trivial mainly because the relations to be coded may have high arity whereas the choice function chooses only from subsets of the structure, not from sets of tuples.

Proof For notational simplicity, we assume that the given Σ_1^1 formula has no free individual variables.

First we consider the case when the given Σ^1_1 formula has the form $\exists P\varphi$ where P is a single relational symbol and φ is first-order. Let k be the arity of P and let α_k be an FO+ε sentence saying that the binary relation

$$x \le y :\Longleftrightarrow\ x = \varepsilon z(x = z \vee y = z)$$

is a linear order with an initial segment I of length $(k+1)(k^2+1)$. (In the finite case, the initial segment condition means simply that the structure contains at least that many elements. In the infinite case, the condition is more meaningful as an infinite ordered set may have no finite initial segment.)

The desired ε-existential formula has the form $(\exists\varepsilon)(\alpha_k \wedge \psi)$. To make the description of ψ easier to understand, we describe what it says about a structure A rather than writing it out syntactically. Split the initial segment I into $k+1$ blocks, each of size k^2+1, and regard each block as a $k\times k$ array S, written row after row, plus an extra element e. Any k-element subset of S determines a function $f : [1,\dots,k] \to [1,\dots,k]$ in the following way. List the k elements according to the order determined by α_k and then set $f(i)$ equal to the column number of the i^{th} element.

Associate with each k-tuple $\bar a = (a_1,\dots,a_k)$ from A a set $\bar a^*$ as follows. Since there are $k+1$ blocks, take the first one which is disjoint from $\{a_1,\dots,a_k\}$, say block $S\cup\{e\}$ (as above). The desired $\bar a^*$ consists of

1. $a_1,\dots,a_k$,
2. e,
3. k elements of S determining the function $f : [1,\dots,k] \to [1,\dots,k]$ such that each a_i is the $f(i)^{th}$ element of $\{a_1,\dots,a_k\}$ in the order defined by α_k.

Observe that we can define $\bar a$ from $\bar a^*$ using the order $\le$ defined by α_k. The main point here is that the block $S\cup\{e\}$ is the only block where $\bar a^*$ has $k+1$ elements. Further, $\{a_1,\dots,a_k\} = \bar a^* - (S\cup\{e\})$. The k elements of $\bar a^* \cap S$ uniquely define the coding f. Now, f allows us to reconstruct the tuple $\bar a$. (e is needed because otherwise $\bar a^*$ would have only k elements in block S and might have k elements in a different block.)

The decoding of $\bar a^*$ to get $\bar a$ can clearly be formalized in FO+ε (on ε-structures satisfying α_k); so can the encoding $\bar a \mapsto \bar a^*$.

The desired ψ is obtained from φ by replacing every occurrence of $P(\bar t)$ with

$$\varepsilon x(x \text{ is a member of } \bar t^*) \text{ is an } e \text{ (the extra element of a block)} \qquad (*)$$

Since $\bar a \to \bar a^*$ is one-to-one, any interpretation of P can be matched by an interpretation of ε such that $(*)$ agrees with P.

The same obviously applies even if $k = 1$; the part of ε involved in $(*)$ is ε applied to sets of size 3 [thanks to e], so that the task of coding a given P, as in $(*)$, doesn't interfere with the task of coding a linear order, as in α_k.

It remains to consider the case when the given Σ^1_1 formula has the form $\exists R_1 \dots \exists R_l\varphi$ where each R_i is a relational symbol and $l > 1$. Set k equal to 1 plus the maximal arity of relation symbols R_i. The desired ε-existential formula has the form $\alpha_{k,l} \wedge \psi$ where the conjuncts are as follows. $\alpha_{k,l}$ is similar to α_k except that it requires the existence of an initial segment of length $\geq \max\{l, (k+1)(k^2+1)\}$. The linear order defined by $\alpha_{k,l}$ allows us to code all relations R_i into a single k-ary relation P where the first argument codes i:

$$R_i(\bar{a}) \iff P(b_i, \bar{a}, \bar{c})$$

where b_i is the i^{th} element in the linear order defined by $\alpha_{k,l}$ and $\bar{c}$ is padding necessary in case the arity of P_i is less than $k-1$. The formula $\exists R_1 \ldots \exists R_l \varphi$ is equivalent to a formula $\exists P \varphi'$ for an appropriate φ'. The desired ψ is constructed as above. $\square$

Corollary 4.6 *For every ε-existential formula $(\exists\,\varepsilon)\varphi$ there exists an equivalent ε-existential formula $(\exists\,\varepsilon)\psi$ satisfying the following requirement.*

- *Every application of ε has the form*

$$\varepsilon x(x = t_1 \vee \cdots \vee x = t_j)$$

where x is not free in any t_i. .

Furthermore, if φ has at most v free variables in any ε-term, then $j \leq 2v+5$.

Proof Using the proof Theorem 4.1, translate $(\exists\,\varepsilon)\varphi$ into a Σ^1_1 formula $\exists f_1 \ldots \exists f_{r-1}\chi$. A j-ary function f can be represented by a $(j+1)$-ary relation R. Accordingly transform $\exists f_1 \ldots \exists f_{r-1}\chi$ into an equivalent Σ^1_1 formula $\exists R_1 \ldots \exists R_{r-1}\chi^*$. Finally, using the proof Theorem 4.5, translate $\exists R_1 \ldots \exists R_{r-1}\chi^*$ into an ε-existential formula $(\exists\,\varepsilon)\psi$, which will automatically have the desired form.

Now suppose that φ has at most v free variables in any ε-term. Then the second-order quantification in $\exists f_1 \ldots \exists f_{r-1}\chi$ involves only functions of arity $\leq v$. Hence the second-order quantification in Σ^1_1 formula $\exists R_1 \ldots \exists R_{r-1}\chi^*$ involves only relations of arity $\leq v+1$. These relations can be coded with one $(v+2)$-ary relation; set $k := v+2$. Then the sets $\bar{a}^*$ in the proof of Theorem 4.5 are of size $\leq 2k+1 = 2v+5$. $\square$

Vaughan Pratt asked whether, without worrying about the bound on j, Corollary 4.6 can be strengthened by requiring that that the terms t_i are first-order, so that $(\exists\,\varepsilon)\psi$ does not have nested ε-terms. We shall show that the answer is positive. Notice, by the way, that, according to our proof of Theorem 4.5, ε may occur in the terms t_i only because ε was used to define an order relation with a sufficiently long initial segment.

It is natural to call an r-ary relation R *irreflexive* if every tuple $(x_1, \ldots, x_r) \in R$ contains r distinct values. R is *symmetric* if all permutations of any tuple in R belong to R.

Lemma 4.7 *Every Σ^1_1 formula φ is logically equivalent to a Σ^1_1 formula ψ where second-order quantification is restricted to irreflexive symmetric relations. Furthermore, if r is the maximal arity of the quantified relation symbols in φ, then the maximal arity of the quantified relation symbols in ψ is* $\max\{r, 3\}$.

Proof Without loss of generality, we may restrict attention to structures with at least four elements.

We convert φ to the desired ψ in three steps. First, it is easy to restrict second-order quantification to irreflexive relations. Consider, for example, a formula $\varphi := \exists P\Phi(P)$ where the relation P is binary. Let Q be a fresh unary relation symbol, and R a fresh binary relation symbol to be interpreted as ranging over irreflexive relations. φ is equivalent to the formula $\exists Q\exists R\Psi$ where Ψ is obtained from Φ by replacing every atomic formula $P(t_1, t_2)$ with the formula

$$\big(t_1 = t_2 \wedge Q(t_1)\big) \vee \big(t_1 \neq t_2 \wedge R(t_1, t_2).\big)$$

Relations P of arity greater than 2 can be handled analogously.

In the second step, we start with a Σ^1_1 formula $\exists\vec{P}\Phi$ with quantification only over irreflexive relations (as produced by the first step) and convert it into an equivalent Σ^1_1 formula of the form

$$(\exists <)\, \exists\vec{R}\,(\Psi_0 \wedge \Psi_1),$$

where the relation variables in $\vec{R}$ range over irreflexive symmetric relations (but $<$ need not be symmetric). The formula Ψ_0 says that $<$ is a discrete linear order with a first element and a last element; here "discrete" means that every element but the first has an immediate predecessor and every element but the last has an immediate successor. (We shall use the notation $x+1$ for the immediate successor of x; similarly, we use $x-1$, $x+2$, etc. with the obvious meanings.) Each k-ary relation P quantified in the original formula is replaced by $k!$ k-ary relations R in the new formula. The intended interpretation is that each of these R's agrees with P when the k arguments are in one of the $k!$ possible orders (with respect to $<$); since each R is symmetric, there's no choice about what it does when its arguments are in any other order. More formally, we obtain Ψ_1 from Φ by replacing atomic subformulas according to the following scheme.

If P is unary, then just one R corresponds to it, and P is simply replaced with R. If P is binary, then two R's correspond to it; we call them R_{12} and R_{21}, labeling them by the two permutations of $\{1, 2\}$. Then $P(t_1, t_2)$ in Φ is replaced with

$$(t_1 < t_2 \wedge R_{12}(t_1, t_2)) \vee (t_2 < t_1 \wedge R_{21}(t_1, t_2))$$

in Ψ_1. Similarly, a ternary P corresponds to six R's, indexed by the permutations of $\{1, 2, 3\}$, and $P(t_1, t_2, t_3)$ is replaced by a disjunction of six clauses, one for each permutation; a typical one of the six clauses is

$$(t_3 < t_1 < t_2 \wedge R_{312}(t_1, t_2, t_3)).$$

Relation variables of higher arities are handled analogously. In the resulting formula, $\exists < \exists\vec{R}(\Psi_0 \wedge \Psi_1)$, each R is used only when its arguments are in one specific order relative to $<$, so it does no harm to the meaning of the formula to interpret the variables R as ranging only over symmetric (and irreflexive) relations.

For the third and final step, we show how to eliminate the one remaining non-symmetric (though irreflexive) second-order variable, $<$, in favor of symmetric ones.

Specifically, we define, on any discrete linear order with first and last elements, a symmetric irreflexive binary relation P and a symmetric irreflexive ternary relation Q such that $<$ is uniformly first-order definable from P and Q, i.e.,

$$x < y \Leftrightarrow \lambda(x, y)$$

where λ is a fixed first-order formula in the vocabulary $\{P, Q\}$. (Here "uniformly" and "fixed" mean that λ does not depend on the order we started with.) Once we do this, we can convert $\exists < \exists\vec{R}(\Psi_0 \wedge \Psi_1)$ into its final form by changing the initial quantifier from $\exists <$ to $\exists P\,\exists Q$ and then replacing every $t_1 < t_2$ in the body of the formula with $\lambda(t_1, t_2)$.

Given a discrete order $<$, we define P and Q by

- $P(x, y)$ means that x, y are neighbors.
- $Q(x, y, z)$ means that there is a permutation (u, v, w) of (x, y, z) such that $v = u + 1$ and $v < w$.

It remains to produce the definition λ of $<$ from P and Q. Temporarily, consider only elements not too near the first or last element in our order. As a first approximation to λ, consider

$$\lambda_1(x, y) :\Leftrightarrow \exists z\,(P(x, z) \wedge Q(x, y, z)).$$

If $x < y$ then $\lambda_1(x, y)$ holds, as we can take z to be the immediate predecessor of x. Unfortunately, $\lambda_1(x, y)$ also holds if $x = y + 1$ or $x = y + 2$. Let

$$\lambda_2(x, y) :\Leftrightarrow (\lambda_1(x, y) \wedge \neg P(x, y) \wedge \neg\exists z(P(x, z) \wedge P(z, y))).$$

This corrects the error in λ_1 but at the cost of introducing a new error; $\lambda_2(x, y)$ holds if and only if $x + 2 < y$. Next, let

$$\lambda_3(x, y) :\Leftrightarrow \exists z\,(\lambda_2(x, z) \wedge \neg\lambda_2(y, z)).$$

Now $\lambda_3(x, y)$ holds if and only if $x < y$, subject to our temporary assumption that x and y are not too near the ends of the ordering.

To remove this temporary assumption and thus to finally produce the desired λ, it suffices to define, in terms of P and Q, the left and right endpoints of the order, for then it is easy, using P, to define the other elements very near the endpoints and to add to λ_3 some clauses to make it agree with $<$ for these elements.

But defining the endpoints is also easy. They are the points that are P-related to only one other point rather than two. And Q lets us distinguish the left from the right endpoint; an endpoint is the left one if and only if Q holds of it, its (unique) neighbor (according to P), and the other endpoint. This completes the proof of the lemma. □

Theorem 4.8 *For every Σ^1_1 sentence φ there exists an equivalent ε-existential formula $(\exists\,\varepsilon)\psi$ satisfying the following requirement.*

- *Every application of* ε *has the form*

$$\varepsilon x(x = t_1 \vee \cdots \vee x = t_j)$$

where the terms t_i *are first-order and* x *is not free in any* t_i. .

Furthermore, if r *is the maximal arity of quantified relation symbols in* φ *and* $s = \max\{r, 3\}$ *then* $j \leq 2s + 1$.

Proof By the previous lemma, we may assume that all quantified relation symbols in φ range over irreflexive symmetric relations. Let r, s be as above. Let m be the number of quantified relation symbols in the given formula φ, so that φ has the form

$$\exists R_1 \cdots \exists R_m \Phi$$

where Φ is first-order. Let $n = (s+1)(m+s)$. Without loss of generality, we may restrict attention to structures with at least n elements.

The desired ψ has the form

$$\exists \varepsilon \exists u_1 \cdots \exists u_n (\Phi_0 \wedge \Phi_1)$$

Φ_0 asserts that the elements $u_1, \ldots, u_n$ are all distinct. Think about elements $u_1, \ldots, u_n$ as a sequence of length n. This sequence splits into $s+1$ blocks of length $m+s$. Φ_1 is obtained from Φ by replacing every atomic formula $R_i(X)$ with a formula $\alpha_i(\bar{x})$ which says the following. If X is the set of elements in the tuple $\bar{x}$, and if $v_1, \ldots, v_{m+s}$ is the first block that does not intersect X, then

$$\varepsilon(X \cup \{v_i, \ldots, v_{i+s}\}) = v_i$$

Notice that X contains at most s elements. So X can intersect at most s blocks. Thus indeed there is a first block $v_1, \ldots, v_{m+s}$ that does not intersect X. The set $Y := X \cup \{v_i, \ldots, v_{i+s}\}$ uniquely determines the set X and the index i. To determine X, find the unique block B that contains $s+1$ elements of Y; then $X = Y - B$. And i is the position in B of the first element of B that belongs to Y.

