Δ-Languages for Sets and LOGSPACE Computable Graph Transformers

Alexei Lisitsa and Vladimir Sazonov*

Abstract

We discuss several versions of a set theoretic Δ-language as a reasonable prototype for "nested" data base query language where data base states and queries are considered, respectively, as hereditarily -finite sets and set theoretic operations. In a previous work such a language exactly corresponding to PTIME-computability was introduced. It is supposed that HF-sets are naturally presented by vertices of acyclic graphs. Here we consider a number of languages for Sub-PTIME computable set operations via corresponding graph transformers. Two such languages lead to a notion of NLOGSPACE and, respectively, DLOGSPACE computable queries over HF which appear the most natural, at our present knowledge, among others considered here. Unlike the "flat" relational data bases the problem of finding sufficiently good corresponding approach for HF proves to be more intricate and, furthermore, gives rise to some interesting questions in finite model theory (cf. Section 13).

1 General Introduction

Computability over sets (of tuples of sets of tuples of sets, etc.) or over "complex objects" is at present rather popular subject, especially in connection with "nested" data bases. There may be distinguished two directions: *typed* (e.g. [1, 13, 19, 22]) and *untyped* (cf. [6, 7, 8, 26, 30, 32, 34]) ones. The first is based on the *direct product* and *powerset* type constructs and presupposes, at least at the beginning, the hyperexponential (i.e. Kalmar elementary) computational complexity. Some additional special efforts are necessary to find restrictions (on the types height, ranges of variables, density/sparsity of data wrt types [13]) giving rise to more *tractable* query languages. The second direction is not concerned with exponentiation because it is based on the notion HF of *Hereditarily-Finite*

*Both authors from Program Systems Institute of Russian Acad. of Sci., Pereslavl-Zalessky, 152140, Russia. e-mail: {sazonov,lisitsa}@logic.botik.ru. Supported by RBRF (project 96-01-01717) and by INTAS (project 93-0972). The final version of the paper was also prepared during visiting by the second author in 1996 Princeton and Rutgers Universities (DIMACS) and Uppsala and Bern Universities.

Sets which may be axiomatized (of course, non-categorically) without any reference to the powerset operation. The tractability of corresponding Δ-languages considered in [30]–[39] and below is achieved quite straightforward. Also some approach to types within this framework oriented to nested databases was outlined in [34].

We confine the discussion on the connection of HF-sets with data bases to the above references and to a general note that nested and HF-like complex data structures seem conceptually very natural for a direct representation of any deeply structured information (cf. [34] for more details). This may be compared e.g. with natural numbers or finite strings of symbols or even "flat" relational structures which are more low-level notions. This paper is mainly concerned with computational and complexity-theoretic aspect of querying over HF, specifically, with an attempt to capture the class of (N)LOGSPACE-computable queries.

It is a crucial problem how to represent HF-sets both in a real computer and in any mathematical model of computation such as Turing machine. I.e. we must have some encoding ν : Codes $\to$ HF of the abstract data structure HF. A general theory of computability over any domain D wrt some encoding ν : Codes = Natural Numbers $\to D$ is presented in [9] as so called *numbering theory*. However, this approach was not originally concerned with complexity of computations. So, we must take into account subtle distinctions between many encodings which would be considered there as equivalent. In particular, we decide here and in the previous works to represent hereditarily-finite sets by vertices of finite *well-founded* (= *acyclic* in the finite case) graphs G with edges $u \to_G v$ corresponding to the membership relation $\in$ in HF. Previously considered versions of Δ-language describe exactly the class of all PTIME-computable operations over HF-sets wrt corresponding (regular [37, 35, 39]) coding of sets, in particular, wrt the graph encoding. (It is unclear which coding and which version of Δ-language are most "genuine"; cf. Concluding Remarks in [34]).

Note, that analogous representation of sets (of sets of sets ...) by graphs was also used in [6, 7] and [2]. However, in [2] it is considered rather unusual set theory with Antifoundation Axiom which says that arbitrary, even non-well-founded (and possibly infinite) graph with a distinguished vertex denotes a uniquely defined set in so called *anti*-well-founded universe of sets[1]. We consider corresponding approach in [36].

Unfortunately, it is rather expensive to recognize in general if any two vertices v_1, v_2 of a graph G denote the same set in the universe HF. (This computational problem proves to be in PTIME but perhaps not in (N)LOGSPACE $\subseteq$ PTIME due to its PTIME-completeness [6]. Such a procedure was one of the basic tools of PTIME realization of corresponding Δ-language [32, 34, 36].) That is why we are attempting here to find a reasonable restriction) of this graph transformation approach for realizing a version of Δ-language without

[1]In such a universe there exists e.g. a (unique) set Ω satisfying the identity $\Omega = \{\Omega\}$.

numerous identifications of vertices denoting the same set.

In particular, we restrict ourselves to *extensional* acyclic graphs where no such identification is possible, and define several notions of a "computable" transformation of such graphs (what may be considered as a kind of *evolving algebra* approach [15]) allowing to realize effectively, actually in (N)LOGSPACE, corresponding versions of the Δ-language. Our present task is to find such a Δ-language and encoding of sets which would correspond to (N)LOGSPACE *as better as possible*.

After a number of intermediate results in Sections 5–8 we present in Sections 9 and 12 some reasonable solution of this task as Main Results, respectively, for the case of NLOGSPACE and (deterministic version of) LOGSPACE-computability over HF. More precisely, we have to confine ourselves to a class of (N)LOGSPACE-computable operations over HF satisfying some sufficiently natural and actually unavoidable additional requirements (defined also in terms of (N)LOGSPACE-computability). In particular, we get exactly all (N)LOGSPACE-predicates over HF without any restriction. However, the syntax of corresponding versions of Δ-language, being effective, has somewhat artificial form. It is interesting question, whether it can be presented as some extension of Δ by finite number of some operations or schemes (like primitive recursion) so that, moreover, all the constructs could be mutually axiomatized.

The Main Results of Sections 9 and 12, as well as those (5–11) of Section 8 and examples in Section 11 (based on some technical considerations on definability of linear order in Section 10), are new in comparison with the previous version of this paper [38]).

The expressive power of many other versions of Δ-language considered in this paper is also faithfully characterized in terms of corresponding classes of graph transformers, however less natural. So, even if *exactly* NLOGSPACE (with no restrictions such as mentioned above) is captured in (9,10), Section 8), the class of described operations is probably not closed under arbitrary compositions (unlike the approach presented in the Main Result). This is because in our encodings of HF-sets there is an essential difference between the codes of n-tuples of sets and n-tuples of codes of the same sets from the point of view of *Sub*-PTIME-computability. It proves that defining Sub-PTIME-computability over HF and corresponding notion of definability is more problematic task than in the case of "flat" data bases [14, 17, 18]. More precisely, our reduction of the "nested" case to the "flat" one involves some peculiar technical problems and considerations, especially for the case of Sub-PTIME. The main reason for this is a *higher abstract level* of HF-sets in comparison with the first-order finite structures.

In fact, we reduce various versions of Δ-language to the language of first-order logic with a transitive closure operator $\mathrm{FO}^{\otimes}$ over finite graphs. It was shown by N.Immerman [17] that in the presence of a linear order ($\prec$) this language (even closed under negations [18]; cf. also [41]) exactly corresponds to NLOGSPACE. It is also used an analogous description of DLOGSPACE, i.e.

deterministic LOGSPACE [17].

The description of PTIME-computability over HF mentioned above is based on a similar approach to PTIME in terms of *recursive* "global" function(al)s in finite segments $\{0, 1, \ldots, \square - 1, \square\}$ of natural numbers [14, 27, 28] or, essentially equivalently, in terms of recursive global predicates over finite linear ordered first-order structures [16, 24, 42]. The last version is usually symbolized as $\mathrm{FO}_{\prec} + \mathrm{LFP} = \mathrm{PTIME}$ where $\prec$ denotes any linear ordering of a finite domain and LFP (instead of $\otimes$ above) is a *least fixed point* (i.e. recursion) operator. Note, that *primitive recursion* over $\{0, 1, \ldots, \square - 1, \square\}$ corresponds to DLOGSPACE [14].

Many considerations of this paper (which do not use $<_{\mathrm{HF}}$, a canonical linear order on HF) are also applicable to any reasonable set theoretical universe $\mathcal{V}$, possibly containing infinite sets. However, in this case, instead of (N)LOGSPACE-computability, we must consider only definability in the language FO and its versions like above $\mathrm{FO}^{\otimes}$, $\mathrm{FO} + \mathrm{LFP}$, etc.

The origins of Δ-language are as follows. A rather weak but natural and elegant class of set theoretic operations, so called *basic* or *rudimentary* operations was considered in [11, 20] with corresponding *Basic Set Theory* [11]. Analogous "predicative" versions of set theory were discussed in [10]. Such a set theory is quite sufficient for many elementary and every-day mathematical considerations. Basic operations constitute a proper subclass of *Primitive Recursive Set Operations* considered in [21]. It was shown in [11] that basic predicates coincide with those definable by Δ_0-formulas [23] in the language $\{\in\}$. It was proved in [30, 31, 32] that provably-total Σ-definable operations of KP_0, Kripke-Platek set theory without foundation axiom (cf. [3]), coincide with these basic (rudimentary) operations. Moreover, it was proved in [32] that KP_0 is, nevertheless, a conservative extension of the only Extensionality Axiom relative to Δ_0-formulas.

The class of basic operations and also the Basic Set Theory were extended in [30]–[39] to more reach versions called there Δ-language and Δ-set theory or *Bounded Set Theory* (by some analogy with *Bounded Arithmetic*; cf. e.g. [5, 25, 28]). Bounded Set Theory and its class of Δ-definable or, equivalently, provably computable operations over sets exactly correspond to PTIME-computability over HF. In this paper we consider the basic language of R.Gandy under the same name Δ and several its extensions corresponding to (N)LOGSPACE-computability.

2 Preliminaries and Technical Introduction

Remember that the universe HF of "pure" *hereditarily-finite sets* is defined inductively as the least class of sets such that $\emptyset \in \mathrm{HF}$ and if $x_1, \ldots, x_n \in \mathrm{HF}$ then $\{x_1, \ldots, x_n\} \in \mathrm{HF}$ Actually, for real data base applications we have to consider, as in [34], a more general universe $\mathrm{HF}(\mathcal{A}, \mathcal{U})$ with *urelements* $\mathcal{U}$ (or atoms) and *attributes* $\mathcal{A}$: let $\mathcal{U} \subseteq \mathrm{HF}(\mathcal{A},\mathcal{U})$ and if $x_1, \ldots, x_n \in \mathrm{HF}(\mathcal{A},\mathcal{U})$ and $A_1, \ldots, A_n \in \mathcal{A}$ then $\{A_1 : x_1, \ldots, A_n : x_n\} \in \mathrm{HF}(\mathcal{A},\mathcal{U})$ where $A_i : x_i$ are elements x_i labeled by the attributes $A_i \in \mathcal{A}$. (We may take $\mathcal{A} = \mathcal{U} =$ a set of some or all finite strings in an alphabet.) However, for simplicity we restrict ourselves here to the pure HF-sets.

PTIME- and, respectively, (N)LOGSPACE denote computability by a Turing machine in the time polynomial in the length of the input and, respectively, by a (nondeterministic) Turing machine using the working tape of the length logarithmic in the length of the input. The typical inputs and outputs for a Turing machine are finite strings in a finite alphabet or, slightly more general, finite graphs, etc. For definiteness we may use the denotation DLOGSPACE for (deterministic) LOGSPACE. It is well-known that DLOGSPACE $\subseteq$ NLOGSPACE $\subseteq$ PTIME and it is an *open question* whether $\subseteq$ are, in fact, proper inclusions here [12].

There is a problem of some ambiguity of the notion of Nondeterministic LOGSPACE-computability of functions in contrast to predicates: different nondeterministically chosen ways of computation may give different results. There is a reasonable direct approach to defining what is NLOGSPACE-computable function. However, we will actually work in terms of an equivalent notion of $\mathrm{FO}^{\otimes}_{\prec}$-definability described below.

The key notion for this paper is the following definition of (say, PTIME- or LOGSPACE-) *computability* of operations $q : \mathrm{HF} \to \mathrm{HF}$ over HF-sets, instead of finite strings. Here $q(s) = a$ means informally, for any $s, a \in \mathrm{HF}$, that s is a data base *state* and a is an *answer* to the *query* q asked about the state s. Let $\nu : \mathrm{Codes} \to \mathrm{HF}$ be any surjective encoding of states/answers. We say that q is (PTIME-, etc-) computable wrt ν if the following diagram

$$\begin{array}{ccc}
\mathrm{HF} & \xrightarrow{q} & \mathrm{HF} \\
{\scriptstyle\nu}\big\uparrow & & \big\uparrow{\scriptstyle\nu} \\
\mathrm{Codes} & \xrightarrow{Q} & \mathrm{Codes}
\end{array}$$

commutes for some (PTIME-, etc.-) computable transformation Q between codes. In other words, $q(\nu(c)) = \nu(Q(c))$ holds for all $c \in \mathrm{Codes}$. For the case of PTIME or (N/D)LOGSPACE we must denote corresponding classes of set-theoretic operations like $\mathcal{PTIME}_\nu$, $\mathcal{NLOGSPACE}_\nu$) or $\mathcal{DLOGSPACE}_\nu$ to show that the corresponding notion of computability over HF *depends* on an encoding ν. In the main case of graph encoding γ considered below we will write $\mathcal{PTIME}$ instead of $\mathcal{PTIME}_\gamma$. (Note, that $\mathcal{NLOGSPACE}_\gamma$ or $\mathcal{DLOGSPACE}_\gamma$ are very sensitive to small variations of this γ.)

Sometimes we have to consider computability of set-theoretical operations $q : \mathrm{HF} \to \mathrm{HF}$ with respect to *different* encodings ν_1 and ν_2 which serve for representation of inputs and outputs, respectively:

$$\begin{array}{ccc} \mathrm{HF} & \xrightarrow{q} & \mathrm{HF} \\ {\scriptstyle\nu_1}\Big\uparrow & & \Big\uparrow{\scriptstyle\nu_2} \\ \mathrm{Codes}_1 & \xrightarrow{Q} & \mathrm{Codes}_2 \end{array}$$

In general, let $\mathcal{C}$ denote some notion of computability over Codes, for example corresponding to some complexity class. More precisely, $\mathcal{C}$ is a recursive set of programs in a reasonable programming language, such as the language of Turing machines. We associate with $\mathcal{C}$ the corresponding class of transformers Codes $\to$ Codes. Then define $\mathcal{C}_\nu^{(1)}$ as the class of $\mathcal{C}$-computable *one-place* operations (and predicates) over HF with respect to ν. Some approaches to defining the class $\mathcal{C}_\nu = \cup_m \mathcal{C}_\nu^{(m)}$ of $\mathcal{C}$-computable *many-place* operations will be discussed soon.

Note, that for any given algorithm Q : Codes $\to$ Codes (from a fixed class $\mathcal{C}$), it may be problematic to decide whether there exists (actually, unique) q making the above diagram commutative. We say that the class $\mathcal{C}_\nu$ of computable operations q over HF has an *effective syntax* if (at least) there exists a recursive (not necessary $\mathcal{C}$-computable) family of programs $Q_\pi \in \mathcal{C}$, $\pi = 0, 1, \ldots$, with all corresponding $q_\pi \in \mathcal{C}_\nu$ existing and exhausting the class $\mathcal{C}_\nu$. (Here the programs Q_π may not exhaust $\mathcal{C}$.)

Alternatively, we may let π to range over formal expressions of a language L, instead of natural numbers. In this case q_π and Q_π may be considered, respectively, as *denotational* and *operational semantics* of any expression (program) π in this language. In contrast to $\mathcal{C}$, each L-program π will have a corresponding q. In this paper we shall take in the role of L suitable versions Δ' of a natural set-theoretic language Δ (cf. Section 3) with a clear denotational semantics and with *tractable* operational semantics, say, in terms of NLOGSPACE-computability.

Another problem consists (mainly for the case of $\mathcal{C}$ = (N)LOGSPACE) in defining a general and sufficiently closed class $\mathcal{C}_\nu$ of *many-place* $\mathcal{C}$-computable operations over HF. To this end we must reasonably generalize ν to the case of an encoding $\nu_m : \mathrm{Codes}_m \to \mathrm{HF}^m$ of all m-tuples of HF-sets. Then $\mathcal{C}$-computability of m-place functions q will be defined in terms of the diagram

$$\begin{array}{ccc} \mathrm{HF}^m & \xrightarrow{q} & \mathrm{HF} \\ {\scriptstyle\nu_m}\Big\uparrow & & \Big\uparrow{\scriptstyle\nu} \\ \mathrm{Codes}_m & \xrightarrow{Q} & \mathrm{Codes} \end{array}$$

with $Q \in \mathcal{C}$ formally a one-place transformer of codes.

The simplest choice is $\mathrm{Codes}_m = \mathrm{Codes}^m$, the set of m-tuples of Codes, and $\nu_m(\langle c_1, \ldots, c_m\rangle) = \nu^m(\langle c_1, \ldots, c_m\rangle) \rightleftharpoons \langle \nu(c_1), \ldots, \nu(c_m)\rangle$. But this choice is in general not the best one because for a reasonable ν we cannot guarantee that the predicates $\nu(c) = \nu(d)$ and $\nu(c) \in \nu(d)$ on $c, d \in$ Codes are $\mathcal{C}$-decidable for a lower complexity class $\mathcal{C}$, which is a very desirable condition. Nevertheless, it is possible that for some $\mathrm{Codes}_2 \neq \mathrm{Codes}^2$ and $\nu_2 \neq \nu^2$ the corresponding (formally one-place) predicates $(\nu_2(c))_1 = (\nu_2(c))_2$ and $(\nu_2(c))_1 \in (\nu_2(c))_2$ are

$\mathcal{C}$-decidable on $c \in \mathrm{Codes}_2$. Here $(\text{-})_i$ denotes i-th projection of a tuple. Even if we are lucky in this choosing ν_2 and, in general, ν_m it is still problematic whether the resulting class $\mathcal{C}_\nu$ is closed under compositions of many-place operations.

Let us note, that a sufficient condition for this is the existence of a transformer I_m in $\mathcal{C}$ such that the diagram

$$\begin{array}{ccc} \mathrm{HF}^m & = & \mathrm{HF}^m \\ {\scriptstyle \nu^m}\uparrow & & \uparrow{\scriptstyle \nu_m} \\ \mathrm{Codes}^m & \xrightarrow{I_m} & \mathrm{Codes}_m \end{array}$$

commutes. That is, we need an I_m in $\mathcal{C}$ which computes a unique code of m-tuple of sets by any given codes of each of these sets. Unfortunately, we evidently cannot hope on this for the case of $m = 2$ if, say, equality relation on HF is $\mathcal{C}$-decidable wrt ν_2, but *not* ν^2.

It proves reasonable (and even inevitable!) to consider also some *inessentially* restricted version $\mathcal{C}'$ of $\mathcal{C}$ so that $\mathcal{C}'_\nu$ would be a better, more natural HF-*analogue* of $\mathcal{C}$ than $\mathcal{C}_\nu$, having all closure properties we need.[2] It is this way, via some $\mathcal{C}'$, we shall approaching the notion of $\mathcal{C}$-computability over HF for the case $\mathcal{C} =$ (N)LOGSPACE. Corresponding $\mathcal{C}'$ will be denoted (in Section 9) as IC(N)LOGSPACE.

An important example of Codes for HF is the class of all finite *acyclic pointed graphs* ($\mathcal{AG}$), i.e. graphs G with no cycles and with a distinguished point (vertex) p in each. We often consider an m-tuple of points distinguished, what corresponds to our choice of Codes_m in this case. Then $\mathcal{C}$ will define a class of computable graph transformers. Let $\gamma : \mathcal{AG} \to \mathrm{HF}$ (or even $\gamma : \mathcal{G} \to \mathrm{HF}$ for *any* class $\mathcal{G}$ of graphs; cf. a generalization below) be Mostowski's *general collapsing* operation (an encoding of HF-sets by graphs) which assigns a set $\gamma(G, p) \in \mathrm{HF}$ to each $\mathcal{AG}$ $\langle G, p\rangle$ in such a way that

$$\gamma(G, p) = \{\gamma(G, p') : p' \to_G p \text{ for some (predecessor to } p) \text{ point } p' \text{ of } G\}.$$

In particular, if p has no predecessors in G then $\gamma(G, p) = \emptyset$. E.g. for G consisting just of three edges $p_1 \to p_2 \to p_3$ and $p_1 \to p_3$ we have $\gamma(G, p_1) = \emptyset$, $\gamma(G, p_2) = \{\emptyset\}$ and $\gamma(G, p_3) = \{\emptyset, \{\emptyset\}\}$.

We shall also write $p \in_G q$ instead of $p \to_G q$ and define formally any graph G as a first-order structure $\langle |G|, \in_G\rangle$ with $|G|$ its set of vertices and with the binary relation $\in_G$ for its edges. Sometimes we will apply γ to graphs with several kinds of edges (of various "colours"), i.e. with additional relations (such as a linear order $\prec_G$ on $|G|$), where $\in_G$ is just the *main* graph relation. Even *more general*, we may consider that $\gamma(G, p)$ is defined also for *any* graph, not necessary acyclic. Just apply γ to the *initial acyclic (or well-founded)* part of G, denoted as $\mathbf{WF}(G)$. Here $\mathbf{WF}(G) \rightleftharpoons \langle W, \in_G\restriction_W\rangle$ with $W \subseteq |G|$ the least set

[2] Actually, the relation between $\mathcal{C}$ and its restricted version $\mathcal{C}'$, as programming languages, may be more complex than the simple set inclusion $\mathcal{C}' \subseteq \mathcal{C}$.

of vertices such that if for any fixed vertex $y \in |G|$ all its predecessors $x \to_G y$ are in W then we must have also $y \in W$.

Let $\mathcal{G}$ be any class of finite pointed graphs, quite arbitrary or a special one such as $\mathcal{EG}, \mathcal{AG}, \mathcal{EAG}^*, \mathcal{EAG}^*_<$ or $\mathcal{EAG}^*_\prec$, etc. defined below. Then the restriction of γ to $\mathcal{G}$ defines corresponding encoding of HF-sets with Codes $= \mathcal{G}$ which will be called a *graph encoding* defined by $\mathcal{G}$. So, we could specify explicitly only $\mathcal{G}$.

We define *extensional* finite graphs ($\mathcal{EG}$) as those for which the ordinary set-theoretic *extensionality axiom* holds:

$$G \models \forall v \in x(v \in y) \& \forall v \in y(v \in x) \Rightarrow x = y,$$

or, equivalently, different vertices $v_1 \neq v_2$ in G must have different sets of predecessors, i.e. $\{v : v \to_G v_1\} \neq \{v : v \to_G v_2\}$.

Let $\in$-*isomorphism* between two graphs be any isomorphism with respect to the main binary relations of these first-order structures and the distinguished elements of these graphs. Sometimes we consider graphs G with a *congruence* relation $=_G$ (or $\approx_G$)[3] instead of the ordinary equality (identity). In this case we must consider an isomorphism between G_1 and G_2 rather as a relation $\leftrightarrow \subseteq |G_1| \times |G_2|$ which preserves all relations on these graphs, including $=_{G_1}$ and $=_{G_2}$, and the distinguished elements (and analogously for $\in$-isomorphism). We may call such an isomorphism as *generalized* one, or isomorphism *up to* the congruence relation $=_G$, in comparison with the ordinary (bijective) isomorphisms. Evidently, for $\in$-isomorphic graphs $\langle G_1, p_1\rangle$ and $\langle G_2, p_2\rangle$ we have $\gamma(G_1, p_1) = \gamma(G_2, p_2)$.

Note, that if $G = \langle |G|; \in_G, =_G\rangle$ is any $\mathcal{AG}$ with $=_G$ an equivalence relation identifying exactly those vertices which denote the same HF-sets in $T[G]$ according to the general collapsing then the latter may be not a congruence. However, we can easily define an *extensional acyclic graph* ($\mathcal{EAG}$) $G' = \langle |G'|, \in_{G'}, =_{G'}\rangle$ isomorphic to $T[G]$ (up to $=_{G'}$) so that $|G'| = |G|$, and $=_{G'}\, = \,=_G$ is a congruence relation wrt $\in_{G'}$. Just take $x \in_{G'} y \rightleftharpoons \exists x' \in_G y(x =_G x')$.

A *transitive subgraph* of a graph G with the main binary relation $\in_G$ is its full subgraph $\langle A, \in_G|_A\rangle$ for any subset $A \subseteq |G|$ satisfying the closure property $u \in_G v \;\&\; v \in A \Rightarrow u \in A$. This is a direct analogy to the ordinary notion of *transitive (sub)set* of the universe $\langle \mathrm{HF}, \in\rangle$. We will denote transitive sets in HF as T, T', etc. For any finite $\mathcal{AG}$ the set $T[G] \rightleftharpoons \{\gamma(G,p) : p \in |G|\}$ is transitive in HF. Evidently, any finite extensional acyclic graph ($\mathcal{EAG}$) G is $\in$-isomorphic (via general collapsing) to a unique transitive set $T = T[G] \in \mathrm{HF}$. We will often identify arbitrary $\mathcal{EAG}$ G with $T[G]$.

It can be shown that $\in$ and $=$ over HF are PTIME-computable wrt encoding $\gamma : \mathcal{AG} \to \mathrm{HF}$. (More precisely, wrt γ^2.) However, they are hardly computable in (N)LOGSPACE (due to PTIME-completeness of the corresponding problem for $=$; cf. [6]). Restricting γ to $\mathcal{EAG} \subseteq \mathcal{AG}$ and considering its corresponding versions $\gamma_m : \mathcal{EAG}_m \to \mathrm{HF}^m$, where $\mathcal{EAG}_m$ (= Codes$_m$) is the class of

[3] i.e. such equivalence relation $=_G$ that $G \models x = x' \& y = y' \& x \in y \Rightarrow x' \in y'$, etc.

$\mathcal{EAG}$'s (= Codes) with an m-tuple of distinguished vertices, makes this problem computationally trivial: different vertices of any $\mathcal{EAG}$ always denote different HF-sets.