In the presence of distinct u_i's, as asserted by Φ_0, we can, using the preceding observations, encode any sequence $R_1, \ldots, R_m$ with $\Phi(R_1, \ldots, R_m)$ into a choice function ε; every $R_i(X)$ will be equivalent to the formula $\varepsilon(X \cup \{v_i, \ldots, v_{i+s}\}) = v_i$ with which we replaced it. Therefore φ is equivalent to $(\exists\, \varepsilon)\psi$.
□

Corollary 4.9 *For every* ε*-existential formula* $(\exists\, \varepsilon)\varphi$ *there exists an equivalent* ε*-existential formula* $(\exists\, \varepsilon)\psi$ *satisfying the following requirement.*

- *Every application of* ε *has the form*

$$\varepsilon x(x = t_1 \vee \cdots \vee x = t_j)$$

where the terms t_i *are first-order and* x *is not free in any* t_i.

Furthermore, if r is the maximum number of free variables in any epsilon-term in φ and $s = \max\{r+1, 3\}$ then $j \le 2s+1$.

Proof By Theorem 4.1, $(\exists\,\varepsilon)\varphi$ is equivalent to a Σ^1_1 formula χ. The proof of Theorem 4.1 guarantees the following: if r is the maximum number of free variables in any epsilon-term in φ then the maximal arity of existentially quantified relation symbols in χ is bounded by r plus one. The "plus one" comes from changing function symbols to relation symbols. Now apply Theorem 4.8. □

Part III

The Independent-Choice Operator

The logic of the independent-choice operator is unusual. Consider for example the proposition

$$\Big(\delta x(x = x)\Big) = \Big(\delta x(x = x)\Big).$$

Even though it has the form $t = t$, it is not necessarily true if the given structure has at least two elements. The two applications of the choice operator are independent and thus the two chosen elements may be equal and may be different. This gives rise to nondeterministic propositional logic studied in Section 5. Further, consider the term

$$\delta x P(x, y)$$

where P is a binary relation. The natural denotation of the term on a given structure is a multi-valued function of y. This gives rise to nondeterministic first-order logic studied in Section 6. Building on this foundation, we introduce and study in Section 7 the extension FO+δ of functional first-order logic with the independent-choice operator δ; we also say a few words on possible extensions of FO+δ. Section 7 is the central section of this part. Finally, in Section 8 we study an alternative independent-choice operator.

5 Nondeterministic Propositional Logic NPL

The fact that the truth values of some propositions, like $(\delta x(x = x)) = (\delta x(x = x))$, are unknown leads to a three-valued logic that happens to be known to logicians. It is called Kleene's strong three-valued logic [Kleene 1952; Section 64]. We do not presuppose that the reader is familiar with Kleene's logic. We are going to define our three-valued logic from scratch.

Remark 1. The third value of Kleene's strong three-valued logic reflects an unknown truth value. Another popular third value in the logic literature, e.g. [Rescher 1969], is "undefined". The purpose of "undefined" is to treat partial functions. There are also logics that have both "unknown" and "undefined"; see for example [Päppinghaus and Wirsing 1983], which is also motivated by computer science. In the ASM tradition, defaults are used to make functions total. Thus we do not need the "undefined".

2. Kleene's book [Kleene 1952] does not have all information that we need. In particular, unlike Kleene, we do not restrict attention to the traditional propositional connectives. □

5.1 Syntax and Semantics

Syntax Syntactically, NPL is similar to ordinary propositional logic (PL) except that propositional variables are split into two categories: deterministic and nondeterministic. In addition to propositional variables, there are propositional constants `true` and `false`.

Semantics A variable assignment ξ gives a particular truth value (`true` or `false`) to all deterministic propositional variables in the domain $Dom(\xi)$ of ξ. In the case of a nondeterministic propositional variable p in $Dom(\xi)$, only the range of possible values is specified. $Rng^\xi(p)$ can be any of the following three sets:

$$\begin{aligned} True &:= \{\mathtt{true}\}, \\ False &:= \{\mathtt{false}\}, \\ Both &:= \{\mathtt{true}, \mathtt{false}\}. \end{aligned}$$

For future use we order the truth values and the nonempty sets of truth values as follows:

$$\mathtt{false} < \mathtt{true}, \qquad False < Both < True$$

Define the ξ-range $Rng^\xi(p)$ of a deterministic propositional variable $p \in Dom(\xi)$ to be $\{\xi(p)\}$. A propositional formula $\varphi(p_1, \ldots, p_j)$ gets a ξ-range of possible values provided all variables of φ do. If $\varphi(p_1, \ldots, p_j)$ has no repeated nondeterministic variables, then

$$Rng^\xi(\varphi) = \{\varphi(v_1, \ldots, v_j) : v_i \in Rng^\xi(p_i)\}$$

where $\varphi(v_1, \ldots, v_j)$ is computed in PL. To deal with repeated nondeterministic variables, we stipulate that *there is no correlation between the values of different occurrences of the same nondeterministic variable.* For example, if p, p_1, p_2 are nondeterministic variables in the domain of ξ and $Rng^\xi(p) = Rng^\xi(p_1) = Rng^\xi(p_2) = Both$, then

$$Rng^\xi(p \vee \neg p) = Rng^\xi(p_1 \vee \neg p_2) = Both$$

Consider a formula $\varphi(\bar{p}, q_1, \ldots, q_k)$ with deterministic variables $\bar{p}$ and nondeterminisitic variables $q_1, \ldots, q_k$. Suppose that there are altogether n different occurrences of the variables $q_1, \ldots, q_k$; if the j^{th} of these n is occurrences is an occurrence of q_i, set $\theta(j) = i$. Let $r_1, \ldots, r_n$ be fresh nondeterministic variables. For every $j = 1, \ldots, n$, replace the j^{th} occurrence of variables $q_1, \ldots, q_k$ (which is an occurrence of $q_{\theta(j)}$) by r_j. This gives a formula $\chi(\bar{p}, r_1, \ldots, r_n)$, where the variables $r_1, \ldots, r_n$ do not have repeated occurrences. $\varphi(\bar{p}, q_1, \ldots, q_k)$ is obtained from $\chi(\bar{p}, r_1, \ldots, r_n)$ by a variable substitution $\langle r_j \mapsto q_{\theta(j)} : j = 1, \ldots, n\rangle$. Define

$$Rng^{\xi}(\varphi(\bar{p}, q_1, \ldots, q_k)) := Rng^{\xi'}(\chi(\bar{p}, r_1, \ldots, r_n))$$

where ξ' agrees with ξ on deterministic variables $\bar{p}$ and $Rng^{\xi'}(r_j) = Rng^{\xi}(q_{\theta(j)})$ for $j = 1, \ldots, n$.

5.2 Validity and Equivalence

Extend the Boolean connectives to NPL as follows. Let $S_1, S_2, \ldots$ range over $\{True, False, Both\}$. Then

$$\begin{array}{rcl} \neg S_1 & := & \{\neg s : s \in S_1\} \\ S_1 \wedge S_2 & := & \{s_1 \wedge s_2 : s_i \in S_i\} \\ S_1 \vee S_2 & := & \{s_1 \vee s_2 : s_i \in S_i\} \end{array}$$

and similarly for the other propositional connectives (if any). We have for example

$$\begin{array}{rcl} \neg Both & = & Both, \\ True \wedge Both & = & Both, \\ True \vee Both & = & True. \end{array}$$

This allows one to compute the ranges of formulas by structural induction. For example, if φ is $\chi \wedge \neg\chi$ and we know already that $Rng^{\xi}(\chi) = Both$ then

$$Rng^{\xi}(\varphi) = Both \wedge \neg(Both) = Both \wedge Both = Both$$

It is easy to see that the ranges defined in the preceding subsection satisfy the obvious recursion equations based on the present definition of the propositional connectives.

Lemma 5.1 *$True, False, Both$ together with the operations $\wedge, \vee, \neg$ form a distributive lattice with involution.*

Proof Straightforward. The involution swaps $True$ and $False$, and leaves $Both$ intact. □

A formula φ is *valid under a variable assignment* ξ if $Rng^{\xi}(\varphi) = True$. A formula φ is *valid* if it is valid under every assignment of truth values to its variables. Say that a connective is *essentially nullary* if it is nullary or else it has a positive arity but its result does not depend on the values of the variables.

Claim 5.2 *No formula without essentially nullary connectives or deterministic variables is valid.*

Proof By induction on a formula φ, check that the range of φ is $Both$ when every nondeterministic variable is assigned $Both$ □

Call two formulas *equivalent* if they have the same range under every assignment of truth values to their variables. Notice that $p \vee \neg p$ is not equivalent to $\mathtt{true}$. Call a formula φ *dull* if, for every nondeterministic variable p in φ, either all occurrences of p are positive or else all occurrences of p are negative.

Lemma 5.3 *Suppose that two Boolean formulas φ and χ are equivalent in the sense of ordinary propositional logic PL (that is if their nondeterministic variables are treated as deterministic).*

1. *If neither formula contains a repeated nondeterministic variable, then they are equivalent.*

2. *If φ has repeated nondeterministic variables but χ doesn't, then $Rng^\xi(\chi) \subseteq Rng^\xi(\varphi)$ for every assignment ξ of truth values to the variables.*

3. *If both formulas are dull, then they are equivalent.*

Proof 1. Induction on the number k of nondeterministic variables. The case $k = 0$ is obvious. Suppose that $k > 0$ and let p be one of the nondeterministic variables. The two formulas can be denoted $\varphi(p)$ and $\chi(p)$. Let ξ be an assignment of truth values to all variables in $\varphi(p)$ or $\chi(p)$. If $\xi(p) = True$, then

$$\begin{aligned} Rng^\xi(\varphi(p)) &= Rng^\xi(\varphi(\mathtt{true})) = && \text{(by the induction hypothesis)} \\ & \quad Rng^\xi(\chi(\mathtt{true})) = Rng^\xi(\chi(p)) \end{aligned}$$

The case of $False$ is similar. Suppose that $\xi(p) = Both$. Using the fact that p has no repeated nondeterministic variables, we have:

$$\begin{aligned} Rng^\xi(\varphi(p)) &= Rng^\xi(\varphi(\mathtt{true})) \cup Rng^\xi(\varphi(\mathtt{false})) \\ &= Rng^\xi(\chi(\mathtt{true})) \cup Rng^\xi(\chi(\mathtt{false})) \\ &= Rng^\xi(\chi(p)) \end{aligned}$$

2. Induction on the number of repeated nondeterministic variables. Let p be one of those variables. For notational simplicity, we consider only the case when φ has exactly two occurrences of p; the more general case will be obvious. The two formulas will be denoted $\varphi(p,p)$ and $\chi(p)$ respectively. More pedantically, introduce fresh nondeterministic variables p_1, p_2 and a formula $\varphi'(p_1, p_2)$ such that φ is $\varphi'(p,p)$ which is a more accurate notation than $\varphi(p,p)$. Let ξ be an assignment of truth values to all relevant variables. If $\xi(p) = True$, then

$$\begin{aligned} Rng^\xi(\varphi'(p,p)) &= Rng^\xi(\varphi'(\mathtt{true},\mathtt{true})) \supseteq && \text{(by the induction hypothesis)} \\ & \quad Rng^\xi(\chi(\mathtt{true})) = Rng^\xi(\chi(p)) \end{aligned}$$

The case of $False$ is similar. Suppose that $\xi(p) = Both$. Then

$$\begin{aligned} & Rng^\xi(\chi(p)) = \\ & Rng^\xi(\chi(\mathtt{true})) \cup Rng^\xi(\chi(\mathtt{false})) \subseteq && \text{(by the induction hypothesis)} \end{aligned}$$

$$Rng^\xi(\varphi(\texttt{true},\texttt{true})) \cup Rng^\xi(\varphi(\texttt{false},\texttt{false})) \subseteq$$
$$Rng^\xi(\varphi(\texttt{true},\texttt{true})) \cup Rng^\xi(\varphi(\texttt{false},\texttt{false})) \cup$$
$$Rng^\xi(\varphi(\texttt{true},\texttt{false})) \cup Rng^\xi(\varphi(\texttt{false},\texttt{true})) =$$
$$Rng^\xi(\varphi(p,p))$$

3. Assume that φ and χ are dull formulas, equivalent in ordinary PL. If some nondeterministic p is positive in one of φ or χ and negative in the other, then (still in ordinary PL) its truth value never influences the truth values of φ and χ. So we can replace p with a fresh variable in χ without damaging our assumptions, and if we show that φ is equivalent in NPL to the new χ then the same conclusion follows for the old χ.

So we may assume that each nondeterministic variable occurs only positively in both φ and χ or occurs only negatively in both. Now consider any assignment ξ of truth values in NPL to the variables of φ and χ. Let ξ^+ be the truth assignment in ordinary PL such that, for every variable p, we have:

$\xi^+(p) = \texttt{true}$ if $Rng^\xi(p) = True$ or else $Rng^\xi(p) = Both$ and p occurs positively in our formulas, and
$\xi^+(p) = \texttt{false}$ if $Rng^\xi(p) = False$ or else $Rng^\xi(p) = Both$ and p occurs negatively in our formulas.

Define ξ^- similarly, but reversing "positively" and "negatively". Because of the monotonicity of the connectives, $Rng^\xi(\varphi)$ consists of the PL-values of φ under ξ^+ and ξ^-. The same goes for χ. By assumption, the PL-values of φ and χ agree, under any truth assignments, in particular under ξ^+ and under ξ^-. Therefore, their NPL ranges under ξ also agree. □

Lemma 5.4 (Substitution) *Let $\varphi(p_1,\dots,p_n)$ result from $\psi(q_1,\dots,q_k)$ by simultaneously substituting $\theta_i(p_1,\dots,p_n)$ for q_i for all $i = 1,\dots,k$. Let ξ be any assignment to $p_1,\dots,p_n$, and let σ be the assignment to $q_1,\dots,q_k$ defined by*

$$Rng^\sigma(q_i) := Rng^\xi_i(\theta(p_1,\dots,p_n))$$

Then $Rng^\xi(\varphi(p_1,\dots,p_n)) = Rng^\sigma(\psi(q_1,\dots,q_k))$.

Proof by induction on ψ, using the fact (see the beginning of this subsection) that ranges can be computed in a compositional way. □

The lemma says that it doesn't matter whether the substitution (of θ_i for q_i) is done syntactically, changing ψ to φ, or semantically, changing ξ to σ.