We will consider various transformations of transitive sets $T \mapsto T'$. They will be also represented by corresponding transformers $G \mapsto G'$ between $\mathcal{EAG}$'s such that $T = T[G]$ implies $T' = T[G']$. For example, in Section 4 we will take

$$T' = T \cup \{q(\bar{x}) : \bar{x} \in T\}$$

for any operation $q(\bar{x})$ over HF of a special kind such that $q(\bar{x}) \subseteq T$ for all $\bar{x} \in T$. If $q(x,y) = \{x,y\}$ then corresponding transformer $G \mapsto G'$ may be defined by $|G'| \rightleftharpoons |G| \cup \{\bullet_{xy} : x, y \in |G|\}$, where $\bullet_{xy}$ are new vertices, $x \longrightarrow \bullet_{xy}$ and $y \longrightarrow \bullet_{xy}$ are new edges, and $\bullet_{xy} =_{G'} \bullet_{uv}$ iff $\{x,y\} = \{u,v\}$ as sets. Also let $\bullet_{xy} =_{G'} z$ iff z has exactly two predecessors $x \longrightarrow z$ and $y \longrightarrow z$.

On the other hand, to define a graph transformer $Q : G \mapsto G'$ (determined up to a graph isomorphism) we may use the first-order language FO$(\in, =)$ over G (with $\in$ and $=$ interpreted as $\in_G$ and $=_G$) as a natural tool. In this case it is reasonable to consider $|G'| \subseteq |G|^k$, $\in_{G'} \subseteq |G|^{2k}$, $=_{G'} \subseteq |G|^{2k}$, etc., for some $k \geq 1$, and define $|G'|$ and $\in_{G'}$, $=_{G'}$ by FO-formulas of k and $2k$ free variables, respectively, with the distinguished vertices of G used as individual constants. The distinguished vertices of G' may be defined as some k-tuples of distinguished vertices of G. (To this end we have to consider that any $\mathcal{EAG}$ G has at least two additional distinguished vertices $p_1 \neq_G p_2$. Say, let p_1 be the unique initial vertex, which has no predecessors and defines the empty set $\emptyset$ in any $\mathcal{EAG}$. Then let p_2 have the only predecessor $p_1 \longrightarrow_G p_2$ and define the singleton set $\{\emptyset\}$.) Alternatively, it may be easier to define distinguished vertices of G' by any FO-formulas $\varphi_1(\bar{x}), \ldots, \varphi_m(\bar{x})$ such that $G \models \exists! \bar{x} \varphi_i(\bar{x})$.[4] This approach to definability relies on the well known concept of first-order interpretations between theories or structures (cf. [17, 40]).

The same approach to definability of graph transformers may be considered for the following extension FO$^{\otimes}$ of the language FO. Just enrich FO by new predicate forming construct:

$$[\lambda \bar{x}\bar{y}.\varphi(\bar{x}, \bar{y}, \bar{z})]^{\otimes}$$

for any definable relation $\lambda \bar{x}\bar{y}.\varphi(\bar{x}, \bar{y}, \bar{z})$ of two lists of variables $\bar{x}$ and $\bar{y}$ of the same length. We call it the "horizontal" *transitive closure* and define its meaning

[4] Such a way of defining the distinguished vertices of G' by the formulas φ_i is essentially equivalent to the above using tuples. So, given the tuples $\bar{c}_1, \ldots, \bar{c}_m$, we may take $\varphi_i(\bar{x}) \rightleftharpoons \bar{x} = \bar{c}_i$. Conversely, given $G \models \exists! \bar{x} \varphi_i(\bar{x})$, $i = 1, \ldots, m$, we may consider without lost of generality that there exist m tuples $\bar{c}_1, \ldots, \bar{c}_m$ of the distinguished vertices of the original graph G such that $G \models \neg\varphi_i(\bar{c}_j)$ and $G \models \bar{c}_i = \bar{c}_j \Leftrightarrow \forall \bar{x}(\varphi_i(\bar{x}) = \varphi_j(\bar{x}))$ for all i, j. This allows to consider the following FO-definable bijection α (up to $=_G$) between tuples: $\bar{c}_i \mapsto \bar{d}_i$ and $\bar{d}_i \mapsto \bar{c}_i$, if $G \models \varphi_i(\bar{d}_i)$, and $\bar{d} \mapsto \bar{d}$ if $\bar{d} \neq \bar{c}_i$ and $G \models \neg\varphi_i(\bar{d})$ for all i. Then composing the given FO-definition of $|G'|, \in_{G'}$ and $=_{G'}$ with α results in a graph isomorphic to G' with the distinguished vertices presented by the tuples $\bar{c}_1, \ldots, \bar{c}_m$.

by

$$[\lambda \bar{x}\bar{y}.\varphi(\bar{x},\bar{y})]^{\otimes}(\bar{u},\bar{v}) \quad \text{iff} \quad \varphi(\bar{u},\bar{u}_1), \varphi(\bar{u}_1,\bar{u}_2), \ldots, \varphi(\bar{u}_n,\bar{v})$$

for some $n \geq 0$ and $\bar{u}_1, \ldots, \bar{u}_n$ ($\in |G|$). It may participate in any other formula as a predicate (both positively and negatively). The free variables $\bar{z}$, if any, serve as parameters. We shall also use more short, however ambiguous notation $[\varphi(\bar{u},\bar{v},\bar{z})]^{\otimes}$ for the formula $[\lambda\bar{x}\bar{y}.\varphi(\bar{x},\bar{y},\bar{z})]^{\otimes}(\bar{u},\bar{v})$ when it is clear from the context which two lists of variables of equal length in φ are considered. For example, $[\langle u,v\rangle \in z]^{\otimes}$, or even $[\langle u,v\rangle \in z^{\otimes}]$, denotes $[\lambda xy.\langle x,y\rangle \in z]^{\otimes}(u,v)$. Note, that to be $\mathcal{EAG}$ is evidently $\mathrm{FO}^{\otimes}$-definable property of graphs.

We may consider also *deterministic* version $\odot$ of transitive closure $\otimes$ which works as $\otimes$ for any formula $\varphi(\bar{x},\bar{y})$ when it defines a (partial) mapping $\bar{x} \mapsto \bar{y}$. Otherwise, applying $\odot$ gives, say, the false predicate (or, alternatively, $\odot$ preliminary corrects φ to make it "deterministic").

$\mathrm{FO}^{\otimes}$ and $\mathrm{FO}^{\odot}$ are essentially identical to the languages $\mathrm{FO}+\mathrm{TC}$ and $\mathrm{FO}+\mathrm{DTC}$, respectively, considered in [17, 18]. (Actually, in [17, 18] TC and DTC coincide with the *reflexive* versions of our $\otimes$- and $\odot$-constructs.) However, the denotation TC (and $*$) will be occupied in our paper for the "vertical" transitive closure in the universe of sets HF:

$$u \in \mathrm{TC}(v) \rightleftharpoons u \in^* v \rightleftharpoons [\lambda xy.x \in y]^{\otimes}(u,v).$$

I.e. we use TC and $*$ just in connection with the relation $\in$. Analogously, let $\in_G^* \rightleftharpoons \in_G^{\otimes}$ for the main graph relation $\in_G$. In general, let FO^* denote a fragment of $\mathrm{FO}^{\otimes}$ where $\otimes$ may be applied *only* to the primitive predicates. (Of course, $=^*$ and $\prec^*$ coincide with $=$ and $\prec$, respectively.)

It is proved in [17] that the notion of definability in $\mathrm{FO}+$ *positive* $\otimes$ in finite *linear ordered* structures is equivalent to NLOGSPACE-computability. Moreover, $\mathrm{FO}+$ *positive* $\otimes$ has the same expressive power in these structures as the full $\mathrm{FO}+\otimes$, i.e. as $\mathrm{FO}^{\otimes}$ [18]. This result is equivalent to the statement CoNLOGSPACE = NLOGSPACE (cf. also [41]) which have been widely believed previously as false. The same holds for $\mathrm{FO}^{\odot}$ and DLOGSPACE (where the equivalence of $\odot$ and *positive* $\odot$ is rather trivial). In particular, we have $\mathrm{FO} \subseteq$ DLOGSPACE. Therefore, we may freely interchange the notions $\mathrm{FO}_{\prec}^{\otimes/\odot}$ and (N/D)LOGSPACE where $\prec$ denotes a linear order.

We will need also an extension $\mathrm{FO}+\mathrm{LFP}$ of the first order language by the *least fixed point* construct

$$\textbf{the-least}\ P.[P(\bar{x}) \leftrightarrow \varphi(\bar{x},P(\bar{x}))]$$

with P occurring in φ positively. This construct is based on an iterative computation the least predicate P satisfying the condition in the brackets and actually subsumes $\otimes$ and $\odot$. It was shown in [16, 42] that definability in $\mathrm{FO}_{\prec}+\mathrm{LFP}$ over finite (linear ordered) models exactly corresponds to PTIME-computability.

Finally, we will use the abbreviations like $\mathcal{EAG}^*_\prec$, $\mathcal{EG}^*$, etc. also with superscript $*$ and subscript $\prec$ to designate that the graphs considered involve additional relations for the transitive closure $\in^*_G$ of the main graph relation $\in_G$ and, respectively, for *any* linear order on the vertices of a graph. The subscript $<$ denotes the *canonical* linear order on any $\mathcal{EAG}$ G which is inherited from the linear order $<_{\mathrm{HF}}$ defined below in Section 3 (due to isomorphism of G and $T[G] \subseteq \mathrm{HF}$).

3 Δ-Languages of Set Theoretic Operations

Define inductively Δ^*-*formulas* and Δ^*-*terms* by the clauses

$$\langle\Delta^*\text{-terms}\rangle ::= \langle\text{variables}\rangle \mid \{a, b\} \mid \cup a \mid \{t(x) : x{\in}^{(*)}a\&\varphi(x)\}$$

$$\langle\Delta^*\text{-formulas}\rangle ::= a{\in}^{(*)}b \mid \varphi\&\psi \mid \varphi \vee \psi \mid \neg\varphi \mid \forall x{\in}^{(*)}a\varphi(x) \mid \exists x{\in}^{(*)}a\varphi(x)$$

where φ and ψ are any Δ^*-formulas, a, b and t are any Δ^*-terms and x is a variable not free in a. The brackets around $*$ mean that there are two versions of the membership relation: $\in$ and its transitive closure $\in^*$. Then Δ_0^*-*formulas* are defined as those Δ^*-formulas involving only atomic terms (i.e. just variables). We write Δ (Δ_0) when $*$ is not used at all. The sublanguage Δ corresponds to the *basic* [11] or *rudimentary* [20] operations. Note, that our using the term Δ is not completely fixed in our different papers. In general, let Δ' denote some reasonable, still "bounded", extension of the class of basic operations. For example, the unrestricted *powerset operation* is considered as intuitively "unbounded".

We shall use Δ^*-terms and formulas both as syntactic objects and as denotations of their values in HF. For example, Δ-separation $\{x \in a : \varphi(x)\}$ for $\varphi \in \Delta$ gives the set of all x in the set a for which $\varphi(x)$ holds and is a partial case of the construct $\{t(x) : x \in a\&\varphi(x)\}$ = "the set of all values of $t(x)$ such that ...". Also $x \in \{a, b\}$ iff $x = a$ or $x = b$, $x \in \cup a$ iff $\exists z \in a(x \in z)$ and $\in^*$ is a transitive closure of the membership relation $\in$ on HF, i.e. $x \in^* y$ iff $x \in^{\otimes} y$ iff $x \in x_1 \in x_2 \in \ldots \in x_n \in y$ for some $n \geq 0$ and $x_1, \ldots, x_n$ in HF. The meaning of logical symbols & ("and"), $\vee$ ("or"), $\neg$ ("not"), $\forall$ ("for all"), $\exists$ ("exists") is well known. Note, that Δ^*-formulas involve only *bounded* quantification $\forall x{\in}^{(*)}a$ and $\exists x{\in}^{(*)}a$. That is why, according to traditions of mathematical logic, we use the name Δ for our language and Δ', etc. for various its versions. These bounded quantifiers have the same meaning as unbounded ones except the variable x ranges only over the set (denoted by) a.

It follows that any Δ'-term $t(\bar{x})$ defines a set-theoretic operation $\lambda\bar{x}.t(\bar{x})$: $\mathrm{HF}^n \to \mathrm{HF}$. For example, we may define the *transitive closure* of a set y as $\mathrm{TC}(y) \rightleftharpoons \{x : x \in^* y\}$. Let us identify the values **true** and **false** respectively with sets $\emptyset$ and $\{\emptyset\}$. Then formulas $\varphi(x)$ may be also considered as a kind of set theoretic terms (operations). So, we could write $\varphi(\bar{x}) = y$ for $y \in \mathrm{HF}$ a truth value. More precisely speaking, any Δ'-formula φ may be represented in this sense as Δ'-term **if** φ **then** $\{\emptyset\}$ **else** $\emptyset$ where, in general,

$$\textbf{if } \varphi \textbf{ then } t_1 \textbf{ else } t_2 \rightleftharpoons \cup\{z \in \{t_1, t_2\} : (\varphi \Rightarrow z = t_1)\&(\neg\varphi \Rightarrow z = t_2)\}.$$

Let us denote by $\lambda\bar{y} \in a.t(\bar{y})$ the graph of a function $t(\bar{y})$ of arguments $\bar{y}$ restricted to the set a. More formally, $\lambda\bar{y} \in a.t(\bar{y})$ is defined as the set in HF of ordered pairs $\{\langle\langle\bar{y}\rangle, z\rangle : \bar{y} \in a \& z = t(\bar{y})\}$ if $\bar{y}$ is *nonempty* list of variables. Otherwise, we let it coincide with $t = t()$. Ordered singletons, pairs, triples, etc. are defined in Δ as $\langle u\rangle \rightleftharpoons u$, $\langle u, v\rangle \rightleftharpoons \{\{u\}\{u, v\}\}$ and $\langle u, v, w\rangle \rightleftharpoons \langle\langle u, v\rangle, w\rangle$, etc. It follows that $\langle\langle\bar{y}\rangle, z\rangle = \langle\bar{y}, z\rangle$. Note, that $\cup\langle u, v\rangle = \{u, v\}$ and for any set of ordered pairs r the set of all the components of these pairs is defined in Δ as $\mathbf{field}(r) \rightleftharpoons \cup\cup r$.[5] Then corresponding *projections* satisfying $(\langle u, v\rangle)_1 = u$ and $(\langle u, v\rangle)_2 = v$ are Δ-definable by $(w)_1 \rightleftharpoons \cup\{x \in \cup w : \exists y \in \cup w. w = \langle x, y\rangle\}$ and symmetrically for $(w)_2$. Also let

$$\mathbf{dom}(r) \rightleftharpoons \{(w)_1 : w \in \mathbf{field}(r)\} \quad \text{and} \quad \mathbf{range}(r) \rightleftharpoons \{(w)_2 : w \in \mathbf{field}(r)\}.$$

In particular, $\mathbf{dom}(\lambda y \in a.t(y)) = a$.

For any list of Δ'-terms and Δ'-formulas $\bar{t}(\bar{y}) = t_0(\bar{y}), t_1(\bar{y}), \ldots, t_n(\bar{y})$ we abbreviate

$$\lambda\bar{y} \in a.[\bar{t}(\bar{y})] \rightleftharpoons \lambda\bar{y} \in a.t_0(\bar{y}), \lambda\bar{y} \in a.t_1(\bar{y}), \ldots, \lambda\bar{y} \in a.t_n(\bar{y}) \ .$$

If α is graph of a function then we write, as usually, $\alpha(x) = a$ instead of $\langle x, a\rangle \in \alpha$ and $y \in \alpha(x)$ instead of $\exists a \in \mathbf{field}\alpha(y \in a \& a = \alpha(x))$. If x is actually a tuple $\langle\bar{x}\rangle$ then we write $\alpha(\bar{x})$ instead of $\alpha(x) = \alpha(\langle\bar{x}\rangle)$. It is easy to define Δ-term $\mathbf{Apply}(\alpha, x)$ whose value is $\alpha(x)$. As in [34], we may define in the Δ-language many other useful operations on sets, e.g. the *cartesian product* $A \times B$, *cartesian power* A^k and *disjoint unions* $A + B$ and $\sum_{i \in I} A_i$ of any two sets A and B and of a family of sets A_i, etc. (We shall use some of these notions also in a more general context that a Δ'-language.)

As usual, any set g of ordered pairs may be considered as a (directed) graph with a pair $\langle u, v\rangle \in g$ playing the role of an edge $u \to_g v$ connecting the vertices u and v. Any pair $\langle g, p\rangle \in \mathrm{HF}$ with g a graph and p its vertex is just a *pointed graph* in the framework of set theory. If $p \notin \mathbf{field}(g)$ then it is considered as an *isolated* vertex.

We will extensively use the fact that any $\Delta^{(*)}$-formula φ is equivalent to a $\Delta_0^{(*)}$-formula [11]. (Cf. also [32, 37] for corresponding proof-theoretic considerations and formal *reductions*, as in lambda calculus, with $\Delta_0^{(*)}$-formulas being *normal forms* for $\Delta^{(*)}$-formulas). For example, the formula $\langle u, v\rangle = w$ is equivalent (and may serve as an abbreviation) to Δ_0-formula

$$\begin{aligned}&\exists s, p \in w(\forall x \in w(x = s \vee x = p)\& u \in s \& u, v \in p \& \\ &\qquad\qquad \forall x \in s(x = u)\&\forall x \in p(x = u \vee x = v))\end{aligned}$$

[5] Here the double union $\cup\cup$ is related with the above definition of ordered pairs.

and $\langle u, v\rangle \in z$ is equivalent to $\exists w \in z(\langle u, v\rangle = w)$. Analogously may be expressed $w = \cup u$, etc.

A set $T \in \mathrm{HF}$ is called *transitive* if $\forall y \in T \forall x \in y(x \in T)$. Evidently, $\mathrm{TC}(x)$ is the least transitive set containing x as a subset.

A *quasi-ordinal* is either the empty set $\emptyset$ or a singleton whose transitive closure consists only of singletons and $\emptyset$:

$$\mathrm{Quasiord}(x) \rightleftharpoons \forall y \in^* \{x\}(y \neq \emptyset \Rightarrow \exists! z(z \in y)).$$

Quasi-ordinals may be identified with the natural numbers by letting $0 \rightleftharpoons \emptyset$ and $n+1 \rightleftharpoons \{n\}$. Any arithmetical operation on natural numbers induces corresponding operation on quasi-ordinals via this bijection. Quasi-ordinals constitute a transitive class.

More popular in the classical set theory is the notion of *ordinal*. This is a transitive set whose all elements are also transitive sets. All finite ordinals may be obtained from $\emptyset$ by the "successor" operation $x' = x \cup \{x\}$.

However, there are notions which are hardly definable in Δ^*. For example, consider *canonical* or *lexicographical linear ordering* $<_{\mathrm{HF}}$ on HF uniquely defined by the

Axiom for $<_{\mathrm{HF}}$.

$$x < y \Leftrightarrow \exists u \in y \setminus x(\{v \in x : u < v\} = \{v \in y : u < v\}).$$

This ordering evidently coincides with that defined by the Ackerman's bijection $\mathbf{Ack} : \mathrm{N} \to \mathrm{HF}$, the most popular encoding of the universe HF:

$$m <_{\mathrm{N}} n \Leftrightarrow \mathbf{Ack}(m) <_{\mathrm{HF}} \mathbf{Ack}(n)$$

where $<_{\mathrm{N}}$ is the ordinary linear ordering of the set of natural numbers N and

$$\mathbf{Ack}(2^{n_1} + 2^{n_2} + \ldots + 2^{n_j}) = \{\mathbf{Ack}(n_1), \mathbf{Ack}(n_2), \ldots, \mathbf{Ack}(n_j)\}$$

for any $j \geq 0$ and $n_1 > n_2 > \ldots > n_j \geq 0$. (In particular, $\mathbf{Ack}(0) = \emptyset$.) Note, that very simple singleton operation $x \mapsto \{x\}$ corresponds to the arithmetical exponential operation $k \mapsto 2^k$ with respect to $\mathbf{Ack}$, which is surely intractable. It follows that we must consider other encodings for the universe HF like collapsing described in the Section 2.

We may extend Δ also by a set-theoretic *collapsing* operation $\mathrm{C} : \mathrm{HF} \to \mathrm{HF}$ which is a *restriction* of the general collapsing γ to the case of (pointed) $\mathcal{EAG}$'s considered as elements of the universe HF. So, $\mathrm{C}(\langle g, p\rangle)$ is the set in HF corresponding to the point p of g under γ. If g is not an extensional or an acyclic graph then let $\mathrm{C}(\langle g, p\rangle) = \emptyset$. In particular, it is definable in $\Delta + \mathrm{C}$ whether any $g \in \mathrm{HF}$ presents an acyclic graph.

We will need two other versions of C, *strong* and *weak*:

- $\tilde{\mathrm{C}} \rightleftharpoons \gamma|_{\mathcal{EG}}$ ($= \emptyset$ for non-$\mathcal{EG}$'s),

- $\hat{\mathrm{C}}(\langle g, <_g, p\rangle) \rightleftharpoons \mathrm{C}(\langle g, p\rangle)$ if $<_g$ is the canonical linear order on the vertices of $\mathcal{EAG}$ g, and $\rightleftharpoons \emptyset$, otherwise. I.e. $\hat{\mathrm{C}}$, unlike C, is properly defined only on $\mathcal{EAG}_<$'s $\subseteq$ HF, i.e. on the canonically ordered $\mathcal{EAG}$'s.

Another particularly important construct is *recursive Δ'-separation* **Rec**:

the-least $p.[p = \{x \in a : \varphi(x, p)\}]$.

It must be considered as a term which denotes the least solution $p \subseteq a$ of the equation in the square brackets (for $\varphi \in \Delta'$ satisfying a reasonable "positivity" condition for the variable p).

We will also consider an extension of Δ^*-language by a new kind of terms $r^{\otimes}$ (instead of formulas, as for the case of FO in Section 2) to denote the "horizontal" *transitive closure* of any relation (set of pairs) r. It will be called $\Delta^{\otimes}$, with corresponding $\Delta^{\otimes}$*-terms* and $\Delta^{\otimes}$*-formulas*. Evidently, $\otimes$ is definable by **Rec**.[6] *Deterministic* version of $\otimes$ is formally defined as

$r^{\odot} \rightleftharpoons D(r)^{\otimes}$ where $D(r) \rightleftharpoons \{\langle x, y\rangle \in r : \exists! z(\langle x, z\rangle \in r)\}$.

Analogously, we use denotations $\Delta_<$ or $\Delta_<^{\otimes}$ if the primitive predicate symbol $<$ for $<_{\mathrm{HF}}$ is included in the language.

The following characterization of the language $\Delta^* + \mathbf{Rec} + \mathrm{C}$ have been obtained in [30, 32, 34].

Theorem 1 *The class of operations over* HF *definable in the language* $\Delta^* + \mathbf{Rec} + \mathrm{C}$ *coincides with the class* $\mathcal{PTIME}$ *of those computable in polynomial time under graph representation* γ *of* HF*-sets or, equivalently, computable by graph transformers definable in* FO + LFP. □

It is essentially used in the proof of this theorem that $<_{\mathrm{HF}}$ is definable in $\Delta^* + \mathbf{Rec}$ [34] (by a term having the form analogous to that of the axiom for $<_{\mathrm{HF}}$). In particular, in the formulation we need not mention any linear ordering in defining graph transformers in FO+LFP. Also, $\mathcal{AG}$'s or $\mathcal{EAG}$'s may be equally used as Codes here.

4 Simply Computable Graph Transformers

Now we turn to Sub-PTIME-computability. *Only extensional acyclic finite graphs* $G = \langle |G|, \in_G\rangle$ will be considered below as codes of sets in HF (with several explicitly mentioned exceptions). Bearing in mind this approach to computability over HF in terms of abstract $\mathcal{EAG}$'s we prefer to deal directly with corresponding transitive sets $T = T[G]$ due to some technical advantages.

[6] Strictly speaking, we must write $\Delta^{*\otimes}$ instead of $\Delta^{\otimes}$ because $\otimes$ does not cover the full strength of $*$. So, TC is not definable even in $\Delta + \mathbf{Rec}$ because TC $\notin \mathcal{PTIME}_{\gamma'}$ for some specific version γ' of γ (cf. [34, 37]) such that, moreover, $\mathcal{PTIME}_{\gamma'}$ is exactly the class of operations over HF which are definable in a corresponding extension of $\Delta + \mathbf{Rec}$.

Let us represent any n-tuple of sets $\bar{\tau} \in \mathrm{HF}$ by $(n+1)$-tuple $\langle T; \bar{\tau} \rangle$ where $\bar{\tau} \in \mathrm{HF}$ and $T \in \mathrm{HF}$ is an arbitrary *transitive set* such that $\bar{\tau} \in T$. Then HF-set $\langle T; \bar{\tau} \rangle$ may be also considered as a graph $g_T = \{\langle u, v \rangle : u \in v \in T\}$ with the distinguished vertices $\bar{\tau}$. Evidently, the result of applying the encoding γ (or its set-theoretical version C) to this graph and its distinguished vertices $\bar{\tau}$ will give the same HF-sets $\bar{\tau}$: $\mathrm{C}(\langle g_T, \bar{\tau} \rangle) = \bar{\tau}$.

Let also any (general or restricted) notion of *computable* transformations of such tuples (graphs) $\langle T; \bar{\tau} \rangle \mapsto \langle T'; \bar{\tau}' \rangle$ be given. Then we say that $f : \mathrm{HF}^n \to \mathrm{HF}$ is (respectively) *computable*, if there exists computable transformation $F : \langle T; \bar{\tau} \rangle \mapsto \langle T'; \tau' \rangle$ such that, independently on any transitive $T \ni \bar{\tau}$, we have $\tau' = f(\bar{\tau})$.

This definition of computability over HF is just a reformulation of that based on the commutative diagram in Section 2 with $\mathcal{EAG}'s$ as Codes. Note, that this is slightly different from the encoding γ because we are considering here only extensional acyclic graphs (i.e. those arising from transitive sets T). This difference is inessential for considering PTIME-computability, however it is indeed crucial for the case of (N)LOGSPACE.