5.3 Normal Forms

Is it true that every formula φ is equivalent to a formula χ built by means of $\wedge, \vee, \neg$ only? The matter is not obvious. Suppose for example that one of our connectives is (*if p then q else r*) which is PL-equivalent to $(p \wedge q) \vee (\neg p \wedge r)$. That equivalence fails in NPL. Indeed, consider the assignment ξ with $Rng^\xi(p) = Both$ and $Rng^\xi(q) = Rng^\xi(r) = True$, and let φ be $(if\ p\ then\ q\ else\ r)$. Then

$$Rng^\xi(\varphi) = \{(if\ s_1\ then\ s_2\ else\ s_3) : s_1 \in Both;\ s_2, s_3 \in True\} = True$$

whereas $Rng^\xi((p \wedge q) \vee (\neg p \wedge r)) = Both$. On the other hand, φ is equivalent to

$$(p \wedge q) \vee (\neg p \wedge r) \vee (q \wedge r).$$

This last observation gives rise to a conjecture that the desired χ is, in PL, a maximal (in an appropriate sense) disjunctive normal form of φ.

Lemma 5.5 *Given an arbitrary n-ary operation α over $\{\texttt{true}, \texttt{false}\}$, define an n-ary operation β over $\{True, False, Both\}$ as follows:*

$$\beta(S_1, \ldots, S_n) := \{\alpha(s_1, \ldots, s_n) :\ each\ s_i \in S_i\}$$

Then β is expressible in NPL by a formula $\chi(p_1, \ldots, p_n)$ in disjunctive normal form where each p_i is a nondeterministic variable.

Proof Variables p_i and their negations will be called *literals*. The term *clause* will be used to mean a conjunction of literals without any repeated variables; the empty clause means `true`. Let χ be the disjunction of all clauses C such that C implies $\alpha =$ `true`; the empty disjunction means `false`. Clearly, χ is a disjunctive normal form for α.

Now consider an arbitrary assignment ξ of $True, False$, or $Both$ to the range of every p_i. We check that ξ gives the same values to β and χ. A *specialization* of ξ assigns a truth value (`true` or `false`) to every occurrence of every variable p_i so that the truth value belongs to $Rng^\xi(p_i)$. A specialization σ is *coherent* if it assigns the same value to all occurrences of the same variable.

The ξ-value of β is the collection of σ-values of α (and therefore of χ) where σ ranges over coherent specializations of ξ. This is included in $Rng^\xi(\chi)$. (But $Rng^\xi(\chi)$ may be larger due to the possibility that, for some p_i with $Rng^\xi(p_i) = Both$, some occurrences of p_i are made true and the other occurrences of p_i are made false. See, for example, the discussion of if-then-else above.) It follows that $Rng^\xi(\chi) = Both$ if the ξ-value of β is $Both$.

Assume that the ξ-value of β is $True$, and let C be the clause composed from all variables p_i such that $Rng^\xi(p_i) = True$ and all negations $\neg p_i$ such that $Rng^\xi(p_i) = False$. Since all coherent specializations of ξ make α true, C implies that α is true. Hence C occurs in χ and therefore $Rng^\xi(\chi) = True$.

Assume that the ξ-value of β is $False$. We must show that all specializations of ξ, not only coherent ones, make χ false. Suppose, toward a contradiction, that some specialization σ makes χ true and therefore makes a particular clause C in χ true. Since each variable occurs at most once in C, there is a coherent specialization σ' that agrees with σ on C and therefore also makes χ true. But this is absurd; all coherent specializations make α false and therefore also make χ false. □

The following theorem shows that NPL is robust with respect to the choice of connectives.

Theorem 5.6 *In NPL, every formula φ is equivalent to a formula using only the connectives $\neg, \wedge, \vee$ (and propositional constants* $\mathtt{true}, \mathtt{false}$*).*

Proof If φ has no repeated variables and α is the operation over $\{\mathtt{true}, \mathtt{false}\}$ represented by φ in PL and β is defined as in the previous lemma, then φ represents β in NPL and thus the theorem follows from the lemma. In general, however, φ may have repeated variables.

Now consider the case that φ has repeated variables. The lemma can be applied to every connective used in φ. Transform all the connectives used in φ to disjunctive normal form. The result is equivalent to φ because NPL truth values can be defined compositionally, so they respect the composition operations by which φ is built from the connectives. $\square$

Corollary 5.7 *In NPL, every formula φ is equivalent to a formula in disjunctive normal form and to a formula in conjunctive normal form.*

Proof Without loss of generality, φ uses only the connectives $\neg, \wedge, \vee$.

So we have an NPL-equivalent form of φ using only negation, conjunction, and disjunction. It can be converted to disjunctive or conjunctive normal form by applying de Morgan's laws and the distributive laws. These laws are correct for NPL, by Lemma 5.3, parts 1 and 3, respectively. $\square$

6 Nondeterministic First-Order Logic NFO

The syntax of NFO is that of FFO except that function symbols are split into deterministic and nondeterministic. All obligatory function symbols (equality signs $=_T$, nullary default functions $\mathtt{default}_T$, the propositional constants $\mathtt{true}, \mathtt{false}$ and the propositional connectives) are deterministic.

The rest of this section is devoted mostly to the semantics of NFO. We give three semantics of NFO which are equivalent — so they are really three ways of viewing one semantics — and which are needed in different situations. But first we generalize the notion of structure. A *nondeterministic structure* A is like a structure except that every nondeterministic function symbol $f : S_1 \times \cdots \times S_j \to T$ is interpreted as a multiple-valued function f^A from $S_1^A \times \cdots \times S_j^A$ to T^A. It is supposed that the set $Rng^A(f(\bar{a}))$ of all possible values of $f(\bar{a})$ is not empty for any $\bar{a} \in S_1^A \times \cdots \times S_j^A$.

6.1 Global Function Semantics

A *nondeterministic global function* γ with vocabulary Υ, variables V and type T assigns to every nondeterministic Υ-structure A a multivalued function γ^A of variables V and type T; it is assumed that every isomorphism from A to A' is also an isomorphism from (A, γ^A) to $(A', \gamma^{A'})$. Often, variables V are given by a tuple $\bar{x}$. In this case, we may speak about the global function $\gamma(\bar{x})$ and local functions $\gamma^A(\bar{x})$.

Every term $\tau(\bar{x})$ with vocabulary Υ, type T and free variables $\bar{x}$ gives rise to a nondeterministic global function GF_τ (or $GF(\tau)$) with vocabulary Υ, type T and free variables $\bar{x}$. Instead of $GF_\tau(\bar{x})$, we often write $GF(\tau(\bar{x}))$. Similarly, if A is an Υ-structure and $\bar{a}$ is a tuple of elements of A substitutable for $\bar{x}$ (so that $\bar{a}$ has the right length and the elements of $\bar{a}$ have the right types), we may write $GF^A(\tau(\bar{a}))$ instead of $GF^A_\tau(\bar{a})$.

To evaluate $GF(\tau(\bar{x}))$ at A, just evaluate $\tau(\bar{x})$ in A. The result may depend on various choices and thus the evaluation procedure is nondeterministic. More exactly, the evaluation procedure is recursive. Let A be an Υ-structure and let $\bar{a}$ be a tuple of elements substitutable for $\bar{x}$. To compute $GF^A(\tau(\bar{a}))$ in A do the following.

- If $\tau(\bar{x})$ is a variable, then just produce the given value of the variable.

- If $\tau(\bar{x})$ is $f(t_1, \ldots, t_j)$, then first compute every $GF^A(t_i(\bar{a}))$. Let the results be $b_1, \ldots, b_j$. Second compute $f(b_1, \ldots, b_j)$. The result may be any possible value in $Rng^A(f(b_1, \ldots, b_j))$.

- If $\tau(\bar{x})$ is $\exists y \varphi(\bar{x}, y)$, let b range over the elements of the type of y. First, compute $GF^A(\varphi(\bar{a}, b))$ for every b; the results form a subset Z of $\{\texttt{true}, \texttt{false}\}$. Second, find the maximal value in Z according to the order $\texttt{false} < \texttt{true}$.

  Notice that you compute $GF^A(\varphi(\bar{a}, b))$ just once for every given b and that these computations are independent. If $\varphi(\bar{x}, y)$ contains nondeterministic functions, the computation of $GF^A(\varphi(\bar{a}, b))$ is nondeterministic, and therefore the computation of $GF^A(\tau(\bar{a}))$ is nondeterministic.

- The case of $\forall y \varphi(\bar{x}, y)$ is similar to the case of $\exists y \varphi(\bar{x}, y)$, except that the minimal (rather than the maximal) value in Z is produced.

6.2 Pebbled Structure Semantics

A *pebbled nondeterministic structure* is a pair $B = (A, \xi)$ where A is a nondeterministic structure and ξ a variable assignment over A. Again, we write $B(x)$ instead of $\xi(x)$; it is assumed of course that $B(x) \in T^B$ if the type of x is T. Furthermore, if $\bar{x} = (x_1, \ldots, x_j)$, we may write $B(\bar{x})$ instead $(B(x_1), \ldots, B(x_j))$.

By induction on terms, define the range $Rng^B(\tau)$ of a term τ in a pebbled nondeterministic structure B appropriate for τ. Note that the ranges are never empty.

- If x is a variable then $Rng^B(x) = \{B(x)\}$.

- $Rng^B(f(t_1, \ldots, t_j)) := \bigcup\{Rng^B(f(a_1, \ldots, a_j)) : a_i \in Rng^B(t_i)\}$

- If x is variable of type T and $\varphi(x)$ is a formula, then

$$\begin{aligned} Rng^B(\exists x \varphi(x)) &:= \max\{Rng^B(\varphi(a)) : a \in T^B\} \\ Rng^B(\forall x \varphi(x)) &:= \min\{Rng^B(\varphi(a)) : a \in T^B\} \end{aligned}$$

  where max and min are taken with respect to the order $False < Both < True$.

Here (and in the rest of the paper) $Rng^B(\varphi(a))$ abbreviates $Rng^{B[x:=a]}(\varphi)$.

Lemma 6.1 (*Rng*/*GF* Lemma) *Suppose that B is the pebbled extension of A with $B(\bar{x}) = \bar{a}$. Then $Rng^B(\tau(\bar{x}))$ is exactly the set of all possible values of $GF_\tau^A(\bar{a})$.*

Proof Obvious. □

Notice that $Rng(\tau)$ also can be seen as a global function of a sort. It takes a *pebbled* structure B as input and produces a set of elements of B as output. $Rng(\tau)$ can be seen as a nondeterministic element or a nondeterministic nullary function. If one takes this point of view, the evaluation procedure for $GF(\tau)$ gives rise to the following procedure of sampling $Rng(\tau)$.

Let B be a pebbled structure appropriate for τ. To sample $Rng^B(\tau)$ do the following.

- If τ is a variable then produce the "pebble" of τ.
- If τ is $f(t_1, \ldots, t_j)$, then first sample every $Rng^B(t_i)$. Let the results be $b_1, \ldots, b_j$. Then sample the range of f at $(b_1, \ldots, b_j)$ in B.
- If τ is $\exists x \varphi(x)$, let a range over the elements of the type of x. For every a, sample $Rng^B(\varphi(a))$. Then take the maximal value.
- If τ is $\forall x \varphi(x)$, let a range over the elements of the type of x. For every a, sample $Rng^B(\varphi(a))$. Then take the minimal value.

6.3 Global Function-Sets

Function-Sets A *global function-set* Γ with vocabulary Υ, variables V and type T assigns to every nondeterministic Υ-structure A a set Γ^A of deterministic functions ζ of variables V and type T; it is assumed that if A, A' are isomorphic then we have the following: For each $\zeta \in \Gamma^A$ there is $\zeta' \in \Gamma^{A'}$ such that $(A, \zeta), (A', \zeta')$ are isomorphic by the same isomorphism, and *vice versa*.

Folding and Unfolding Consider a vocabulary Υ, type T and variables $x_1, \ldots, x_j$ of types $S_1, \ldots, S_j$ respectively. Let A range over Υ-structures, and let $\bar{a}$ range over $S_1^A \times \cdots \times S_j^A$, and let ζ range over functions from $S_1^A \times \cdots \times S_j^A$ to T^A. If Γ is a global function-set with vocabulary Υ, variables $x_1, \ldots, x_j$ and type T, then the *folding* of Γ is the nondeterministic global function γ with vocabulary Υ, variables V and type T such that

$$\text{the set of possible values of } \gamma^A(\bar{a}) \text{ is } \{\zeta(\bar{a}) : \zeta \in \Gamma^A(\bar{a})\}.$$

If γ is a nondeterministic global function with vocabulary Υ, variables $x_1, \ldots, x_j$ and type T, then the *unfolding* of γ is the global function-set Γ with vocabulary Υ, variables $x_1, \ldots, x_j$ and type T such that

$$\Gamma^A = \{\zeta : \text{ for every } \bar{a},\ \zeta(\bar{a}) \text{ is a possible value of } \gamma^A(\bar{a})\}.$$

If you unfold a given nondeterministic global function γ and then fold the resulting global function-set, you get back the original nondeterministic global function γ. If you fold a given global function-set Γ and then unfold the resulting nondeterministic global function, you get a global function-set $\bar{\Gamma}$ such that $\Gamma^A \subseteq \bar{\Gamma}^A$ for all A. We will say that $\bar{\Gamma}$ is the *closure* of Γ and that Γ is *closed* if $\Gamma = \bar{\Gamma}$.

Lemma 6.2 (Mixing Lemma) *Suppose that Γ is a closed global function-set and let $\zeta_1, \zeta_2 \in \Gamma^A$. If ζ_3 is a function such that every $\zeta_3(\bar{a})$ equals either $\zeta_1(\bar{a})$ or $\zeta_2(\bar{a})$, then $\zeta_3 \in \Gamma^A$.*

Proof Obvious. □

Let Γ and Δ be global function-sets with the same vocabulary Υ, same type and same variables. Call Γ and Δ *similar* if they have the same folding.

Lemma 6.3 *If Γ and Δ are similar and Δ is closed, then for every Υ-structure A, $\Gamma^A \subseteq \Delta^A$.*

Proof Obvious. □

6.4 Global Function-Set Semantics

A term $\tau(x_1 \ldots, x_j)$ with vocabulary Υ, type T and free variables $x_1, \ldots, x_j$ of types $S_1, \ldots, S_j$ respectively gives rise to a global function-set GFS_τ (or $GFS(\tau)$) with vocabulary Υ, variables $x_1, \ldots, x_j$ and type T. A member ζ of $GFS(\tau)$ is obtained by evaluating τ once at every value of the tuple $(x_1, \ldots, x_j)$. To be more specific, let A range over Υ-structures, let $\bar{a}$ range over $S_1^A \times \cdots \times S_j^A$, and let ζ range over functions from $S_1^A \times \cdots \times S_j^A$ to T^A. Evaluate $\tau(\bar{a})$ once for every $\bar{a}$. This way you get one function ζ. GFS_τ^A is the set of all functions ζ obtainable in this way (by making different nondeterministic choices). For example, if τ is $g(f(x))$, and $0, 1$ are the only values for x, and $2, 3$ are possible values for $f(0), f(1)$ respectively, and $4, 5$ are possible values for $g(2), g(3)$ respectively, then the function $\langle 0 \mapsto 4, 1 \mapsto 5 \rangle$ is a member of GFS_τ^A.

Lemma 6.4 (GF/GFS Lemma)

- *$GF(\tau)$ is the folding of $GFS(\tau)$.*
- *$GFS(\tau)$ is the unfolding of $GF(\tau)$.*
- *$GFS(\tau)$ is closed.*

Proof The third claim follows from the second. The first two claims follow from the definitions of $GF(\tau)$ and $GFS(\tau)$. □

6.5 Equivalent Terms

Call terms s, t *equivalent* (symbolically $s \iff t$) if $Rng^B(s) = Rng^B(t)$ for every pebbled structure B appropriate for both of them. Note that this agrees with equivalence of formulas, as defined in Section 5.

Lemma 6.5 (Equivalence Lemma)

- *Two terms s and t are equivalent if and only if $GF(s) = GF(t)$.*
- *Two terms s and t are equivalent if and only if $GFS(s) = GFS(t)$.*
- $\varphi \wedge \chi \iff \neg(\neg\varphi \vee \neg\chi)$
- $\forall x \varphi(x) \iff \neg\exists x \neg\varphi(x)$

Proof Obvious. □

Concerning the third claim above, we can also refer to NPL.

7 Independent-Choice Logic FO+δ

7.1 Syntax

Extend functional first-order logic with the following construct.

- If x is a variable of type T and φ is a formula, then $\delta x \varphi$ is a term of type T. All occurrences of x in $\delta x \varphi$ are bound. For any other variable y, every free (respectively bound) occurrence of y in φ remains free (respectively bound) in $\delta x \varphi$.

Often we write $\delta x \varphi(x)$, rather than $\delta x \varphi$, even though it is not required that x occurs free in $\varphi(x)$.

7.2 Semantics

FO+δ terms are interpreted on ordinary, rather than nondeterministic, structures. Nevertheless the semantics of FO+δ is similar to that of NFO because of the intrinsic nondeterminism of the δ operator.

7.2.1 Global Function Semantics

The definitions of global functions in Section 6 remain valid except that now we restrict attention to ordinary structures. The recursive procedure for evaluating $GF(\tau(\bar{x}))$ generalizes readily to FO+δ; we need only add the following clause. Recall that A is an Υ-structure and $\bar{a}$ is a tuple of elements substitutable for $\bar{x}$, and that our goal is to compute $GF^A(\tau(\bar{a}))$.

- If $\tau(\bar{x})$ is $\delta y \varphi(\bar{x}, y)$ and T is the type of y, let b range over T^A. Evaluate $GF^A(\varphi(\bar{a}, b))$ for each b. If there are elements b with $GF^A(\varphi(\bar{a}, b)) = \texttt{true}$, then choose nondeterministically one such b as the result; otherwise, output $\texttt{default}(T^B)$ (that is $\texttt{default}_T^B$).

Remark Evaluate each $GF^A(\varphi(\bar{a}, b))$. These evaluations are independent (that is, choices made in one evaluation are irrelevant for the others) and can be performed concurrently. Alternatively, you can choose arbitrarily $b_1 \in T^B$ and evaluate $GF^A(\varphi(\bar{a}, b_1))$. If it evaluates to `true`, then output b_1. Otherwise choose arbitrarily $b_2 \in T^B$ and evaluate $GF^A(\varphi(\bar{a}, b_2))$. If it evaluates to `true`, then output b_2. Otherwise choose arbitrarily $b_3 \in T^B$ and so on. If all $GF^A(\varphi(\bar{a}, b))$ evaluate to `false`, then output $\texttt{default}(T^B)$. □

7.2.2 Pebbled Structure Semantics

The inductive definition of $Rng(\tau)$ of Section 6 generalizes readily to FO+δ; we need only add the following clause.

- $$\begin{aligned} Rng^B(\delta x \varphi(x)) &:= X \cup Y \quad \text{where} \\ X &= \{a \in T^B : Rng^B(\varphi(a)) > False\} \\ Y &= \begin{cases} \{\texttt{default}(T^B)\} & \text{if every } Rng^B(\varphi(a)) < True \\ \emptyset & \text{if some } Rng^B(\varphi(a)) = True \end{cases} \end{aligned}$$

In other words, if it is possible to evaluate $\varphi(a)$ as true, then this a is a possible value of $\delta x \varphi(x)$. In addition, if it is possible to evaluate $\varphi(a)$ as false for all a, then $\texttt{default}(T^B)$ is a possible value of $\delta x \varphi(x)$.

The Rng/GF Lemma remains true.

Lemma 7.1 (Rng/GF Lemma) *The set of possible values of $GF_\tau^A(a_1, \ldots, a_j)$ is equal to $Rng^B(\tau)$ where B is the pebbled expansion of A with $B(x_i) = a_i$ for $i = 1, \ldots, j$.*

The recursive sampling procedure acquires the following clause. Recall that B is a pebbled structure appropriate for τ and that our goal is to sample $Rng^B(\tau)$.

- If τ is $\delta x \varphi(x)$ and T is the type of x, let a range over T^B. For each a, independently sample $Rng^B(\varphi(a))$. If, for some a, the result is `true`, choose nondeterministically one such a as the result; otherwise, output $\texttt{default}_T^B$.

7.2.3 Global Function and Global Function-Set Semantics

The definition of global function-sets in Section 6 remains valid except that now we restrict attention to ordinary structures. The GF/GFS Lemma remains true.

Lemma 7.2 (GF/GFS Lemma)

- *$GF(\tau)$ is the folding of $GFS(\tau)$.*
- *$GFS(\tau)$ is the unfolding of $GF(\tau)$.*
- *$GFS(\tau)$ is closed.*

7.3 Equivalent Terms

Call terms s, t *equivalent* if $Rng^B(s) = Rng^B(t)$ for every pebbled structure B appropriate for both of them.

Lemma 7.3

- *Two terms s and t are equivalent if and only if $GF(s) = GF(t)$.*
- *Two terms s and t are equivalent if and only if $GFS(s) = GFS(t)$.*
- $\varphi \wedge \chi$ *is equivalent to* $\neg(\neg\varphi \vee \neg\chi)$.
- $\forall x\varphi(x)$ *is equivalent to* $\neg\exists x\neg\varphi(x)$.

Proof Obvious. □

7.4 First-Order Expressibility of Ranges

Theorem 7.4 *Let τ be an FO + δ term, and let y be a fresh individual variable of the type of τ. There exists an FO formula $\tilde{\tau}(y)$ expressing that $y \in Rng(\tau)$. The free variables of $\tilde{\tau}(y)$ are those of τ plus y.*

Proof Induction on τ.

- If τ is a variable, then $\tilde{\tau}$ is $(y = \tau)$.
- If τ is $f(t_1, \ldots, t_j)$ and $x_1, \ldots, x_j$ are fresh variables, then

$$\tilde{\tau} := (\exists x_1, \ldots, x_j)\Big[y = f(x_1, \ldots, x_j) \wedge \bigwedge_i \tilde{t}_i(x_i)\Big]$$

- Suppose that τ is $\exists x\varphi(x)$. By the induction hypothesis, there exists a first-order formula $\tilde{\varphi}(x, y)$ that expresses that $y \in Rng(\varphi(x))$. Set

$$\tilde{\tau} := \Big(y = \texttt{true} \wedge \exists x\tilde{\varphi}(x, \texttt{true})\Big) \vee \Big(y = \texttt{false} \wedge \forall x\tilde{\varphi}(x, \texttt{false})\Big)$$

- If τ is $\forall x\varphi(x)$, then

$$\tilde{\tau} := \Big(y = \texttt{true} \wedge \forall x\tilde{\varphi}(x, \texttt{true})\Big) \vee \Big(y = \texttt{false} \wedge \exists x\tilde{\varphi}(x, \texttt{false})\Big)$$

- If τ is $\delta x \varphi(x)$, then

$$\tilde{\tau} := \tilde{\varphi}(y, \mathtt{true}) \vee \big(y = \mathtt{default} \wedge \forall x \tilde{\varphi}(x, \mathtt{false})\big)$$

□

Corollary 7.5 (Normal Form) *Every term t is equivalent to a term $\delta x \varphi(x)$ where $\varphi(x)$ is first-order.*

Proof The desired $\varphi(x)$ asserts that x belongs to $Rng(t)$. □

Remark One may wonder whether FO+ε has the similar property, that is whether every ε-term t is equivalent to an ε-term of the form $\varepsilon x \varphi(x)$ where $\varphi(x)$ is first-order? The answer is negative. The following term t is a counter-example:

$$\varepsilon x P\left(x, \varepsilon y Q(y)\right)$$

Indeed assume, by contradiction, that t is equivalent to some term $\varepsilon x \varphi(x)$. Consider an ε-structure (A, F) where $|A| = \{a, a', b, b'\}$, $Q = \{a, b\}$, $P = \{(a, a'), (b, b')\}$ and $F(Q) = a$, so that t evaluates to a' on (A, F). Let $X = \{x : \varphi(x)\}$ at A. Since $\varepsilon x \varphi(x)$ is equivalent to t, we have that $F(X) = a'$. It follows that either $a' \in X$ or $X = \emptyset$ and thus in either case $X \neq Q$. Let G be a choice function for A such that $G(Q) = b$ and $G(X) = F(X) = a'$. In (A, G), t evaluates to b', but $\varepsilon x \varphi(x)$ still evaluates to a'. This gives the desired contradiction. □

An FO+δ formula with range *Both* on some structures cannot be equivalent to a first-order formula because no first-order formula has range *Both* on any structure. But this is the only restriction on simulating FO+δ formulas by means of first-order formulas.

Corollary 7.6 *Let $\varphi(\bar{x})$ be an FO+δ formula and let B range over pebbled structures appropriate for φ. There are first-order formulas $\chi_1(\bar{x})$ and $\chi_2(\bar{x})$ such that, for every B,*

$$\begin{aligned} Rng^B(\varphi(\bar{x})) > False &\iff B \models \chi_1(\bar{x}) \\ Rng^B(\varphi(\bar{x})) = True &\iff B \models \chi_2(\bar{x}) \end{aligned}$$

Proof By the previous theorem, there exists a first-order formula $\tilde{\varphi}(\bar{x}, y)$, where y is a Boolean variable, such that, for every B, we have

$$\begin{aligned} \mathtt{true} \in Rng^B(\varphi(\bar{x})) &\iff B \models \tilde{\varphi}(\bar{x}, \mathtt{true}) \\ \mathtt{false} \in Rng^B(\varphi(\bar{x})) &\iff B \models \tilde{\varphi}(\bar{x}, \mathtt{false}) \end{aligned}$$

Set

$$\begin{aligned} \chi_1(\bar{x}) &:= \tilde{\varphi}(\bar{x}, \mathtt{true}) \\ \chi_2(\bar{x}) &:= \neg\tilde{\varphi}(\bar{x}, \mathtt{false}) \end{aligned}$$

□

Remark By analogy with ε-existential formulas, one may want to introduce δ-existential formulas $(\exists\delta)\varphi(\bar{x})$ where $\varphi(\bar{x})$ is an FO+δ formula. It is natural to require that $(\exists\delta)\varphi(\bar{x})$ is true on a pebbled structure (A, ζ) if the independent choices in $\varphi(\bar{x})$ can be performed in such a way over (A, ζ) that $\varphi(\bar{x})$ evaluates to true. In the notation of Corollary 7.6, $(\exists\delta)\varphi(\bar{x})$ is equivalent to $\chi_1(\bar{x})$, and thus δ-existential logic is no more expressive than first-order logic. □

An FO+δ formula φ is *δ-invariant* (or *deterministic*) over a class K of pebbled structures appropriate for φ if φ never has value *Both* over K, so that if $B \in K$ then all evaluations of φ in B give the same value for φ.

Corollary 7.7 *If φ is deterministic on K, then it is equivalent to a first-order formula over K.*

Proof Let χ_1, χ_2 be as in the previous proof. Both χ_1 and χ_2 are equivalent to φ. □

Remark on the power of δ. The power of δ becomes apparent in dynamic situations where, for example, the choice operator may apply to commands rather than formulas. Consider, for instance, the basic single-source shortest paths algorithm that operates on a weighted directed graph with a distinguished vertex *source*. We assume that the input is given by:

- a set of vertices with a distinguished vertex *source*,
- a set of edges,
- functions $e.1$ and $e.2$ which give the first and the second vertex of the given edge e respectively, and
- the *length* function $\ell(e)$ from the edges to non-negative reals.

The algorithm computes, for every vertex v, the shortest distance $dist(v)$ from *source* to v as well as a shortest path from *source* to v. The shortest path will be given by a predecessor function *pred*; its elements will be (in the reverse order) $v, pred(v), pred^2(v), \ldots, source$. Initially, $dist(source) = 0$, $dist(v) := \infty$ for all $v \neq s$, and $pred(v) := v$. In the traditional ASM notation, we have:

```
if ∀e : dist(e.2) ≤ dist(e.1) + ℓ(e) then
   Mode := Final
else
   choose e : dist(e.2) > dist(e.1) + ℓ(e)
      dist(e.2) := dist(e.1) + ℓ(e)
      pred(e.2) := e.1
   endchoose
endif
```

□

Remark on first-order expressibility of ranges in the case of FO+ε. The notion of range makes sense in the case of FO+ε but the ranges are not necessarily first-order expressible there, even if one restricts attention to finite structures. This follows from [Otto 1998]. Let us elaborate on that, restricting attention to sentences and finite structures. Define the range $Rng^A(\varphi)$ of an FO+ε sentence φ in a finite structure A as the set of truth values of φ in ε-structures (A, F) where F ranges over all choice functions for the base set of A. Say that $Rng(\varphi)$ is first-order expressible on finite structures if there exists a first-order sentence $\chi(p)$ with a propositional variable p such that, for every finite structure A of sufficiently rich vocabulary,

$$\begin{aligned} A \models \chi(\texttt{true}) &\iff \texttt{true} \in Rng^A(\varphi) \\ A \models \chi(\texttt{false}) &\iff \texttt{false} \in Rng^A(\varphi) \end{aligned}$$

Equivalently, but avoiding the use propositional variables, define $Rng(\varphi)$ to be first-order expressible on finite structures if there exist first-order sentences χ, ψ such that, for every finite structure A of sufficiently rich vocabulary,

$$\begin{aligned} A \models \chi &\iff \texttt{true} \in Rng^A(\varphi) \\ A \models \psi &\iff \texttt{false} \in Rng^A(\varphi) \end{aligned}$$

Now let φ be the FO+ε sentence from [Otto 1998] which is ε-invariant on finite structures but not equivalent to any first-order sentence on finite structures. If χ, ψ witnessed the first-order expressibility of $Rng(\varphi)$, then φ would be equivalent to χ on finite structures in the sense defined in Section 3. $\square$

7.5 The `let` Construct

Consider a term τ which has several occurrences of a term s. For example, τ may be $s = s$. Different occurrences of s may evaluate to different values. For example, the term $s = s$ may evaluate to `false`. Is there a way to guarantee that all occurrences of s in τ evaluate to the same value? The construct `let` allows us to do that.