In this and the next sections we consider some class of simple DLOGSPACE-computable transformations which exactly corresponds to Δ^*-language. Then in Sections 6 and 7 both the class of transformations and Δ^*-language will be extended twice inside NLOGSPACE with preserving this correspondence, the last step giving rise to transformations definable by the First-Order Logic with Transitive Closure operator $\mathrm{FO}^{\otimes}$. Developing further these considerations in Section 8 we capture for the first time the notion of NLOGSPACE-computability over HF, however not so naturally as desired, and obtain some other variations of the mentioned results. Finally, the Main Results of Section 9 and 12 presents our best achievement in approaching to set-theoretic NLOGSPACE- and DLOGSPACE-computability.

Consider any first-order formula $\varphi(x, \bar{y})$, $\bar{y} = y_1, \ldots, y_s$, possibly with unbounded quantifiers, in the language $\{\in, \in^*, =\}$, i.e. FO^*-formula in the language $\{\in, =\}$. It defines the following *restricted powerset operation* applied to any transitive set T

$$\mathcal{P}_\varphi(T) \rightleftharpoons \{\{x \in T : \varphi^T(x, \bar{y})\} : \bar{y} \in T\}$$

where φ^T is the result of replacing all quantifiers $\forall v$ and $\exists v$, respectively, by $\forall v \in T$ and $\exists v \in T$ with the natural interpretation in T of the language $\{\in, \in^*, =\}$. The restricted powerset consists of all subsets of T *definable* by the formula $\varphi(x, \bar{y})$ of x with parameters $\bar{y}$ ranging over T. We associate with $\varphi(x, \bar{y})$ a formal construct denoted as $[\![x; \bar{y}.\varphi]\!]$ with bounded variables x and $\bar{y}$ (playing different roles). This defines also an *elementary transformation of the first kind* of arbitrary (nonempty) transitive sets T with any sequence of distinguished elements $\bar{\tau} = \tau_1, \ldots, \tau_k$ of a fixed length k (for the input set T):

$$[\![x; \bar{y}.\varphi]\!] : \langle T; \bar{\tau} \rangle \mapsto \langle T \cup \mathcal{P}_\varphi(T); \{x \in T : \varphi^T(x, \bar{\tau})\}, \bar{\tau} \rangle \ .$$

Here the output transitive set T' is $T \cup \mathcal{P}_\varphi(T)$ which has $\{x \in T : \varphi^T(x, \bar{\tau})\} \in \mathcal{P}_\varphi(T)$ as an *additional* distinguished element defined by formula φ. We need all the old distinguished elements τ_i to be preserved with an agreement that they are substituted for y_i, respectively. In the case the length of $\bar{\tau}$ is less than the length of $\bar{y}$, the empty set $\emptyset \in T$ is substituted for the extra y_i's.

Define *elementary transformation of the second kind* as replacing the distinguished values $\bar{\tau} = \tau_1, \ldots, \tau_k$ with a fixed k by a finite sequence $\tau_{j_1}, \ldots, \tau_{j_m}$ of these values in T defined by any fixed sequence of numbers $\bar{j} = \langle j_1, \ldots, j_m \rangle$, $1 \le j_1, \ldots, j_m \le k$, $m \ge 0$:

$$[\![\bar{j}]\!] : \langle T; \tau_1, \ldots, \tau_k \rangle \mapsto \langle T; \tau_{j_1}, \ldots, \tau_{j_m} \rangle.$$

Then any finite composition of n elementary transformations is called *simply computable* (in this restricted sense) *transformation* in n steps. Denote the class of these transformations as SIMPLE. This is a useful *technical* notion of computability. For example, the transformation

$$[\![x; \bar{y}.\varphi]\!]_0 : \langle T; \bar{\tau} \rangle \mapsto \langle T \cup \mathcal{P}_\varphi(T); \bar{\tau} \rangle$$

is simply computable in two steps. Often, by mentioning a transformation of the first kind we, in fact, mean just such slightly more weak one which does not generate any additional distinguished element. This transformation is rather natural if we just need to use some values $\{x \in T : \varphi^T(x, \bar{y})\}$, $\bar{y} \in T$, in an extension of T for further computation steps, but *not as the distinguished* ones. The reader himself may easily decide what exactly is understood in each concrete situation. Moreover, it is often unreasonable to mention explicitly (trivial) elementary transformations of the second kind. Our computable transformations will usually begin with a "saturation" of the initial T by a number of new sets and then by distinguishing some of them.

Evidently, for any SIMPLE transformer $F : \langle T; \ldots \rangle \mapsto \langle T'; \ldots \rangle$ we have $T \subseteq T'$. We will see in Section 9 that this "inflationary" (and also some "coherent") character of graph transformers is inevitable in a precise sense.

Note that, because we interpret any FO*-formula in transitive sets T, the formula $\forall v(v \in x \leftrightarrow v \in y)$ gives just the ordinary equality relation $x = y$ in T. We shall often use such useful observations implicitly. For other kinds of formulas relativized to T it depends on T whether their meaning in T coincide with the *expected* one. Therefore, we usually should take care of including in T sufficiently many elements to guarantee this coincidence.

For this purpose we shall use a *saturation transformation* which is finite iteration $[\![x; uv.(x = u \vee x = v)]\!]_0^n$ of the elementary transformation $[\![x; uv.(x = u \vee x = v)]\!]_0$. For example, this transformation will extend T by all *unordered* pairs $\{u, v\}$, $u, v \in T$ (because $\{u, v\} = \{x \in T : x = u \vee x = v\}$) as well as *ordered* pairs $\langle u, v \rangle = \{\{u\}, \{u, v\}\}$, triples $\langle \langle u, v \rangle, w \rangle$ and more long tuples for sufficiently large n. Evidently,

Proposition 1 *Any simply computable transformation*

$$F : \langle T; \bar{\tau}\rangle \mapsto \langle T'; \bar{\tau}'\rangle : \mathrm{HF} \to \mathrm{HF},$$

being a finite composition, is Δ^**-definable, (actually* Δ*-definable, if* $\in^*$ *is not involved in* FO**-formulas defining the transformation).*

□

4.1 SIMPLE $\subseteq$ FO $\subseteq$ DLOGSPACE

Simply computable transformations may be considered rather as transformations over *arbitrary* $\mathcal{EAG}$'s. Then any elementary transformation of the first kind $[\![x; \bar{y}.\varphi(x, \bar{y})]\!]$ adds a number of new vertices and edges to the given graph G. They must (evidently) correspond to the sets $\{x \in G : G \models \varphi(x, \bar{y})\}$, $\bar{y} \in G$, and, respectively, to the membership relation to any such set. Some of these new vertices should be *identified* (literally, or by a congruence relation) one with another and possibly with the old vertices of G, if they occasionally denote the same set in the universe under consideration. (Cf. also corresponding example in Section 2.) Note, that if $\in^*$ is indeed involved in formulas defining transformations then it is reasonable for more *effective* realization to consider $\mathcal{EAG}^*$'s, i.e. graphs G with *two* kinds of edges, corresponding both to $\in$ and $\in^*$ relations.[7] Then $\in^*$-relation in the resulting graph of any elementary transformation could be easily computed from the old version of $\in^*$ by introducing several additional $\in^*$-edges going to new vertices without the tedious direct computing the whole transitive closure of the resulting version of $\in$. Just use the recursive definition

$$y \in^* y' \leftrightarrow y \in y' \vee \exists w.(y \in^* w \& w \in y')$$

which allows to compute new version of $\in^*$ (to the left of $\leftrightarrow$) by new version of $\in$ computed above and the old version of $\in^*$ (to the right of $\leftrightarrow$).

It may be proved immediately that so defined simply computable transformations considered in terms of $\mathcal{EAG}^*$'s are DLOGSPACE-computable. However, we will demonstrate a more strong fact that these transformations are FO-*computable* (*definable*; cf. the definition in Section 2).

More precisely, consider the first-order language $\langle =, \in, \in^*, \emptyset, \{\emptyset\}, \bar{\tau}\rangle$ where symbols $=, \in, \in^*$ are interpreted as binary relations with the natural meaning in any $\mathcal{EAG}^*$ G so that $=_G$ is an identity or even a congruence relation wrt $\in_G$ (and therefore $\in^*_G$ and the constants $\emptyset, \{\emptyset\}, \bar{\tau} = \tau_0, \ldots, \tau_k$ are interpreted respectively as the "empty-set"-object, the "singleton-set"-object and arbitrary "set"-objects of G. (Without restriction of generality we may consider that transitive sets T considered above always contain elements $\emptyset$ and $\{\emptyset\}$.) Any simply computable transformation

$$G \mapsto G' : \mathcal{EAG}_k \to \mathcal{EAG}_m$$

[7] This may be considered as a *materialization* of $\in^*$ [4].

may be simulated by FO-formulas in this language denoted as $\bar{x} \approx \bar{y}$, $E(\bar{x};\bar{y})$, $E^*(\bar{x};\bar{y})$ and a list $\bar{\pi}_1, \ldots, \bar{\pi}_m$ of tuples consisting of the constants $\emptyset$, $\{\emptyset\}$, τ_1, …, τ_k. All these relations $\approx$, E and E^*, corresponding to $=_{G'}$, $\in_{G'}$ and $\in^*_{G'}$, are considered as binary relations over tuples $\bar{x}$, $\bar{y}$ with $\approx$ be a congruence wrt E, and E^* be the transitive closure of E. The list of tuples of constants $\bar{\pi}_1, \ldots, \bar{\pi}_m$ corresponds to the distinguished elements of the output graph G'.

Let, for example, we have one step transformation $[\![x;\bar{y}.\varphi(x,\bar{y})]\!] : G \mapsto G'$ with $\bar{y} = y_1, \ldots, y_s$. Proceed along the description of a SIMPLE transformer of $\mathcal{EAG}^*$'s given in the first paragraph of this section. Take $|G|^n$ for $n = s+1$ and assume that any n-tuple $\langle \bar{y}, z\rangle \in |G|^n$ denotes an "old" vertex $y_1 \in |G|$ if $z = \emptyset$ and $\langle \bar{y}, z\rangle$ denotes a "new" vertex $\bar{y}$ if $z \neq \emptyset$. Intuitively, any "new" object $\bar{y}$ represents the set $\{x \in |G| : G \models \varphi(x,\bar{y})\}$. Bearing in mind this interpretation of the tuples $\bar{y}$,z, define formulas E, E^* and $\approx$ as natural analogies and extensions of the relations $\in$, $\in^*$, and $=$:

$$
\begin{aligned}
E(\bar{y},z;\bar{y}',z') &\rightleftharpoons (z = z' = \emptyset \& y_1 \in y_1') \vee \\
&\quad\ (z = \emptyset \& z' \neq \emptyset \& \varphi(y_1,\bar{y}')) \vee \\
&\quad\ (z \neq \emptyset \& z' = \emptyset \& \exists u(\forall x(x \in u \Leftrightarrow \varphi(x,\bar{y})) \& u \in y_1')) \vee \\
&\quad\ (z \neq \emptyset \& z' \neq \emptyset \& \exists u(\forall x(x \in u \Leftrightarrow \varphi(x,\bar{y})) \& \varphi(u,\bar{y}'))),
\end{aligned}
$$

$$
\bar{y}, z \approx \bar{y}', z' \rightleftharpoons \forall \bar{u} v (E(\bar{u},v;\bar{y},z) \Leftrightarrow E(\bar{u},v;\bar{y}',z'))\ ,
$$

$$
\begin{aligned}
E^*(\bar{y},z;\bar{y}',z') &\rightleftharpoons E(\bar{y},z;\bar{y}',z') \vee \\
&\quad\ \exists w \exists u \in^* w(\bar{y},z \approx u,\ldots,\emptyset) \& E(w,\ldots,\emptyset;\bar{y}',z'))
\end{aligned}
$$

with $\bar{\pi}_1 \rightleftharpoons \bar{\tau}, \{\emptyset\}$ and $\bar{\pi}_{i+1} \rightleftharpoons \tau_i, \emptyset, \emptyset, \ldots, \emptyset, \emptyset$ for $1 \leq i \leq k$. (Compare the definition of E^* with the above recursive description of $\in^*$ in terms of $\in$.)

Additionally, we may define the embedding relation (mentioned above informally) between vertices of the original and resulting graph by FO-formula

$$\mathrm{Emb}(y;\bar{y},z) \rightleftharpoons (y = y_1 \ \&\ z = \emptyset).$$

Here y is a vertex in the original graph and the list $\bar{y}, z$ represents a vertex of the resulting graph.