7.5.1 Syntax

Extend FO+δ with the following term-formation rule:

- If x is a variable of type S, and s is a term of type S in which x does not occur free, and $t(x)$ is a term of type T, then

  $$\texttt{let } x \texttt{ be } s \texttt{ in } t(x)$$

  is a term of type T. All occurrences of x in the new term are bound. For any other variable y, all free (respectively bound) occurrences of y in s or $t(x)$ remain free (respectively bound).

7.5.2 Semantics

Extend the recursive definition of the range of a term with the following clause.

$$Rng^B(\texttt{let } x \texttt{ be } s \texttt{ in } t(x)) \;:=\; \bigcup\{Rng^B(t(a)) : a \in Rng^B(s)\}$$

Notice that ($\texttt{let}\ x\ \texttt{be}\ s\ \texttt{in}\ t(x)$) may not mean the same as $t(s)$ even if s is substitutable for x in $t(x)$. The reason is that in ($\texttt{let}\ x\ \texttt{be}\ s\ \texttt{in}\ t(x)$) a single evaluation of s provides the value for all occurrences of x in $t(x)$, whereas in $t(s)$ each occurrence of s is evaluated independently.

7.5.3 The Elimination of `let`

Surprisingly `let` can be eliminated.

Theorem 7.8 *The term* ($\texttt{let}\ x\ \texttt{be}\ s\ \texttt{in}\ t(x)$) *is equivalent to the term*

$$\delta y\Big[\exists x\Big(\varphi(x) \wedge \chi(x,y)\Big)\Big]$$

where $\varphi(x)$ is a first-order formula expressing that $x \in Rng(s)$, and y is a fresh variable, and $\chi(x,y)$ is a first-order formula expressing that $y \in Rng(t(x))$.

Proof Let λ be the term ($\texttt{let}\ x\ \texttt{be}\ s\ \texttt{in}\ t(x)$) and let $\rho(y)$ be the formula $\exists x(\varphi(x) \wedge \chi(x,y))$. Suppose that S, T are the types of x, y respectively, B is a pebbled structure appropriate for $\exists x\varphi(x)$, and a, b range over S^B, T^B respectively. For brevity, we do not mention B.

First, pick any $b \in Rng(\lambda)$ and fix an $a \in Rng(s)$ such that $b \in Rng(t(a))$. Then $(\varphi(a) \wedge \chi(a,b))$ is true, hence $\rho(b)$ is true and therefore $b \in Rng(\delta y \rho(y))$.

Second, pick any $b \in Rng(\delta y \rho(y))$. We consider two cases.

Case when $\rho(b)$ is true. There is an a such that $(\varphi(a) \wedge \chi(a,b))$ is true. Hence $a \in Rng(s)$ and $b \in Rng(t(a))$, so that $b \in Rng(\lambda)$.

Case when $\rho(b)$ is false. Clearly, $b = \texttt{default}_T$ and $\rho(b')$ is false for every b' in T. But this impossible. Indeed, pick any $a \in Rng(s)$ and any $b' \in Rng(t(a))$. Then $\varphi(a) \wedge \chi(a,b')$ holds and therefore $\rho(b')$ holds. □

Remark It may seem that the term $\lambda := (\texttt{let}\ x\ \texttt{be}\ s\ \texttt{in}\ t(x))$ is equivalent to the following simpler term

$$\rho := \delta y\Big[\exists x\Big(x = s \wedge y = t(x)\Big)\Big]$$

where y is a fresh variable, but this is not correct. It is possible that $Rng(\rho) - Rng(\lambda)$ contains $\texttt{default}_T$. Indeed assume that $\texttt{default}_T \notin Rng(\lambda)$. Nevertheless, if s and each instance of $t(x)$ have at least two values, then, for any particular values of x and y, the term $x = s \wedge y = t(x)$ may evaluate to `false`. Hence for any particular value of y, the term $\exists x(x = s \wedge y = t(x))$ may evaluate to `false`, and therefore ρ may evaluate to $\texttt{default}_T$. □

7.6 The *if-then-else* Construct

We introduce another useful and definable construct.

Syntax Extend FO+δ with the following term-formation rule:

- If φ is a formula and t_1, t_2 are terms of the same type T, then

$$\textit{if } \varphi \textit{ then } t_1 \textit{ else } t_2$$

  is a term of type T.

Semantics Extend the recursive definition of the range of a term with the following clause.

$$Rng^B(\textit{if } \varphi \textit{ then } t_1 \textit{ else } t_2) \ := \ \begin{cases} Rng^B(t_1) & \text{if } Rng^B(\varphi) = True \\ Rng^B(t_1) \cup Rng^B(t_2) & \text{if } Rng^B(\varphi) = Both \\ Rng^B(t_2) & \text{if } Rng^B(\varphi) = False \end{cases}$$

The Elimination of ***if-then-else*** The if-then-else construct is definable in FO+δ.

Theorem 7.9 *Term (if φ then t_1 else t_2) is equivalent to*

$$\delta y\Big[\big(\varphi \wedge \varphi_1(y)\big) \vee \big(\neg\varphi \wedge \varphi_2(y)\big)\Big]$$

where y is a fresh variable and each φ_i expresses that $y \in Rng(t_i)$.

The proof is similar to but simpler than the previous proof. We skip it.

Remark (*if φ then t_1 else t_2*) is not necessarily equivalent to

$$\delta y\Big[\big(\varphi \wedge y = t_1\big) \vee \big(\neg\varphi \wedge y = t_2\big)\Big]$$

□

7.7 Quantifier "Elimination"

Let s, t be terms and V a set of distinct variables that contains all free variables of s and all free variables of t. The nondeterministic global functions GF_s, GF_t can be seen as endowed with variables V. The terms s and t equivalent if the two nondeterministic global functions coincide. We show that the quantifiers can be expressed by means of δ.

Lemma 7.10 *$\exists x\varphi(x)$ is equivalent to*

$$\Big(\big(\delta x(x \neq \texttt{default} \wedge \varphi(x))\big) \neq \texttt{default}\Big) \vee \varphi(\texttt{default})$$

where `default` *is the default of the type of x.*

Notice that Hilbert's ε-elimination of $\exists$, by $\varphi(\varepsilon x\varphi(x))$, doesn't work with δ. Furthermore, the natural fix using `let`, namely (`let` x `be` $\delta y\varphi(y)$ `in` $\varphi(x)$), doesn't work either. Here is a counterexample. Let $\varphi(x)$ be $(x \neq \delta z P(z))$ and consider a three-element structure with a unary relation P that contains two of the three elements.

Proof Let λ be $\exists x \varphi(x)$, and let ρ be the alleged equivalent. Further, let ρ_1, ρ_2 be the first and second disjuncts of ρ respectively. Suppose that T is the type of x, B is a pebbled structure appropriate for $\exists x \varphi(x)$, and a ranges over T^B. For brevity, we omit the superscript B.

1. Assume that $\texttt{true} \in \mathit{Rng}(\lambda)$ and fix an a with $\texttt{true} \in \mathit{Rng}(\varphi(a))$. If $a = \texttt{default}$, then $\texttt{true} \in \mathit{Rng}(\rho_2)$ and therefore $\texttt{true} \in \mathit{Rng}(\rho)$. Assume that $a \neq \texttt{default}$. Then $a \in \mathit{Rng}(\delta x(x \neq \texttt{default} \wedge \varphi(x))$, $\texttt{true} \in \mathit{Rng}(\rho_1)$ and therefore $\texttt{true} \in \mathit{Rng}(\rho)$.

2. Assume that $\texttt{false} \in \mathit{Rng}(\lambda)$. Then every $\mathit{Rng}(\varphi(a)) \leq \mathit{Both}$. It follows that

(i) $\mathit{Rng}(a \neq \texttt{default} \wedge \varphi(a)) \leq \mathit{Both}$ for all a, so that $\texttt{default} \in \mathit{Rng}(\delta x(x \neq \texttt{default} \wedge \varphi(x)))$, and therefore $\texttt{false} \in \mathit{Rng}(\rho_1)$; and

(ii) $\mathit{Rng}(\texttt{default}) \leq \mathit{Both}$ and therefore $\texttt{false} \in \mathit{Rng}(\rho_2)$.

Thus $\texttt{false} \in \mathit{Rng}(\rho)$.

3. Assume that $\texttt{true} \in \mathit{Rng}(\rho)$, so that $\texttt{true} \in \mathit{Rng}(\rho_1)$ or $\texttt{true} \in \mathit{Rng}(\rho_2)$. It suffices to find an element a with $\texttt{true} \in \mathit{Rng}(\varphi(a))$.

First suppose that $\texttt{true} \in \mathit{Rng}(\rho_1)$. Then $\mathit{Rng}(\delta x(x \neq \texttt{default} \wedge \varphi(x)))$ has a non-default value a. Clearly $\texttt{true} \in \mathit{Rng}(\varphi(a))$.

Second suppose that $\texttt{true} \in \mathit{Rng}(\rho_2)$. The desired $a = \texttt{default}$.

4. Assume that $\texttt{false} \in \mathit{Rng}(\rho)$. So $\texttt{false} \in \mathit{Rng}(\rho_1)$ and $\texttt{false} \in \mathit{Range}(\rho_2)$. It suffices to prove that $\texttt{false} \in \mathit{Rng}(\varphi(a))$ for all a.

Since $\texttt{false} \in \mathit{Rng}(\rho_1)$, we have that $\texttt{default} \in \mathit{Rng}(\delta x(x \neq \texttt{default} \wedge \varphi(x)))$. So $\texttt{false} \in \mathit{Rng}(a \neq \texttt{default} \wedge \varphi(a))$ for all a, and therefore $\texttt{false} \in \mathit{Rng}(\varphi(a))$ for all $a \neq \texttt{default}$.

It remains to prove that $\texttt{false} \in \mathit{Rng}(\varphi(\texttt{default}))$. But this exactly the fact that $\texttt{false} \in \mathit{Range}(\rho_2)$. □

Theorem 7.11 *Every term is equivalent to a term with no quantifiers (but possibly with δ).*

Proof Use Lemmas 7.10 and 7.3 □

7.8 Multiple Choice

7.8.1 Motivation

In the ASM context, a multiple choice is common. See for example the following version of the single-source shortest paths algorithm

```
if ∀u, v : dist(v) ≤ dist(u) + ℓ(u, v) then
   Mode := Final
else
   choose u, v : dist(v) > dist(u) + ℓ(u, v)
      dist(v) := dist(u) + ℓ(u, v)
      pred(v) := u
   endchoose
endif
```

where $\ell(u, v) = \infty$ if there is no edge from u to v.

This leads to the vector version of the δ operator: $\delta\bar{x}\varphi(\bar{x})$. The problem is how to extract from the chosen vector the components which may be useful, e.g. to construct more complicated terms.

One recipe is to use projection functions. For example,

- $[\delta(x, y)\varphi(x, y)].1$ gives the chosen x, and
- $[\delta(x, y)\varphi(x, y)].2$ gives the chosen y.

This recipe does not work because $[\delta(x, y)\varphi(x, y)].1$ and $[\delta(x, y)\varphi(x, y)].2$ do not necessarily refer to the same chosen vector.

Another recipe is to introduce vector equality and require that a term $\delta\bar{x}\varphi(\bar{x})$ may appear only in the context $\bar{y} = \delta\bar{x}\varphi(\bar{x})$ where the variables $\bar{y}$ do not occur freely in $\delta\bar{x}\varphi(\bar{x})$. For example, we may have a formula

- $(y_1, y_2) = \delta(x_1, x_2)\varphi(x_1, x_2)$

where y_1, y_2 do not occur in $\varphi(x_1, x_2)$. Consider a structure A where the type T_1^A of x_1 is not singleton or the type T_2^A of x_2 is not singleton, and let b_1, b_2 range over T_1^A, T_2^A respectively. Notice that, for all (b_1, b_2), the equality may evaluate to `false`. Thus, this approach can (in some evaluations) yield unintended results. It works better in connection with the alternative semantics of δ introduced in Section 8, because the equality can evaluate to `true` at just those (b_1, b_2) that in the range of $\delta\bar{x}\varphi(\bar{x})$, and because the alternative semantics takes into account all evaluations together.

The recipe that we adopt is the following.

7.8.2 Syntax

Extend FO+δ with the following rules:

- If j is an integer ≥ 2, and $\bar{x}$ is a tuple $x_1, \ldots, x_j$ of distinct variables of types $S_1, \ldots, S_j$ respectively, and $\varphi(\bar{x})$ is a formula, then

$$\delta\bar{x}\varphi(\bar{x})$$

  is a *vector term* of type $S_1 \times \cdots S_j$. The variables $\bar{x}$ are bound in new term. For every other variable y, all free (respectively bound) occurrences of y in $\varphi(\bar{x})$ remain free (respectively bound) in the vector term. The number j is the *dimension* of the vector term.

- If $\delta\bar{y}\varphi(\bar{y})$ is a vector term and $\tau(\bar{x})$ is a term of type T and no variable in $\bar{x}$ is free in $\varphi(\bar{y})$, then
$$\texttt{let } \bar{x} \texttt{ be } \delta\bar{y}\varphi(\bar{y}) \texttt{ in } \tau(\bar{x})$$
is a term of type T. Variables $\bar{x}$ are bound in the new term. For every other variable y, all free (respectively bound) occurrences of y in $\delta\bar{x}\varphi(\bar{x})$ or $\tau(\bar{x})$ remain free (respectively bound) in the new term.