5 Δ^* vs. SIMPLE

Theorem 2 *For any Δ^*-term or Δ^*-formula $t(\bar{y})$ there exists corresponding* SIMPLE *transformation*

$$\langle T;\bar{\tau}\rangle \mapsto \langle \tilde{T}; t(\bar{\tau})\rangle\ .$$

Conversely, any SIMPLY *computable set theoretic operation* $f(\bar{\tau})$ *(via corresponding transformation* $F : \langle T; \bar{\tau}\rangle \mapsto \langle T'; f(\bar{\tau})\rangle$ *where* $f(\bar{\tau})$ *depends only on* $\bar{\tau}$ *and not on* $T \ni \bar{\tau}$*) is* Δ^**-definable. Moreover, if* $\in^*$ *is not used in the definition of* F *then a weaker operation* $F'(T; \bar{\tau}) \rightleftharpoons (F(\langle T; \bar{\tau}\rangle))_2 = f(\bar{\tau})$ (with the dummy argument T for $\bar{\tau} \in T$) *is* Δ*-definable.*

Proof The converse part of the theorem is easy. Proposition 1 implies Δ^*-definability of $F(\langle T; \bar{\tau}\rangle)$ $(= \langle T'; f(\bar{\tau})\rangle)$. Therefore, $F'(T; \bar{\tau})$ $(= f(\bar{\tau}))$ is Δ'-definable. Finally, we get rid of the dummy argument T by taking $f(\bar{\tau}) = F'(\mathrm{TC}(\bar{\tau}); \bar{\tau})$.

Consider the direct part. The induction argument, to be successful, requires proving more

General Statement.[8] *For any list* $\bar{t}(\bar{y}) = t_1(\bar{y}), \ldots, t_n(\bar{y})$ *of* Δ^**-terms and* Δ^**-formulas* (considered as corresponding two-valued operations) *with* $\bar{y}$ *a list containing all free variables of* $\bar{t}$ *there exists corresponding* SIMPLE *transformation*

$$\langle T\rangle \mapsto \langle \tilde{T}; \lambda\bar{y} \in T.[\bar{t}(\bar{y})]\rangle$$

(with the graphs $\lambda\bar{y} \in T.[\bar{t}(\bar{y})]$ being elements of $\tilde{T}$).

Then the following Lemma will give the required result as formulated in Theorem 2.

Application Lemma. *For any m,n,k there exists* SIMPLE *transformation* $\mathrm{App}_{m,n,k}$ *such that for any set-theoretic* $(m+n+k)$*-ary function* $f = \lambda\bar{x}\bar{y}\bar{z} \in a.f(\bar{x}, \bar{y}, \bar{z})$*, with the arguments ranging over a set* a (and considered as a graph) *with the lists* $\bar{x}$*,*$\bar{y}$*,* $\bar{z}$ *containing respectively* m, n *and* k *variables we have:*

$$\mathrm{App}_{m,n,k} : \langle T; f, \bar{\tau}\rangle \mapsto \langle T'; \lambda\bar{x}\bar{z} \in a.f(\bar{x}, \bar{\tau}, \bar{z}), \bar{\tau}\rangle\ .$$

Proof of Application Lemma. The required transformation $\mathrm{App}_{m,n,k}$ is the composition of elementary transformation of the first kind

$$[\![w; u\bar{y}.\exists\bar{x}\bar{z}v(\langle\langle\bar{x}, \bar{y}, \bar{z}\rangle, v\rangle \in u \& w = \langle\langle\bar{x}, \bar{z}\rangle, v\rangle)]\!] :$$
$$\langle T; f, \bar{\tau}\rangle \mapsto \langle T'; \lambda\bar{x}\bar{z} \in a.f(\bar{x}, \bar{\tau}, \bar{z}), f, \bar{\tau}\rangle\ .$$

and the elementary transformation of the second kind, omitting the extra f from the distinguished elements of the output graph.

□

Proof of General Statement. If each $t_i(\bar{y})$ is a variable y_{j_i} then the required transformation

$$\langle T\rangle \mapsto \langle \tilde{T}, \lambda\bar{y} \in T.[y_{j_1}, y_{j_2}, \ldots, y_{j_m}]\rangle$$

[8] We need this General Statement especially for considering the case when $t(\bar{y})$ is $\{s(\bar{y}, x) : x \in a(\bar{y}) \& \varphi(\bar{y}, x)\}$ and, in particular, for showing that simply computable operations over HF are closed under arbitrary (not only unary) compositions.

is obtained (i) by using elementary transformation of the first kind:

$$[\![x;(x=x)]\!]:\langle T\rangle\mapsto\langle T\cup\{T\},T\rangle.$$

(which allows to consider the given transitive set T as a distinguished value, of a new transitive set $T\cup\{T\}$), (ii) by a saturation transformation (to have available in some further extension T' of $T\cup\{T\}$ all ordered tuples of the kind $\langle\bar{y},y_j\rangle$, $\bar{y}\in T$) and (iii) by using the elementary transformations $[\![z;\tau.\exists\bar{y}(\bar{y}\in\tau\&z=\langle\bar{y},y_{j_i}\rangle)]\!]$, $1\le i\le m$ (with the distinguished object T substituted for τ). This final step generates the required distinguished elements $\lambda\bar{y}\in T.y_{j_i}$. Of course, suitable elementary transformations of the second kind must be used in some intermediate steps.

Now, without lost of generality we may assume that $t_1(\bar{y})$ is not a variable (by using elementary transformations of the second kind). The following cases are possible.

Case 1 $t_1(\bar{y})$ is $\{a(\bar{y}),b(\bar{y})\}$. Then, by induction hypothesis, there exists a SIMPLE transformation

$$\langle T\rangle\mapsto\langle T';\lambda\bar{y}\in T.a(\bar{y}),\lambda\bar{y}\in T.b(\bar{y}),\lambda\bar{y}\in T.[t_2(\bar{y}),t_3(\bar{y}),\ldots]\rangle.$$

Note, that by $\lambda\bar{y}\in T.a(\bar{y}),\lambda\bar{y}\in T.b(\bar{y})\in T'$ and by transitivity of T' we have in T' all tuples of the kind $\langle\bar{y},a(\bar{y})\rangle,\langle\bar{y},b(\bar{y})\rangle$, and the elements $a(\bar{y}),b(\bar{y})$ for all $\bar{y}\in T$. By composing this transformation with a saturation transformation we will get $\langle T'';\lambda\bar{y}\in T.a(\bar{y}),\lambda\bar{y}\in T.b(\bar{y}),\lambda\bar{y}\in T.[t_2(\bar{y}),t_3(\bar{y}),\ldots]\rangle$. for some extension T'' of T' containing all pairs $\{a(\bar{y}),b(\bar{y})\}$ and ordered tuples of the kind $\langle\bar{y},\{a(\bar{y}),b(\bar{y})\}\rangle$. Finally, apply

$$[\![z;\alpha\beta.\exists\bar{y}uv[z=\langle\bar{y},\{u,v\}\rangle\&u=\alpha(\bar{y})\&v=\beta(\bar{y})]]\!],$$

with $\lambda\bar{y}\in T.a(\bar{y})$ and $\lambda\bar{y}\in T.b(\bar{y})$ substituted for α and β, and also suitable elementary transformation of the second kind. This gives the required transformation

$$\langle T\rangle\mapsto\langle\tilde{T};\lambda\bar{y}\in T.\{a(\bar{y}),b(\bar{y})\},\lambda\bar{y}\in T.[t_2(\bar{y}),t_3(\bar{y}),\ldots]\rangle.$$

with $T\subseteq T'\subseteq T''\subseteq\tilde{T}$. (From now on we will not mention trivial elementary transformations of the second kind such that were used above.)

Case 2 $t_1(\bar{y})$ is $\cup a(\bar{y})$. By induction hypothesis, there exists a SIMPLE transformation

$$\langle T\rangle\mapsto\langle T';\lambda\bar{y}\in T.a(\bar{y}),\lambda\bar{y}\in T.[t_2(\bar{y}),t_3(\bar{y}),\ldots]\rangle.$$

Compose it with

$$[\![x;u.(x\in\cup u)]\!]_0=[\![x;u.\exists v(x\in v\in u)]\!]_0:\langle T';\alpha,\ldots\rangle\mapsto\langle T'';\alpha,\ldots\rangle,$$

where $T'' = T \cup \{\cup u : u \in T'\}$ (with α intended to be $\lambda\bar{y} \in T.a(\bar{y})$), and then with a saturation transformation $\langle T''; \alpha, \ldots\rangle \mapsto \langle T'''; \alpha, \ldots\rangle$ and also with

$$[\![z; \alpha.\exists \bar{y}uw.[z = \langle \bar{y}, w\rangle \& u = \alpha(\bar{y}) \& w = \cup u]]\!] : \\ \langle T'''; \alpha, \ldots\rangle \mapsto \langle \tilde{T}; \lambda\bar{y} \in T. \cup \alpha(\bar{y}), \ldots\rangle.$$

This gives the required transformation

$$\langle T\rangle \mapsto \langle \tilde{T}; \lambda\bar{y} \in T. \cup a(\bar{y}), \lambda\bar{y} \in T.[t_1(\bar{y}), t_2(\bar{y}), \ldots]\rangle.$$

Case 3 $t_1(\bar{y})$ is $\mathrm{TC}(a(\bar{y}))$. By induction hypothesis, there exists a SIMPLE transformation

$$\langle T\rangle \mapsto \langle T'; \lambda\bar{y} \in T.[a(\bar{y}), t_2(\bar{y}), t_3(\bar{y}), \ldots]\rangle\ .$$

Compose it with $[\![x; u.(x \in^* u)]\!]$, so that all $\mathrm{TC}(u)$ with $u \in T$ will be available as elements, and then with a saturation transformation, which will give all members (tuples) of the set $\lambda\bar{y} \in T.\mathrm{TC}(a(\bar{y}))$, and, finally, with

$$[\![z; \alpha.\exists \bar{y}uv[z = \langle \bar{y}, v\rangle \& u = \alpha(\bar{y}) \& \forall l(l \in v \leftrightarrow l \in^* u)]]\!],$$

where $\lambda\bar{y} \in T.a(\bar{y})$ must be substituted for α. This gives the required transformation

$$\langle T\rangle \mapsto \langle \tilde{T}; \lambda\bar{y} \in T.[\mathrm{TC}(a(\bar{y})), t_2(\bar{y}), t_3(\bar{y}), \ldots]\rangle\ .$$

Case 4 $t_1(\bar{y})$ is $\{s(\bar{y}, x) : x \in a(\bar{y}) \& \varphi(\bar{y}, x)\}$. By induction hypothesis, there exist SIMPLE transformations

$$\langle T\rangle \mapsto \langle T'; \lambda\bar{y} \in T.[a(\bar{y}), t_2(\bar{y}), t_3(\bar{y}), \ldots]\rangle.$$

and

$$\langle T'; \alpha, t_2, t_3, \ldots\rangle \mapsto \langle T''; \lambda\bar{y}x \in T'.[s(\bar{y}, x), \varphi(\bar{y}, x)], \alpha, t_2, t_3, \ldots\rangle.$$

with $T \subseteq T' \subseteq T''$. We need these two steps because otherwise the variable x in the corresponding λ-prefix $\lambda\bar{y}x \in T$ possibly does not range over the whole set $a(\bar{y})$ when $\bar{y}$ range over T. Let us compose these transformations with transformations F_1, *saturation* and F_2, where

$$F_1 = [\![s; \sigma\phi\alpha\bar{y}.(s \in \{\sigma(\bar{y}, x) : x \in \alpha(\bar{y}) \& \phi(\bar{y}, x) = \emptyset\})]\!]_0$$

enriches T'' by all sets $\{s(\bar{y}, x) : x \in a(\bar{y}) \& \varphi(\bar{y}, x)\}$ such that $\bar{y}$ ranges over T (with the expression in the round brackets denoting the formula $\exists x(x \in \alpha(\bar{y}) \& \phi(\bar{y}, x) = \mathbf{true} \& s = \sigma(\bar{y}, x)))$, and

$$F_2 = [\![z; \sigma\phi\alpha.\exists \bar{y}\tau \\ (z = \langle \bar{y}, \tau\rangle \& \bar{y} \in T \& \tau = \{\sigma(\bar{y}, x) : x \in \alpha(\bar{y}) \& \phi(\bar{y}, x) = \emptyset\})]\!].$$

Here $\bar{y} \in T$ abbreviates and is equivalent to $\exists a(\langle \bar{y}, a\rangle \in \alpha)$ and $\tau = \{\ldots\}$ may be rewritten by extensionality and by using the above deciphering the formula $s \in \{\ldots\}$. This gives the required transformation

$$\begin{aligned}\langle T\rangle \mapsto \langle \tilde{T}; \lambda\bar{y} \in T.\{s(\bar{y}, x) : x \in a(\bar{y}) \& \varphi(\bar{y}, x)\},\\ \lambda\bar{y} \in T.[t_2(\bar{y}), t_3(\bar{y}), \ldots]\rangle\end{aligned}$$

(also by omitting the distinguished elements $\sigma_0, \phi_0 \rightleftharpoons \lambda\bar{y}x \in T'[s(\bar{y}, x), \varphi(\bar{y}, x)]$ and $\alpha_0 \rightleftharpoons \lambda\bar{y} \in T.a(\bar{y})$).