7.8.3 Semantics

If $\bar{x}$ is a tuple $(x_1, \ldots, x_j)$ of distinct variables of types $S_1, \ldots, S_j$ respectively, and if B is a pebbled structure where none of the variables x_i is pebbled, define

$$Rng^B(\bar{x}) := S_1^B \times \cdots \times S_j^B$$

Extend the definition of the ranges with the following clause:

- Let $\delta\bar{x}\varphi(\bar{x})$ be a vector term. Suppose that B is a pebbled structure appropriate for $\delta\bar{x}\varphi(\bar{x})$ and let $\bar{a}$ range over $Rng^B(\bar{x})$. Then

$$Rng^B(\delta\bar{x}\varphi(\bar{x})) := X \cup Y \quad \text{where}$$

$$X = \{\bar{a} : Rng^B(\varphi(\bar{a})) > False\}$$
$$Y = \begin{cases} \{(\texttt{default}^B_{S_1}, \ldots, \texttt{default}^B_{S_j})\} & \text{if every } Rng^B(\varphi(\bar{a})) < True \\ \emptyset & \text{if some } Rng^B(\varphi(\bar{a})) = True \end{cases}$$

Finally extend the definition of the ranges with the following clause:

$$Rng^B(\texttt{let } \bar{x} \texttt{ be } \delta\bar{x}\varphi(\bar{x}) \texttt{ in } t(\bar{x})) \;\; := \;\; \bigcup\{Rng^B(t(\bar{a})) : \;\bar{a} \in Rng^B(\delta\bar{x}\varphi(\bar{x}))\}$$

7.8.4 Elimination of Vector Terms

We illustrate how to eliminate vector terms.

Lemma 7.12 *Suppose that x, y are variables of types S, T respectively, and let z be a variable that does not occur in* $\texttt{let } (x,y) \texttt{ be } \delta(x,y)\varphi(x,y) \texttt{ in } t(x,y)$. *The following claims are equivalent:*

1. $z \in Rng\Big(\texttt{let } (x,y) \texttt{ be } \delta(x,y)\varphi(x,y) \texttt{ in } t(x,y)\Big)$,
2. $\exists x \exists y \Big[\texttt{true} \in Rng(\varphi(x,y)) \wedge z \in Rng(t(x,y))\Big] \vee$
   $\forall x \forall y \Big[\texttt{false} \in Rng(\varphi(x,y)) \wedge z \in Rng(t(\texttt{default}_S, \texttt{default}_T))\Big]$

Proof is obvious. $\square$

Corollary 7.13 *Let τ be a term* $\Big(\texttt{let } (x,y) \texttt{ be } \delta(x,y)\varphi(x,y) \texttt{ in } t(x,y)\Big)$, *$z$ be a fresh variable, and $\chi(z)$ be a first-order formula expressing that $z \in Rng(\tau)$. Then τ is equivalent to $\delta z \chi(z)$.*

7.9 Extensions of FO+δ

In our opinion, the issue of various extensions of FO+δ deserves attention. Here we give only a couple of initial remarks.

7.9.1 $L^{\omega}_{\infty,\omega}$+ δ

The definition of logic $L^{\omega}_{\infty,\omega}$ can be found in [Ebbinghaus and Flum 1995]. The pebble-structure semantics of FO+δ straightforwardly extends to $L^{\omega}_{\infty,\omega}$+$\delta$. Theorem 7.4 remains true; in that sense δ does not increase the expressive power of $L^{\omega}_{\infty,\omega}$.

Theorem 7.14 *Let τ be an $L^{\omega}_{\infty,\omega}$+$\delta$ term, and let y be a fresh individual variable of the type of τ. There exists an $L^{\omega}_{\infty,\omega}$ formula $\tilde{\tau}(y)$ expressing that $y \in Rng(\tau)$. Furthermore, the only variables of $\tilde{\tau}(y)$ are those of τ plus y. Similarly, the only free variables of $\tilde{\tau}(y)$ are those of τ plus y.*

The proof is similar to that of Theorem 7.4. To take care of infinite conjunction, notice that

$$\begin{aligned} y \in Rng(\bigwedge_{i\in I} \psi_i) \quad \iff \quad & [(y = \texttt{true}) \wedge \bigwedge_{i\in I} \texttt{true} \in Rng(\psi_i)] \vee \\ & [(y = \texttt{false}) \wedge \bigvee_{i\in I} \texttt{false} \in Rng(\psi_i)] \end{aligned}$$

The clause for infinite disjunction is similar.

In Subsection 7.4, we have derived a number of corollaries from Theorem 7.4. Similar corollaries can be derived from Theorem 7.14.

7.9.2 FO + DTC + δ

TC and DTC denote the transitive closure operator and the deterministic transitive closure operator respectively; for the definitions see for example [Ebbinghaus and Flum 1995, Section 6.1]. We extend the pebble-structure semantics of FO+δ to account for DTC. For notational simplicity, we explain the semantics in the case where DTC applies to binary relations over elements rather than tuples of elements.

Let $\varphi(x, y)$ be an arbitrary formula where x and y have the same type T. Further let A be a structure of the vocabulary of φ, and let $a, b \in T^A$. We define the range of

$$[DTC_{x,y}\varphi(x, y)](a, b)$$

by explaining how to sample that range (that is how to produce each of the truth values in it). For each pair $(x, y) \in T^A \times T^A$, evaluate $\varphi(x, y)$ once. This gives a binary relation $R \subseteq T^A \times T^A$. If $DTC(R)$ includes (a, b), put `true` in $Rng([DTC_{x,y}\varphi(x, y)](a, b))$; otherwise put `false` in $Rng([DTC_{x,y}\varphi(x, y)](a, b))$.

It is easy to check that

$$[TC_{x,y}E(x, y)](a, b) \iff \texttt{true} \in Rng\Big([DTC_{x,y}(y = \delta z E(x, z))](a, b)\Big)$$

where E is an arbitrary binary relation. It is not known whether (and the experts do not believe that) the transitive closure of an arbitrary binary relation E is expressible in FO + DTC.

7.9.3 FO + IFP + δ

IFP denotes the inflationary fixed-point operator [Ebbinghaus and Flum 1995, Section 6.1]. Restrict attention to finite structures and assume that the only connectives are $\neg, \wedge, \vee$.

We extend the pebble-structure semantics of FO+δ to account for IFP. For notational simplicity, we explain the semantics in the case where the quantified relation symbol is unary.

Let $\varphi(x, P)$ be an arbitrary formula where x is a variable of some type T and P is a relation with profile $T \to$ `Boole`. Further let A be a structure of the vocabulary of φ minus $\{P\}$, and let $a \in T^A$. We define the range of

$$[IFP_{x,P}\varphi(x, P)](a)$$

by explaining how to sample that range. Start by setting $P_0 := \emptyset$. Suppose that P_i has been computed. If $i > 0$ and $P_i = P_{i-1}$, then check whether P_i contains a; if yes then put `true` in the range, and if not then put `false` in the range. Suppose that $i = 0$ or else $i > 0$ but $P_i \neq P_{i-1}$. In this case, compute P_{i+1} as follows. For every $x \in T^A$, evaluate $\varphi(x, P_i)$; this gives a relation $R \subseteq T^A$. Set $P_{i+1} := P_i \cup R$.

Call an FO+IFP+δ formula φ IFP-*positive* if no IFP is in the scope of a negation in φ.

Proposition 7.15 *For every FO+IFP+δ sentence φ that is IFP-positive, there exists an existential second-order sentence χ expressing that* `true` $\in Rng(\varphi)$.

Proof Sketch The proof is straightforward in the case of structures with built-in order. Notice that the sequence $P_0, P_1, \ldots$ is polynomially bounded in length and can be easily indexed so that one relation of higher arity can describe the whole sequence. In the general case, χ has the form $\exists P(\chi_0(P) \wedge \chi_1(P))$ where $\chi_0(P)$ asserts that P is a linear order and $\chi_1(P)$ uses that order to express φ. $\square$

As usual, several inductions can be combined into one. The proof is similar to that of the Simultaneous Induction Lemma for the least fixed-point operator in [Moschovakis 1974, page 12]. To clarify things, we first explain the semantics of a simultaneous induction

$$IFP_{\langle x,P;y,Q\rangle}\langle\varphi(x, P, Q);\ \chi(y, P, Q)\rangle$$

where x, y are variables of some type T and P, Q are unary relation symbols with the same profile $T \to$ `Boole`. Fix a structure A of sufficiently rich vocabulary and let $a \in T^A$. The simultaneous induction builds a pair of relations $\langle X; Y\rangle$. In FO+IFP, one may ask for example whether $a \in X$. In FO+IFP+δ, we ask instead what the range of this assertion is. To sample this range do the following.

> Start by setting $P_0 := \emptyset$ and $Q_0 := \emptyset$. Suppose that P_i and Q_i have been computed. If $i > 0$ and $P_i = P_{i-1}$ and $Q_i = Q_{i-1}$, then check whether P_i contains a; if yes then put `true` in the range, and if not then put `false`

in the range. Suppose that $i = 0$ or else $i > 0$ but either $P_i \neq P_{i-1}$ or $Q_i \neq Q_{i-1}$. In this case, compute P_{i+1} and Q_{i+1} as follows. For every $x \in T^A$, evaluate $\varphi(x, P_i, Q_i)$; this gives a relation $P' \subseteq T^A$. Similarly, for every $y \in T^A$, evaluate $\chi(y, P_i, Q_i)$; this gives a relation $Q' \subseteq T^A$. Set $P_{i+1} := P_i \cup P'$ and $Q_{i+1} := Q_i \cup Q'$.

This simultaneous induction can be replaced with an appropriate single induction of the form

$$IFP_{x,y,\alpha,R}\psi(x, y, \alpha, R)$$

where α is a Boolean variable and R is a ternary relation symbol with profile $T \times T \times \texttt{Boole} \to \texttt{Boole}$. Here $P(x)$ is represented by $R(x, \texttt{default}, \texttt{true})$, and $Q(y)$ is represented by $R(\texttt{default}, y, \texttt{false})$ (where $\texttt{default}$ is of course $\texttt{default}_T$). See [Moschovakis 1974] for details.

Example 7.16 Using the `let` construct, we can produce a linear order over a given type T. Let u, v be variables of type T and let P be a relation symbol of type $T \times T \to \texttt{Boole}$. The desired induction is

$$IFP_{u,v,P}[\texttt{let } y = \delta x(\neg P(x, x)) \texttt{ in } (P(u, u) \wedge v = y) \vee (u = v = y)]$$

Using this as a part of a simultaneous induction, we can express for example the parity of T. □

In fact, `let` is not needed.

Proposition 7.17 *For every existential second-order sentence $\exists P\varphi(P)$, there is an IFP-positive FO+IFP+δ sentence χ such that*

$$\exists P\varphi(P) \iff \texttt{true} \in Rng(\chi)$$

Proof Sketch For simplicity, we consider the case when P is unary. The profile of P is $T \to \texttt{Boole}$ for some T. Let x, y be variables of type T. The induction

$$IFP_{x,P}[\delta y(x = x) = x]$$

gives an arbitrary set of elements of type T because $\delta y(x = x)$ is evaluated independently for each x. (This would have been the case even if we had written $\delta y(\texttt{true})$, but $\delta y(x = x)$ emphasizes the point.) However, replacing P in $\varphi(P)$ with the result of this induction does not give the desired χ. The problem is that different occurrences of the IFP formula will be evaluated independently. The way out is this. Use the given induction as a part of a larger simultaneous induction which first constructs an arbitrary set P of elements of type T and then computes $\varphi(P)$. □

8 An Alternative Independent-Choice Operator

Consider a term $\delta x\varphi(x)$ and let T be the type of x, B be a pebbled structure appropriate for $\delta x\varphi(x)$, and a range over T^B. To evaluate $\delta x\varphi(x)$ at B, you may start by evaluating every $\varphi(a)$. If there is an a such that $\delta x\varphi(x)$ evaluates to `true`, choose nondeterministically one such a as the result; otherwise output the default element of T^B.

This procedure can be criticized on the following grounds.

- It may produce the default element even though $Rng^B(\varphi(a)) > False$ for some a, and thus another execution of the procedure may output that a.
- It may output some element a with $Rng^B(\varphi(a)) = Both$ even though there may exist an element $b \in T^B$ with $Rng^B(\varphi(b)) = True$.

Instead of evaluating interpreted formulas $\varphi(a)$, we may evaluate their ranges. For each a, compute $Rng^B(\varphi(a))$ and set $M = \max\{Rng^B(\varphi(a)) : a \in T^B\}$; if $M = False$ then output the default element in T^B; otherwise choose arbitrarily one element in $\{a : Rng^B(\varphi(a)) = M\}$ and output it. We can make the new evaluation procedure cleaner by giving up the defaults and using instead the following principle: when no one is eligible, everyone becomes eligible[4].

This leads to appropriate changes in the definitions of the range of the term $\delta x\varphi(x)$ in B and in the term evaluation procedure. The new semantics gives us an alternative independent-choice operator which will be called δ'.

8.1 The Syntax and Semantics of FO+δ'

Syntax Extend functional first-order logic with the following construct:

- If x is a variable of type T and φ is a formula, then $\delta' x\varphi$ is a term of type T. All occurrences of x in $\delta' x\varphi$ are bound.

Semantics We restrict attention to the pebble-structure semantics of FO+δ'. The inductive definition of $Rng^B(\tau)$ of Section 6 generalizes readily to FO+δ'; we need only add the following clause.

- If $\tau = \delta' x\varphi(x)$, T is the type of x, and a ranges over T^B, then

$$\begin{aligned} Rng^B(\delta' x\varphi(x)) &:= \{a \in T^B : Rng^B(\varphi(a)) = M\} \quad \text{where} \\ M &= \max\{Rng^B(\varphi(a)) : a \in T^B\} \end{aligned}$$

[4]It may be interesting to note the use of this principle in the American judicial system. A judge must disqualify himself from hearing any case where he has a conflict of interest. But if a case produces a conflict of interest for *all* judges, then no judge can disqualify himself. There were cases when all judges had a conflict of interest. For example, federal judges sued for raises, since the constitution prohibits reducing their salaries, and inflation had, *de facto,* reduced their salaries.

In other words, our first preference is to choose an element a with $Rng^B(\varphi(a)) = True$. If this is impossible, so that $Rng^B(\varphi(a)) \leq Both$ for all a, then we turn to our second preference: to choose an element a with $Rng^B(\varphi(a)) = Both$. If this is impossible also, so that $Rng^B(\varphi(a)) = False$ for all a, then we choose any $a \in T^B$ whatsoever.

Remark Notice that there are two independent distinctions between the evaluations of $\delta x\varphi(x)$ and that of $\delta' x\varphi(x)$. Let Δ, Δ' be evaluators of $\delta x\varphi(x)$ and $\delta' x\varphi(x)$ respectively.

1. Δ' is biased toward elements a with $Rng^B v(\varphi(a)) = True$, whereas the more pragmatic and computationally oriented Δ is willing to pick any element a with $\mathtt{true} \in Rng(a)$.

2. In the case $\neg\exists x\varphi(x)$, Δ produces the default element of type T, whereas Δ' produces an arbitrary element of type T.

Accordingly, one can study four different operators. We have chosen δ as our main operator because it is most natural from the computational point of view. □

8.2 First-Order Expressibility of Ranges

Theorem 8.1 *Let τ be an FO+δ' term, and let y be a fresh individual variable of the type of τ. There exists an FO formula $\tilde{\tau}(y)$ expressing that $y \in Rng(\tau)$.*

Proof The proof is similar to that of Theorem 7.4. We need only show how to treat the case when τ is a δ' term.

- Suppose that τ is $\delta' x\varphi(x)$. By the induction hypothesis, there exists a first-order formula $\tilde{\varphi}(x, y)$ that expresses that $y \in Rng(\varphi(x))$. Set

$$\begin{aligned} \tilde{\tau} \;&:=\; \big(\tilde{\varphi}(y, \mathtt{true}) \wedge \neg\tilde{\varphi}(y, \mathtt{false})\big) \\ &\vee\; \big(\tilde{\varphi}(y, \mathtt{true}) \wedge \forall x\tilde{\varphi}(x, \mathtt{false})\big) \\ &\vee\; \forall x\neg\tilde{\varphi}(x, \mathtt{true}) \end{aligned}$$

□

Corollary 8.2 (Normal Form) *Every term t is equivalent to a term $\delta' x\varphi(x)$ where $\varphi(x)$ is first-order.*

Proof The desired $\varphi(x)$ asserts that x belongs to $Rng(t)$. □

Define an FO+δ term τ and an FO+δ' term τ' to be *equivalent* if they have the same ranges in every pebbled structure appropriate to both of them.

Lemma 8.3 *If $\varphi(x)$ is a first-order formula (which may have additional free variables) such that $\exists x\varphi(x)$ is logically true, then $\delta x\varphi(x)$ is equivalent to $\delta' x\varphi(x)$.*

Proof Obvious. □

Corollary 8.4 *Every FO+δ' term τ' is equivalent to some FO+δ term τ, and vice versa.*

Proof We prove the first claim, the proof of the second claim is similar. By the previous theorem, there exists a first-order formula $\varphi(x)$ that expresses that $x \in Rng(\tau')$. Since the ranges are never empty, the formula $\exists x \varphi(x)$ is logically true. The desired $\tau := \delta x \varphi(x)$. By the previous lemma, τ is equivalent to τ'. □

8.3 The `let` Construct

Extend FO+δ' with the `let` construct exactly as in the previous section. Again, `let` can be eliminated.

Theorem 8.5 *The term* (`let` x `be` s `in` $t(x)$) *is equivalent to the term*

$$\delta' y \Big[\exists x \Big(\varphi(x) \wedge \chi(x, y)\Big)\Big]$$

where $\varphi(x)$ is a first-order formula expressing that $x \in Rng(s)$, and y is a fresh variable, and $\chi(x, y)$ is a first-order formula expressing that $y \in Rng(t(x))$.

Proof Let λ be the term (`let` x `be` s `in` $t(x)$) and let $\rho(y)$ be the formula $\exists x(\varphi(x) \wedge \chi(x, y))$. Suppose that S, T are the types of x, y respectively, B is a pebbled structure appropriate for $\exists x \varphi(x)$, and a, b range over S^B, T^B respectively. For brevity, we do not mention B.

First we pick any $b \in Rng(\lambda)$ and show that $b \in Rng(\delta' y \rho(y))$. Fix an $a \in Rng(s)$ such that $b \in Rng(t(a))$. Then

$$\texttt{true} \in Rng(\varphi(a) \wedge \chi(a, b)) \leq Rng(\rho(b))$$

Since ρ is first-order, we have

$$True = Rng(\rho(b)) = \max\{Rng(\rho(b')) : b' \in T\}$$

so that $b \in Rng(\delta' y \rho(y))$.

Second we pick any $b \in Rng(\delta' y \rho(b))$ and show that $b \in Rng(\lambda)$. Let $M = \max\{Rng(\rho(b') : b' \in T\}$. Since ρ is first-order, $M \neq Both$. Since $b \in Rng(\delta' y \rho(b))$, we have that $Rng(\rho(b)) = M$. We consider the two possible cases.

Case $M = True$. Then $\rho(b)$ is true and therefore there is an a such that $\varphi(a) \wedge \chi(a, b)$ is true. Hence $a \in Rng(s)$ and $b \in Rng(t(a))$ and thus $b \in Rng(\lambda)$.

Case $M = False$. Then $\rho(b')$ is false for all b' in T. But this impossible. Indeed, pick any $a \in Rng(s)$ and any $b' \in Rng(t(a))$. Then $\varphi(a) \wedge \chi(a, b')$ is true and therefore $(\exists x(\varphi(x) \wedge \chi(x, b))$ is true. □

As in the case of FO+δ, we can add the *if-then-else* construct to FO+δ', but then that construct can be eliminated.

8.4 Quantifier "Elimination" for FO+δ' with `let`

Let φ be an arbitrary formula in the extension of FO+δ' with the `let` construct.

Lemma 8.6 *$\exists x\varphi(x)$ is equivalent to*

$$\texttt{let } y \texttt{ be } \delta' x\varphi(x) \texttt{ in } \varphi(y)$$

Proof Suppose that T is the type of x. Let B be a pebbled structure appropriate for $\exists x\varphi(x)$, a range over T^B and $M := \max\{Rng^B(\varphi(a)) : a \in T^B\}$. By the definition of the ranges of existential formulas, $Rng^B(\exists x\varphi(x)) = M$. By the definition of the ranges of δ' terms,

$$Rng^B(\delta' x\varphi(x)) := \{a \in T^B : Rng^B(\varphi(a)) = M\}$$

Using the definition of the ranges of `let` terms, we have

$$\begin{aligned} Rng^B(\texttt{let } x \texttt{ be } \delta' x\varphi(x) \texttt{ in } \varphi(x)) &= \bigcup\{Rng^B(\varphi(a)) : a \in Rng^B(\delta' x\varphi(x))\} \\ &= \bigcup\{Rng^B(\varphi(a)) : Rng^B(\varphi(a)) = M\} \\ &= M \end{aligned}$$

□

A First-Order Logic with the Witness Operator

Abiteboul and Vianu extended relational logic with a *witness operator* W [Abiteboul and Vianu 1991].[5] The extended logic will be denoted FO+W. The semantics of FO+W is described informally in their paper. We formalize that semantics by means of a generalization of global relations and make a couple of observations.

This section presupposes Subsection 2.1. In particular, we will use the notion of global relations.

A.1 Syntax of FO+W

FO+W is obtained from relational logic (RFO) by the following formation rule. If φ is a formula, $\bar{y}$ is a non-empty tuple of free variables of φ and $\bar{t}$ is a sequence of terms such that $length(\bar{t}) = length(\bar{y})$ then

$$[W\bar{y}\varphi](\bar{t})$$

is a formula. Within the square brackets, all occurrences of variables $\bar{y}$ are bound, but all free (respectively bound) occurrences of any other variable in φ remain free (respectively bound) in the new formula. Outside the square brackets, no occurrence of any variable is bound.

It may be convenient to make the free variables of φ explicit and write $\varphi(\bar{x}, \bar{y})$ where $\bar{x}$ is a tuple (possibly empty) of all remaining free variables of φ. Then the new formula is $[W\bar{y}\varphi(\bar{x}, \bar{y})](\bar{t})$.

[5] They also proved numerous results and gave numerous references on the nondeterminism in Datalog based languages.

Remark on the original notation. The original notation is $W\bar{y}\varphi(\bar{x},\bar{y})$ which corresponds to $[W\bar{y}\varphi(\bar{x},\bar{y})](\bar{y})$ in our notation. It denotes a relation whose variables are exactly those of φ (see the next subsection). We modified the original notation so as to make implicit variables explicit. □

A.2 Semantics of FO+W

A *global relation-set* Γ with vocabulary Υ and variables V assigns to every nondeterministic Υ-structure A a set Γ^A of relations ζ of variables V; it is assumed that if A, A' are isomorphic then we have the following: For each $\zeta \in \Gamma^A$ there is $\zeta' \in \Gamma^{A'}$ such that $(A,\zeta),(A',\zeta')$ are isomorphic by the same isomorphism, and *vice versa*. Notice that, if V' is a set of variables that includes V, then each $\zeta \in \Gamma^A$ can be viewed as a relation with variables V'.

Recall the notion of uniformization of a relation $P(\bar{x},\bar{y})$ [Moschovakis 1980, page 33]. Here $\bar{x}$ is a tuple of variables of length ≥ 0 and $\bar{y}$ is a tuple of variables of length ≥ 1. An $(\bar{x} \mapsto \bar{y})$-*uniformization* of P is any relation $Q \subseteq P$ satisfying the following condition:

$$\forall\bar{x}\Big[\exists\bar{y}P(\bar{x},\bar{y}) \longrightarrow (\exists \text{ unique } \bar{y})Q(\bar{x},\bar{y})\Big]$$

By induction on formula φ, define a global relation-set GRS_φ (or $GRS(\varphi)$).[6] Let A range over structures whose vocabulary includes that of φ.

- If φ is atomic, then $GRS^A_\varphi := \{GR^A_\varphi\}$.
- If φ is $\neg\chi$ then GRS^A_φ consists of the complements $\bar{\eta}$ of relations η in GRS^A_χ.
- Suppose that φ is $\chi \wedge \psi$ and let V be the set of free variables of φ. Think of members of GRS^A_χ and members of GRS^A_ψ as relations with variables V.

  $$GRS^A_\varphi := \{\eta \cap \theta : \eta \in GRS^A_\chi, \theta \in GRS^A_\psi\}$$

  Similarly, to apply other connectives to global relation-sets, we apply them pointwise to the relations in the sets. For example,

  $$GRS^A(\chi \leftrightarrow \psi) := \{(\eta \cap \theta) \cup (\bar{\eta} \cap \bar{\theta}) : \eta \in GRS^A_\chi, \theta \in GRS^A_\psi\}$$

- If φ is $\exists y\chi(\bar{x},y)$, then GRS^A_φ consists of the relations $\exists y\eta(\bar{x},y)$ where η ranges over GRS^A_χ. The case of $\forall$ is similar.
- If φ is $[W\bar{y}\chi(\bar{x},\bar{y})](\bar{t})$ and U is the collection of all $(\bar{x} \mapsto \bar{y})$-uniformizations of relations in GRS^A_χ then

  $$GRS^A_\varphi := \{\eta(\bar{x},\bar{t}) : \ \eta(\bar{x},\bar{y}) \in \ U\}$$

[6] Were this a stand-alone section, a better notation would be $[\![\varphi]\!]$.

A.3 Abiteboul and Vianu's Conjectures

Call an FO+W formula φ *deterministic* or *W-invariant* if, for every structure A of the vocabulary of φ, the relation-set GRS_φ^A is singleton. Say that a first-order formula χ *expresses* a deterministic FO+W formula φ if, for every structure A of the vocabulary of φ, $GRS_\varphi^A = \{GR_\chi^A\}$. Abiteboul and Vianu conjectured that every deterministic FO+W formula is first-order expressible. As far as we know, the conjecture is open.

Abiteboul and Vianu introduced a procedural version of first-order logic called FO^+. An FO^+ formula (or program) "is a sequence of statements of the form $R := \varphi$ where R is a relation variable and φ is an FO formula". They proved some results about $FO^+ + W$, and about the extensions of FO+W with the inflationary fixed-point operator and the partial fixed point operator. They conjectured that deterministic FO^+ formulas are first-order expressible. That stronger conjecture was recently refuted by Martin Otto [Otto 1998]. But not much is known about FO+W itself.

A.4 FO+W vs. FO+δ

In the case of minimal functional first-order logic, Boolean-valued relations can be viewed as relations over the `domain` (the only non-Boolean type). This allows us to give the following definition. An FO+W formula φ is *equivalent* to an FO+δ formula χ if $GRS_\varphi = GFS_\chi$.

For a short while we suspected that every FO+W formula is equivalent to an FO+δ formula. It is easy to see, however, that this is not true. Let R be a binary relation symbol.

Claim A.1 *$[WyR(x,y)](y)$ is not equivalent to any FO+δ formula.*

Proof Let $\varphi := [WyR(x,y)](y)$. Consider a structure A with three elements $0,1,2$ where $R^A = \{(0,1),(0,2)\}$. R^A has exactly two uniformizations: $\zeta_1 := \{(0,1)\}$ and $\zeta_2 := \{(0,2)\}$. Accordingly $GRS_\varphi^A = \{\zeta_1, \zeta_2\}$.

By contradiction, suppose that $\varphi(x,y)$ is equivalent to an FO+δ formula. Then GRS_φ^A is closed and, by the Mixing Lemma (Lemma 6.2), it contains, *e.g.* a function η such that $\eta(0,1) = \zeta_1(0,1) = \texttt{true}$ and $\eta(0,2) = \zeta_2(0,2) = \texttt{true}$. But $\eta \notin GRS_\varphi^A$. This gives the desired contradiction. $\square$

Remark Instead of $[WyR(x,y)](y)$, we could use $[WyQ(y)](y)$. The role of $\{(0,1),(0,2)\}$ would be played by $\{1,2\}$ and we would use $(\bar{x} \to y)$-uniformization where $\bar{x}$ is the empty tuple. $\square$

A.5 FO+W vs. Relational Logic

Let φ range over FO+W sentences and let A range over structures appropriate for φ. Each GRS_φ^A is a nonempty set of nullary relations. Nullary relations can be identified with the corresponding truth values. Thus GRS_φ^A is a nonempty subset of $\{\texttt{true}, \texttt{false}\}$.

Say that φ is *first-order expressible* if there are first-order sentences χ_0, χ_1 such that for every A

$$A \models \chi_0 \iff \texttt{false} \in GRS(\varphi), \quad A \models \chi_1 \iff \texttt{true} \in GRS(\varphi)$$

We will exhibit an FO+W sentence that is not expressible in first-order logic.

We start with a little excursion to graph theory. (Perfect) matching is usually defined for bipartite graphs. But it can be defined also for arbitrary digraphs. A *(perfect) matching* for a digraph $G = (V, E)$ is a permutation $f : V \to V$ such that every $(v, fv) \in E$.