Case 5 $t_1(\bar{y})$ is a formula $\varphi(\bar{y})\&\psi(\bar{y})$. By induction hypothesis, there exists a SIMPLE transformation

$$\langle T\rangle \mapsto \langle T'; \lambda\bar{y} \in T.[\varphi(\bar{y}), \psi(\bar{y}), t_2(\bar{y}), t_3(\bar{y}), \ldots]\rangle.$$

By composing this transformation with a saturation transformation and then with

$$\begin{aligned}[\![z; \alpha\beta.\exists\bar{y}[(z = \langle \bar{y}, \mathbf{true}\rangle \& \alpha(\bar{y}) = \mathbf{true} \& \beta(\bar{y}) = \mathbf{true}) \vee\\ (z = \langle \bar{y}, \mathbf{false}\rangle \& \neg(\& \alpha(\bar{y}) = \mathbf{true} \& \beta(\bar{y}) = \mathbf{true}))]]\!]\end{aligned}$$

we obtain the required transformation

$$\langle T\rangle \mapsto \langle \tilde{T}; \lambda\bar{y} \in T.[\varphi(\bar{y})\&\psi(\bar{y}), t_2(\bar{y}), t_3(\bar{y}), \ldots]\rangle.$$

Case 6 $t_1(\bar{y})$ is a formula $\forall x \in a(\bar{y}).\varphi(\bar{y}, x)$. By induction hypothesis, there exist SIMPLE transformations

$$\langle T\rangle \mapsto \langle T'; \lambda\bar{y} \in T.[a(\bar{y}), t_2(\bar{y}), t_3(\bar{y}), \ldots]\rangle.$$

and

$$\langle T'; \alpha, t_2, t_3, \ldots\rangle \mapsto \langle T''; \lambda\bar{y}x \in T'.\varphi(\bar{y}, x), \alpha, t_2, t_3, \ldots\rangle.$$

with $T \subseteq T' \subseteq T''$. By composing these transformations with a saturation transformation and then with

$$\begin{aligned}[\![z; \phi\alpha.\exists\bar{y}[(z = \langle \bar{y}, \mathbf{true}\rangle \& \forall x(x \in \alpha(\bar{y}) \Rightarrow \phi(\bar{y}, x) = \mathbf{true})) \vee\\ (z = \langle \bar{y}, \mathbf{false}\rangle \& \neg\forall x(x \in \alpha(\bar{y}) \Rightarrow \phi(\bar{y}, x) = \mathbf{true}))]]\!]\end{aligned}$$

we obtain (by omitting the distinguished elements $\phi_0 \rightleftharpoons \lambda\bar{y}x \in T'.\varphi(\bar{y}, x)$ and $\alpha_0 \rightleftharpoons \lambda\bar{y} \in T.a(\bar{y})$) the required transformation

$$\langle T\rangle \mapsto \langle \tilde{T}; \lambda\bar{y} \in T.[\forall x \in a(\bar{y})\varphi(\bar{y}, x), t_2(\bar{y}), t_3(\bar{y}), \ldots]\rangle.$$

Case 7 The cases if $t_1(\bar{y})$ is is a formula $a(\bar{y}) \in b(\bar{y})$, $\varphi(\bar{y}) \vee \psi(\bar{y})$, $\neg\varphi(\bar{y})$ or $\exists x \in a(\bar{y}).\varphi(\bar{y}, x)$ are analogous to the considered above.

$\square$

$\square$

6 $\Delta^{\otimes}$ vs. SIMPLE$^{\otimes}$

Now we will consider $\Delta^{\otimes}$-definability over HF and respectively generalized class of SIMPLE$^{\otimes}$-transformers. Here, instead of FO*-formulas, i.e. formulas in the language FO$(\in, \in^*, =)$ we will use formulas in the language FO$^{\otimes}(\in, =)$. We need not $\in^*$ because it is definable as $\in^{\otimes}$.

As above, it may be proved that each SIMPLE$^{\otimes}$ transformation of $\mathcal{EAG}$'s is NLOGSPACE-computable (instead of DLOGSPACE for SIMPLE transformations of the previous section) and, moreover, FO$^{\otimes}$-*computable* (*definable*). Then Theorem 2 also can be extended to the present case.

Theorem 3 *For any list* $\bar{t}(\bar{y}) = t_1(\bar{y}), t_2(\bar{y}), \ldots, t_n(\bar{y})$ *of* $\Delta^{\otimes}$*-terms and* $\Delta^{\otimes}$*-formulas there exists corresponding* SIMPLE$^{\otimes}$ *transformation*

$$\langle T\rangle \mapsto \langle T'; \lambda\bar{y} \in T.[\bar{t}(\bar{y})]\rangle$$

(with $\lambda\bar{y} \in T.[t_i(\bar{y})]$ elements of T'), *and therefore* (by Application Lemma as in Theorem 2) *there exists* SIMPLE$^{\otimes}$ *transformation*

$$\langle T; \bar{\tau}\rangle \mapsto \langle \tilde{T}; [\bar{t}(\bar{\tau})]\rangle\ .$$

Conversely, any SIMPLE$^{\otimes}$*-computable set theoretic operation* $f(\bar{\tau})$ *(via some* SIMPLE$^{\otimes}$ *transformation* $F : \langle T; \bar{\tau}\rangle \mapsto \langle T'; f(\bar{\tau})\rangle$*) is* $\Delta^{\otimes}$*-definable. To define a weaker operation* $F'(T; \bar{\tau}) \rightleftharpoons f(\bar{\tau})$ (with the dummy argument T) for $\bar{\tau} \in T$ *we need not use* $*$ *(i.e.* TC*) in* $\Delta^{\otimes}$*.*

Proof The converse part of the theorem is evident because any FO$^{\otimes}$-formula over a transitive set T may be replaced by the $\otimes$-construct of $\Delta^{\otimes}$-language applied to the result of suitable $\Delta^{\otimes}$-separation construct. Again, as in Theorem 2, the equality $F(\bar{\tau}) = F'(\mathrm{TC}(\bar{\tau}), \bar{\tau})$ is used in the final step.

The proof of the direct part essentially consists in adding to the proof of Theorem 2

Case $t_1(\bar{y})$ is $r^{\otimes}(\bar{y})$. By induction hypothesis, there exists a computable transformation

$$\langle T\rangle \mapsto \langle T'; \lambda\bar{y} \in T.[r(\bar{y}), t_2(\bar{y}), t_3(\bar{y}), \ldots]\rangle\ .$$

Then compose it with $[\![x; u.\exists lm.(x = \langle l, m\rangle \& [\langle l, m\rangle \in u]^{\otimes})]\!]$ and

$$[\![z; \rho.\exists zw\bar{y}u.$$
$$[z = \langle \bar{y}, w\rangle \& u = \rho(\bar{y}) \& \forall x.(x \in w \Leftrightarrow \exists lm.(x = \langle l, m\rangle \& [\langle l, m\rangle \in u]^{\otimes}))]\!].$$

This will give the required transformation

$$\langle T\rangle \mapsto \langle \tilde{T}; \lambda\bar{y} \in T.[r^{\otimes}(\bar{y}), t_2(\bar{y}), t_3(\bar{y}), \ldots]\rangle$$

where $\tilde{T} = T' \cup \{$all pairs $\langle u, v\rangle$ from all transitive closures of all sets of pairs $\in T'\} \cup \ldots$.

$\square$

7 $\Delta^{\otimes} + \mathrm{C}$ vs. $\mathrm{FO}^{\otimes}$

So, we have described $\Delta^{(\otimes)}$-definability in terms of SIMPLE$^{(\otimes)}$ graph transformers which are also FO$^{(\otimes)}$-definable. Of course, the languages FO and FO$^{\otimes}$ are more natural, especially because $\mathrm{FO}^{\otimes}_{\prec}$ corresponds to the complexity class NLOGSPACE. In this section we consider a version of Δ which may be exactly characterized in terms of FO$^{\otimes}$.

Let consider rather *special* extension $\Delta^{\otimes} +$ *external* C of $\Delta^{\otimes}$-language by the *collapsing* operation $\mathrm{C}(\langle g, v\rangle)$ (cf. Section 2) such that it may occur in any term t only as its *head* (i.e. as $t = \mathrm{C}(t')$ with no C in t'), if any. It is an *open question* whether this restriction is indeed essential for the expressive power of the full $\Delta^{\otimes} + \mathrm{C}$. Unfortunately, only for the above restricted version of this language we are able to get the following characterization.

Theorem 4 *The class of* FO$^{\otimes}$*-computable operations* $f : \mathrm{HF}^n \to \mathrm{HF}$ *(via corresponding* FO$^{\otimes}$*-definable transformations* $F : \langle T; \bar\tau\rangle \mapsto \langle\tilde T; f(\bar\tau)\rangle$*) coincides with the class of operations definable in the language* $\Delta^{\otimes} +$ *external* C.

Proof For any tuple $\bar t$ of terms in $\Delta^{\otimes}$ (with no C) FO$^{\otimes}$-computability of the corresponding operations was considered in Theorem 3 and in the immediately preceding note (cf. also Theorem 2 and Section 4.1). The case of operation $\mathrm{C}(\tau)$ is treated quite straightforward. In terms of the graph $\langle T; \tau\rangle$ the resulting graph $\langle\tilde T; \mathrm{C}(\tau)\rangle$ is defined, up to isomorphism, by two FO$^{\otimes}$-formulas $\mathcal{T}(a, b) = \mathcal{T}(T, \tau; a, b)$ (for the graph $\tilde T$ with the edges $\langle a, b\rangle$) and $\mathcal{C}(c) = \mathcal{C}(T, \tau; c)$ (for the distinguished vertex c corresponding to the set $\mathrm{C}(\tau)$) defined by the following steps according to the definition of C:

$$H(a, b) \rightleftharpoons (\exists gv(\langle g, v\rangle = \tau \& \langle a, b\rangle \in g)^T,$$

$$\mathrm{Ext}[H] \rightleftharpoons (\forall bb'(\forall a(H(a, b) \Leftrightarrow H(a, b')) \Rightarrow b = b'))^T,$$

$$\mathrm{Acycl}[H] \rightleftharpoons (\neg\exists a H^{\otimes}(a, a))^T,$$

$$\mathcal{T}(a, b) \rightleftharpoons \mathrm{Ext}[H] \& \mathrm{Acycl}[H] \& H(a, b),$$

$$\mathcal{C}(c) \rightleftharpoons (\exists g(\langle g, c\rangle = \tau))^T.$$

It remains to note, that if some terms $\bar t = t_1, \ldots, t_n$ have a *unique* corresponding FO$^{\otimes}$-definable transformation $\langle T; \bar\tau\rangle \mapsto \langle\tilde T; \bar t(\bar\tau)\rangle$ and an operation $g(x_1, \ldots, x_n)$ also has corresponding such transformation then the same holds for the composition $t = g(t_1, \ldots, t_n)$. Finally, take $g = \mathrm{C}$ and $n = 1$.

Conversely, let $f(\bar\tau)$ be any n-ary operation over sets FO$^{\otimes}$ -computable by a graph transformer F. We consider that $F : \mathrm{HF} \to \mathrm{HF}$. Then it can be represented as the following ($\Delta^{\otimes}$ + *external* C)-definable operation, in fact a composition (applied from the right to the left)

$$f = \mathrm{C} \circ F \circ G$$

where G is an n-ary Δ^*-definable operation over sets transforming any n input sets $x_1,\dots,x_n$ into their well-founded extensional graph representation (with the distinguished n-tuple of vertices):

$$G(x_1,\dots,x_n) = \langle \in\!|_{\mathrm{TC}(\{x_1,\dots,x_n\})}; x_1,\dots,x_n\rangle\ ,$$

F is evidently also $\Delta^{\otimes}$-definable by imitating in $\Delta^{\otimes}$ its $\mathrm{FO}^{\otimes}$-description. (Input graph for F is considered as set of a specific kind; corresponding output graph is defined by using Δ-definable notion of cartesian product and by imitating unbounded quantifiers of $\mathrm{FO}^{\otimes}$ and the logical construct $\otimes$ by bounded quantifiers and set-theoretic version of $\otimes$ in $\Delta^{\otimes}$.) Finally, collapsing operation C serves to 'extract' a resulting set from its graph representation.

□

8 Further Results

Remember, that *any* class of finite pointed graphs $\mathcal{G}$ defines corresponding graph encoding of HF-sets with Codes = $\mathcal{G}$. Given any Δ' and class(es) of TRANSFORMERS[9] of graphs from $\mathcal{G}_1$ to $\mathcal{G}_2$, we may introduce the following abbreviation

$$\Delta' \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{G}_1 \xrightarrow[\text{TRANSFORMERS}_2]{\text{TRANSFORMERS}_1} \mathcal{G}_2\right)$$

of the statement: '*Δ'-definable operations over* HF *coincide with those computable by* TRANSFORMERS$_i$, $i = 1,2$'. One of the classes of transformers may be omitted in this denotation. If both are present then it is asserted that they compute the *same* class of operations over HF, namely those Δ'-definable (even if these classes of TRANSFORMERS are different).