Lemma A.2 *A digraph $G = (V, E)$ admits a matching if and only if there is $E' \subseteq E$ such that (V, E') is a disjoint union of cycles.*

Proof Use the fact that G is a disjoint union of cycles if every vertex has exactly one outgoing edge and exactly one incoming edge. □

Lemma A.3 *There is no first-order formula φ such that, for every finite digraph G, $G \models \varphi$ if and only if G admits a matching.*

Proof A graph $(X \cup Y, (X \times Y) \cup (Y \times X))$, where X, Y are disjoint finite sets, admits a matching if and only if X and Y have the same number of nodes. Use Ehrenfeucht-Fraïsse games to check that, for every φ there is a number n such that φ does not distinguish between

$$(X \cup Y, (X \times Y) \cup (Y \times X)) \text{ and } (X' \cup Y', (X' \times Y') \cup (Y' \times X'))$$

provided that each of the four sets X, Y, X', Y' contains at least n nodes. □

Theorem A.4 *There is an FO+W formula φ that is not first-order expressible.*

Proof The desired φ is

$$\forall y \exists x [WyE(x, y)](y)$$

where E is a binary relation symbol. Think about E as the edge relation of a digraph $G = (V, E)$ and restrict attention to finite digraphs. Let $E'(x, y) := [WyE(x, y)](y)$ be an $(x \mapsto y)$-uniformization of $E(x, y)$, and consider the function $y = f(x)$ from V to V whose graph is $E'(x, y)$. Intuitively, φ asserts that f is surjective and thus that G admits a matching. More formally check that $\texttt{true} \in GRS^G_\varphi$ if and only if G admits a matching. Now use Lemma A.3. □

A.6 The Expresssive Power of FO+W

In this subsection, we restrict attention to finite structures.

Let Υ be any finite vocabulary and let A range over Υ-structures. Say that a class K of Υ-structures (or the property $P(A) :\Leftrightarrow A \in K$) is *definable* in FO+W if there exists an FO+W sentence φ of vocabulary Υ such that

$$\texttt{true} \in GRS^A_\varphi \text{ if and only if } A \in K$$

In the previous section, we saw that the property of digraphs to admit matching is FO+W expressible. Let *Digraph Matching* be the decision problem whether a digraph admits a matching.

Theorem A.5 *Digraph Matching is PTime complete with respect to logspace reductions.*

Proof To establish that Digraph Matching is PTime, we reduce it to Bipartite Matching which is a known PTime complete problem [Greenlaw, Hoover and Ruzzo 1995]. Let $G = (V, E)$ be a digraph and let η be a one-to-one mapping from V to onto a set V' disjoint from V. Construct a bipartite graph $H = (V \cup V', F)$ where V is one part, V' is the other part, and

$$F := \{\{u, \eta v\} : (u, v) \in E\}$$

If G admits a matching ζ, then $\{\{u, \eta\zeta u\} : u \in V\}$ is the desired matching for H. Conversely, if $\{\{u, \theta u\} : u \in V\}$ is a matching for H, then $\zeta(u) := \eta^{-1}(\theta u)$ is a matching for G. Indeed, $(u, \zeta u) \in E$ because $\{u, \eta\zeta u\} = \{u, \theta u\} \in F$. Further, ζ is injective: if $\zeta(u) = \zeta(v)$, then $\theta u = \eta(\zeta u) = \eta(\zeta v) = \theta v$ and therefore $u = v$. Since we deal with finite digraphs, ζ is also surjective. (Actually, we do not need to appeal to finiteness here. The surjectivity can be shown directly. Let v be any vertex in V. Since h is a matching for H, there exists a vertex $u \in V$ such that $\theta u = \eta v$. But then $\zeta(u) = \eta^{-1}(\theta u) = v$.)

To establish that Digraph Matching is PTime hard, we reduce Bipartite Matching to D. Let $H = (V \cup V', E')$ be a bipartite graph with left part V and right part V'. Without loss of generality, we may assume that V and V' have the same number of vertices. Let η be a one-to-one mapping from V onto V'. Construct a digraph $G = (V, E)$ where

$$E := \{(u, \eta^{-1} v) : \ \{u, v\} \in F\}$$

so that $F := \{\{u, \eta v\} : (u, v) \in E\}$. The same proof as above shows that G admits a matching if and only H does. □

Remark It is easy to modify the sentence φ in the proof of the Theorem A.4 so that the modified sentence expresses Bipartite Matching. This would eliminate the need for Digraph Matching. It is interesting though to notice how simple φ can be. □

Question A.6 What is exactly the expressive power of FO+W? Is it true that only PTime properties are expressible in FO+W? Is it true that every PTime property is expressible in FO+W?

A.7 The Capricious Character of FO+W

A.7.1 Connectives

It is not true that, in FO+W, additional propositional connectives can be expressed by means of the obligatory ones. Here is one example of such inexpressibility.

Proposition A.7 *Let φ be the formula $[WxP(x)](x)$ and ψ the formula $Q(x)$. Then $\varphi \leftrightarrow \psi$ is not equivalent to any formula θ built from φ and ψ by means of* $\mathtt{true}, \mathtt{false}, \wedge, \vee, \neg$ *(and any other connectives which are monotone, increasing or decreasing).*

Here "equivalent" means having the same GRS^A for all A. In fact, we need only three structures A, B, and C to witness the desired inequivalence. All three structures have domain $\{0,1\}$ and interpret Q as $\{0\}$, but they interpret P differently: $P^A = \{0\}$, $P^B = \{1\}$ and $P^C = \{0,1\}$.

Proof Suppose θ were a counterexample. Consider first what happens in A. Here GRS^A_φ contains only one relation, namely $\{0\}$, and so GRS^A_θ, that is $GRS^A(\varphi \leftrightarrow \psi)$, also contains only one relation, namely $\{0,1\}$. Looking in particular at the element $0 \in |A|$ (that is evaluating the three formulas at 0 in A), we see that the propositional combination θ of φ and ψ must evaluate to true when both φ and ψ are evaluated to `true`. *Decreasing* the truth value of a *negative* occurrence of a propositional variable can only *increase* the truth value of the whole function. So

(1) θ is true if
all positive occurrences of φ are evaluated as true,
all negative occurrences of φ are evaluated as false,
and ψ is evaluated as true.

Note that, because all connectives in θ are monotone, every occurrence of φ or ψ in θ is positive or negative, so (1) makes good sense. (In contrast, the occurrence of φ in $(\varphi \leftrightarrow \psi)$ is neither positive nor negative.)

Now consider B, where $GRS^B_\varphi = \{\{1\}\}$, $GRS^B_\psi = \{\{0\}\}$ and therefore $GRS^B_\theta = \{\emptyset\}$. Looking in particular at the element $1 \in |B|$, we see that θ is false when φ is true and ψ is false. *Decreasing* the truth value of *positive* occurrence can only *decrease* the truth value of the whole formula. So

(2) θ is false if
all positive occurrences of φ are evaluated as false,
all negative occurrences of φ are evaluated as true,
and ψ is evaluated as false.

Finally, consider C, where $GRS^C_\varphi = \{\{0\},\{1\}\}$, $GRS^C_\psi = \{\{0\}\}$ and therefore $GRS^C_\theta = \{\{0,1\},\emptyset\}$. We shall obtain a contradiction by showing that $\{0\} \in GRS^C_\theta$.

According to the definition of GRS^C_θ, a relation in it can be obtained by interpreting each occurrence of φ as $\{0\}$ or $\{1\}$, interpreting each occurrence of ψ as $\{0\}$, and then combining these according to the way θ is built by connectives from φ and ψ. It is crucial that the definition allows the various occurrences of φ to be interpreted independently. Let us interpret all positive occurrences of φ as $\{0\}$ and all negative occurrences of φ as $\{1\}$. Thus, $\varphi(0)$ is interpreted as true in its positive occurrences and false in its negative occurrences, and of course $\psi(0)$ is true. So, by (1), we obtain the value `true` for $\theta(0)$. Similarly, $\varphi(1)$ is false at positive occurrences and true at negative occurrences, and $\psi(1)$ is false. So, by (2), we obtain the value `false` for $\theta(1)$. Thus we have obtained a relation in GRS^C_θ that holds at 0 but not at 1. This is the required contradiction. □

A.7.2 Propositional Laws

Some of the usual propositional laws fail in FO+W. For example, the distributive law

$$p \wedge (q_0 \vee q_1) \iff (p \wedge q_0) \vee (p \wedge q_1)$$

fails. Indeed, it is easy to check that

$$[WxP(x)](x) \wedge (Q_0(x) \vee Q_1(x))$$

is not equivalent to

$$\big([WxP(x)](x) \wedge Q_0(x)\big) \vee \big([WxP(x)](x) \wedge Q_1(x)\big)$$

in the structure A be the structure with base set $\{0, 1\}$ and relations $P^A = \{0, 1\}$, $Q_0^A = \{0\}$ and $Q_1 = \{1\}$.

A.7.3 The extension of FO+W with function symbols

The extension of FO+W with function symbols behaves badly with respect to substitution of terms for free variables. To see the problem, let φ be the formula $[WyR(x, y)](y)$ and let ψ be its substitution instance $[WyR(f(x), y)](y)$. ψ is also the result of applying W to $R(f(x), y)$. Fix a structure A where, for every a, there exists b with $R(a, b)$ and where there are distinct a_1, a_2 with $f(a_1) = f(a_2)$; we will omit superscripts A. Since ψ is the result of applying W to $R(f(x), y)$, GRS_ψ consists of the uniformizations of the relation $\{(a, b) : R(f(a), b)\}$, *i.e.*, it consists of the graphs of functions g that assign to each a some b with $R(f(a), b)$. Notice that $g(a_1)$ may differ from $g(a_2)$. Such a g is not obtainable by composing f with another function that assigns to each a some b with $R(a, b)$. In other words, the graph of g, which is in GRS_ψ, is not obtainable from any relation in GRS_φ by substitution via f. One cannot get the effect of the syntactical substitution by any semantical substitution as in the proof of Lemma 5.4.

References

Abiteboul, Simon and Vianu 1990 Serge Abiteboul, Eric Simon and Victor Vianu, "Non-Deterministic Languages to Express Deterministic Transformations", Proc. of ACM Symposium on Principles of Database Systems, 1990.

Abiteboul, Vianu 1991 Serge Abiteboul and Victor Vianu, "Nondeterminism in logic-based languages", Annals of Math and AI, no. 3, 151–186, 1991.

ASM Web Site The Michigan Web Site on Abstract State Machines, http://www.eecs.umich.edu/gasm/, maintained by James K. Huggins.

Börger, Grädel and Gurevich 1996 Egon Börger, Erich Grädel and Yuri Gurevich, "The Classical Decision Problem", Springer Verlag (Perspectives in Mathematical Logic), 1996.

Caicedo 1995 Xavier Caicedo, “Hilbert's ε-Symbol in the Presence of Generalized Quantifiers”, in M. Krynicki et al. (editors), “Quantifiers: Logics, Models and Computation”, Volume II, Kluwer Publishers, 1995, 63–78.

Church 1936 Alonzo Church, “An unsolvable problem of elementary number theory”, American Journal of Mathematics 58 (1936), 345-363. [Davis 1965, pages 88–107.]

Davis 1965 Martin Davis, “The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions”, Raven Press, New York, 1965.

Del Castillo, Gurevich and Stroetmann 1998 Giuseppe Del Castillo, Yuri Gurevich and Karl Stroetmann, “Typed Abstract State Machines”, [ASM Web Site].

Ebbinghaus and Flum 1995 Heinz-Dieter Ebbinghaus and Jörg Flum, “Finite Model Theory”, Springer 1995.

Fagin 1974 Ron Fagin, “Generalized first-order spectra and polynomial time recognizable sets”, in R. M. Karp (editor), “Complexity of Computation”, SIAM–AMS Proceedings 7 (1974), 43–73.

Garey and Johnson 1979 Michael R. Garey and David S. Johnson, “Computers and intractability: a guide to the theory of NP completeness”, Freeman, 1979.

Greenlaw, Hoover and Ruzzo 1995 Raymond Greenlaw, H. James Hoover, Walter L. Ruzzo, Limits to Parallel Computation: P-Completeness Theory, Oxford University Press, 1995.

Gurevich 1984 Yuri Gurevich, “Toward logic tailored for computational complexity”, in M. Richter et al., editors, “Computation and Proof Theory” (Logic Colloquium 1983), Springer Lecture Notes in Math. 1104 (1984), 175–216.

Gurevich 1988 Yuri Gurevich, “Logic and the challenge of computer science”, in E. Börger, editor, “Current Trends in Theoretical Computer Science”, Computer Science Press, 1988, 1–57.

Gurevich 1991 Yuri Gurevich, “Evolving Algebras: An Attempt to Discover Semantics ”, Bulletin of European Assoc. for Theor. Computer Science, no. 43, Feb. 1991, 264–284. A slightly revised version appeared in G. Rozenberg and A. Salomaa, editors, “Current Trends in Theoretical Computer Science”, World Scientific, 1993, 266–292.

Gurevich 1995 Yuri Gurevich, “Evolving Algebra 1993: Lipari Guide”, in E. Börger, editor, “Specification and Validation Methods”, Oxford University Press, 1995, 9–36. See also “May 1997 Draft of the ASM Guide”, Tech Report CSE-TR-336-97, EECS Dept, University of Michigan, 1997. Found at http://www.eecs.umich.edu/gasm/

Gurevich 1997 Yuri Gurevich, "A Guide to Abstract State Machines", unpublished notes.

Hilbert and Bernays 1939 David Hilbert and Paul Bernays, "Grundlagen der Mathematik", Vol. 2, Springer Verlag, 1939.

Kleene 1952 Stephen Cole Kleene, "Introduction to Metamathematics", D. Van Nostrand, NewYork, 1952 (reprinted by Wolters-Noordhoff — Groningen and North-Holland — Amsterdam in 1971).

Leisenring 1969 A. C. Leisenring, "Mathematical Logic and Hilbert's ε-symbol", Macdonald Technical and Scientific, London, 1969.

Moschovakis 1974 Yiannis N. Moschovakis, "Elementary Induction on Abstract Structures", North-Holland, Amsterdam, 1974.

Moschovakis 1980 Yiannis N. Moschovakis, "Descriptive Set Theory", North Holland, Amsterdam, 1980.

Otto 1998 Martin Otto, "Epsilon-Logic is More Expressive than First-Order Logic over Finite Structures", to appear.

Päppinghaus and Wirsing 1983 Peter Päppinghaus and Martin Wirsing, "Nondeterministic Three-Valued Logic: Isotonic and Guarded Truth-Functions", Studia Logica XLII.1 (1983), 1–22.

Rescher 1969 Nicholas Rescher, "Many-valued Logics", McGraw-Hill, New-York, 1969.

Trakhtenbrot 1950 Boris A. Trakhtenbrot, "Impossibility of an algorithm for the decision problem on finite classes", Doklady Akad. Nauk 70 (1950), 569-572.

Turing 1936 Alan M. Turing, "On computable numbers, with an application to the Entscheidungsproblem", Proc. of London Mathematical Society 2, no. 42 (1936), 230-236, and no. 43 (1936), 544-546. [Davis 1965, pages, 115–153.]