Then we may formulate the above Theorems 1–4 and a number of additional results as follows:

$$\Delta^* + \mathbf{Rec} + \mathrm{C} \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG} \xrightarrow[\text{PTIME}]{\text{FO + LFP}} \mathcal{EAG}\right) \tag{1}$$

$$\Delta^* \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}^* \xrightarrow{\text{SIMPLE}} \mathcal{EAG}^*\right) \tag{2}$$

$$\Delta^{\otimes} \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}^{(*)} \xrightarrow{\text{SIMPLE}^{\otimes}} \mathcal{EAG}^{(*)}\right) \tag{3}$$

$$\Delta^{\otimes} + \textit{external}\ \mathrm{C} \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}^{(*)} \xrightarrow[\text{?NLOGSPACE?}]{\text{FO}^{\otimes}} \mathcal{EAG}^{(*)}\right) \tag{4}$$

[9] Strictly speaking, each concrete transformer must have a type $m \to n$ where natural numbers m and n denote the number of distinguished vertices in the input and, respectively, output graphs. Usually we take $n = 1$. Then such a transformer may compute an m-place operation $\mathrm{HF}^m \to \mathrm{HF}$ (if corresponding square diagram commutes).

$$\Delta^*_< \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}^*_< \xrightarrow{\text{SIMPLE}} \mathcal{EAG}^*_<\right) \tag{5}$$

$$\Delta^\otimes_< \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}^{(*)}_< \xrightarrow{\text{SIMPLE}^\otimes} \mathcal{EAG}^{(*)}_<\right) \tag{6}$$

$$\Delta^* + \mathit{external}\ \tilde{\mathrm{C}} \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}^* \xrightarrow{\text{FO}} \mathcal{EG}\right) \tag{7}$$

$$\Delta^*_< + \mathit{external}\ \tilde{\mathrm{C}} \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}^*_< \xrightarrow{\text{FO}} \mathcal{EG}_\prec\right) \tag{8}$$

$$\Delta^\otimes_< + \mathit{external}\ \mathrm{C} \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}^{(*)}_< \xrightarrow[\text{NLOGSPACE}]{\text{FO}^\otimes} \mathcal{EAG}^{(*)}_\prec\right) \tag{9}$$

$$\Delta^\otimes_< + \mathit{external}\ \hat{\mathrm{C}} \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}^{(*)}_< \xrightarrow[\text{NLOGSPACE}]{\text{FO}^\otimes} \mathcal{EAG}^{(*)}_<\right) \tag{10}$$

Here brackets surrounding the superscript $*$ to '$\mathcal{EAG}$' mean that $*$ may be equally omitted because its presence does not matter for the case of SIMPLE$^\otimes$ or FO$^\otimes$-transformers. Question marks in (4) denote that it is an *open question* whether NLOGSPACE may indeed stay there by analogy with (1) (despite it is known that the class of FO$^\otimes$-transformations of unordered graphs is properly contained in those of the complexity NLOGSPACE, as well as for FO + LFP and PTIME in (1)).

Instead of proving each of these results we give some brief comments sufficient to recover all the details. The proof (and formulations) of (5) and (6) extends those of (2) and (3) (i.e. of Theorems 2 and 3):

- by including in the language of SIMPLE$^\otimes$ transformers the predicate relation $<$ for the canonical linear order,
- by noting that $<_{\mathrm{HF}}$ becomes trivially SIMPLE$^\otimes$ computable because, at present, the input graphs must involve corresponding ordering $<$ and
- by FO-definable extending the relation $<$ from the input to the output graph (say, under any elementary transformation) according to the Axiom for $<_{\mathrm{HF}}$ in Section 3.

Analogously, the proofs of (7–10) are obtained from that of (4) (i.e. of Theorem 4). So, as in the case of FO$^\otimes$-formulas defining the operation C, it is respectively definable a natural graph transformer $\langle G, p\rangle \mapsto \langle G', p'\rangle$ computing C (or $\tilde{\mathrm{C}}$). It just extracts from G a "horizontal" graph $\langle G', p'\rangle$ defined by the vertex p as an "ordered pair" consisting of a "set of pairs (edges)" and a "point" (in the sense relativized to G) corresponding, respectively to $\in_{G'}$ and p'. However, in the case of (8) or (9), unlike (10), while computing the collapsing C or $\tilde{\mathrm{C}}$ by such a graph transformer, we are able to obtain only *some* linear order $\prec$ on the resulting graph (whose vertices comprise a subset of those of the input graph) by the corresponding restriction of $<$. Unfortunately, it is unknown to the authors how to transform in NLOGSPACE any linear ordering $\prec$ on $\mathcal{EAG}$

to the canonical one. A difference between the types of input and output graphs is the price for the result like (9) in comparison with (1–6) and (10).

In the cases of (7,8 and 9) we cannot realize the collapsing operation C or $\tilde{C}$ by a graph transformer of the type $\mathcal{EAG}^*_{(<)} \to \mathcal{EAG}^*_{(<)}$ because in FO and, respectively, in $FO^{\otimes}$ we cannot define $*$ and, respectively, canonical order $<$ and check in FO whether a graph is acyclic. However, we can easily check extensionality property of a graph in FO to recognize whether it belongs to the (proper) domain $\mathcal{EG} \subseteq$ HF of $\tilde{C}$, before passing this graph to the output.

In (9,10) we actually have an identity of the classes $FO^{\otimes}$ = NLOGSPACE of corresponding graph transformers due to the presence of a linear order on $\mathcal{EAG}_<$'s. Therefore we could say that $\Delta^{\otimes}_<$ + *external* C ($\hat{C}$) corresponds to NLOGSPACE-computability over HF, if to ignore an essential drawback that using the collapsing operation has a strange restriction in the language. (On the other hand, it seems that in real applications we need to use collapsing not so frequently.) Evidently, $\hat{C}$ is expressible in $\Delta^{\otimes}_<$ + *external* C (by using a Δ-operation which "forgets" the linear order on $\mathcal{EAG}_<$'s) and the class of operations over HF from (10) is contained in that from (9). We do not know whether this inclusion is proper.

Only for (1,2,3,5,6) we can guarantee that corresponding classes of set-theoretic operations are closed under composition without any restrictions. However, in these cases, except (1), graph transformers do not constitute any computational complexity class. In (4) and (10) the closure under composition may be shown only for one-place operations because it is unknown to the authors whether, given several $\mathcal{EAG}_{(<)}$'s, we can $FO^{\otimes}$-define a *unique* $\mathcal{EAG}$ with transitive subgraphs isomorphic to the original $\mathcal{EAG}$'s (that is we do not know how to identify by the allowed means the vertices of different $\mathcal{EAG}_{(<)}$'s which denote the same HF-set).

If we are interested only in NLOGSPACE-computable *predicates* over HF, then we need no collapsing and have the following version of (6,9,10):

$$\Delta^{\otimes}_< \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}^{(*)}_< \xrightarrow[\text{NLOGSPACE}]{\text{SIMPLE}^{\otimes}/\text{FO}^{\otimes}} \{\mathbf{true,\ false}\}\right) \tag{11}$$

where for this kind of transformers $\text{SIMPLE}^{\otimes} \sim_{\mathrm{HF}} \text{FO}^{\otimes}$ = NLOGSPACE due to the presence of linear order.

It seems that (6) and (10) are the most natural solutions in this Section of our initial problem to find a version of Δ-language defining a class of set-theoretic operations (queries) which constitute a natural Sub-PTIME complexity class. This is witnessed in the case of (6) by the statement (11), by $FO^{\otimes}$-decidability whether a graph is an $\mathcal{EAG}^{(*)}_<$ (with the canonical ordering $<$; cf. also Note 4 below for a proof of even more strong statement) and by good closure properties. A witness in the case of (10) is a description of "all" NLOGSPACE-computable set-theoretic operations (under corresponding encoding of HF-sets by $\mathcal{EAG}_<$'s) so that *one-place* operations are closed under compositions. Of course both these descriptions appear *not sufficiently satisfactory* in comparison with (1). However, Section 9 gives yet another and more natural approach, intermediate between those of (6) and (10).

9 Main Result for NLOGSPACE

Let us consider a class of so called IC-FO$^{\otimes}$-*transformers* of $\mathcal{EAG}_<$'s (or, equivalently, IC-NL-*transformers*)

$$\left(\mathcal{EAG}_< \xrightarrow[\text{IC-NL}]{\text{IC-FO}^{\otimes}} \mathcal{EAG}_<\right).$$

(Of course, $\mathcal{EAG}^*_<$ could be equivalently used here, because $\otimes$ is available.) This is the class of FO$^{\otimes}$-transformers which are additionally (FO$^{\otimes}$-)*inflationary* and (FO$^{\otimes}$-)*coherent*. (IC-NL-transformers are defined in the same way and evidently must coincide with IC-FO$^{\otimes}$-ones due to coincidence of NLOGSPACE and FO$^{\otimes}$ on linear ordered finite structures.) Here 'inflationary' means the following:

> Given any input $\mathcal{EAG}_<$-graph G, it is FO$^{\otimes}$-definable (in terms of G) not only the corresponding output $\mathcal{EAG}_<$-graph G' (with G' a subset of some cartesian power $|G|^k$), but also an *auxiliary* FO$^{\otimes}$-definable $\in$-isomorphism Emb_G of G onto a transitive subgraph of G' (like that in Section 4.1). Moreover, *only* the distinguished vertices of the output graph G', but not the graph itself, may depend syntactically on the distinguished vertices of the input graph G (as in the definition of elementary transformers $[\![x;\bar{y}.\varphi]\!]$).
>
> Instead of the abstract graphs G and described transformers we may consider, as above, corresponding to them $\in$-isomorphic canonically ordered non-empty transitive sets $T = T[G]$ in HF and transformers
>
> $$F : \langle T;\bar{\tau}\rangle \mapsto \langle F_0(T); F_1(\bar{\tau}), F_2(\bar{\tau}), \ldots\rangle$$
>
> with $\bar{\tau}$ and $\bar{F}(\bar{\tau})$ corresponding lists of the distinguished vertices so that $\langle T'; \in|_{T'}, <|_{T'}, =|_{T'}\rangle$, for $T' \rightleftharpoons F_0(T) = T[G']$, is isomorphic to (a quotient of) some FO$^{\otimes}$-definable structure in terms of $\langle T; \in|_T, <|_T, =|_T\rangle$ on a cartesian power T^k and there exists an FO$^{\otimes}$-definable embedding of $\langle T, \in|_T, <|_T, =|_T\rangle$ into $\langle T', \in|_{T'}, <|_{T'}, =|_{T'}\rangle$ (in fact, in T^k).

'Coherent' means existence of one more *auxiliary* FO$^{\otimes}$-construction:

> Let we have any two input $\mathcal{EAG}_<$-graphs G_1 and G_2 with a linear preorder[10] relation $\leq_{G_1+G_2}$ on $|G_1| + |G_2|$ corresponding to the

[10] We cannot use here an *order* relation because some vertices in G_1 must be identified with those in G_2 if they denote the same sets in HF. Also it would be insufficient for the

canonical linear order on the ordinary union of transitive sets $T[G_1] \cup T[G_2]$ in HF. Then analogous relation $\leq_{G'_1+G'_2}$ on $G'_1 + G'_2$ is also FO$^{\otimes}$-definable in terms of two-sorted structure $\langle G_1 + G_2; \leq_{G_1+G_2}\rangle$. Again, *no* syntactic dependence on the distinguished vertices of G_1 and G_2 is allowed.

As usually, IC-FO$^{\otimes}$-transformers of $\mathcal{EAG}_<$'s define a class of respectively computable operations over HF which we denote as $\mathcal{IC}$-$\mathcal{NL}$ or as $\mathcal{IC}$-$\mathcal{FO}^{\otimes}_<$. A natural motivation for introducing these technical notions is given in the following theorem.

Let Δ' be any extension of the language $\Delta^*_<$ by a family of operations over HF with no such restriction on the syntax as in the case of *external* C.

Theorem 5 *Suppose that all Δ'-definable operations $q : \mathrm{HF} \to \mathrm{HF}$ are computable by some* FO$^{\otimes}$*-definable transformers between $\mathcal{EAG}^*_<$'s. Then they are also computable by* IC-FO$^{\otimes}$*-transformers.*[11]

Proof For any Δ'-definable q it is evidently definable in $\Delta^*_<+q \subseteq \Delta'$ an operation

$$\tilde{Q} : \langle T; <_T, \in_T, \in^*_T\rangle \mapsto \langle T'; <_{T'}, \in_{T'}, \in^*_{T'}, \lambda\tau \in T.q(\tau)\rangle : \mathrm{HF} \to \mathrm{HF}$$

with $T' \rightleftharpoons \mathrm{TC}(\{\lambda\tau \in T.q(\tau)\})$ transitive sets (so that $\lambda\tau \in T.q(\tau) \in T'$ and $T \subseteq T'$). Therefore, under condition of the theorem, $\tilde{Q}$ must be computable (up to isomorphism) by some FO$^{\otimes}$-definable transformer between $\mathcal{EAG}^*_<$'s:

$$\langle \mathcal{T}; <_{\mathcal{T}}, \in_{\mathcal{T}}, \in^*_{\mathcal{T}}, \langle T; \ldots\rangle\rangle \mapsto \langle \mathcal{T}'; <_{\mathcal{T}'}, \in_{\mathcal{T}'}, \in^*_{\mathcal{T}'}, \langle T'; \ldots, \lambda\tau \in T.q(\tau)\rangle\rangle$$

where we consider that $\mathcal{T}$ and $\mathcal{T}'$ are transitive sets in HF, $\langle T; \ldots\rangle \in \mathcal{T}$ and $\langle T'; \ldots, \lambda\tau \in T.q(\tau)\rangle \in \mathcal{T}'$. This implies that $\tilde{Q}$ is also FO$^{\otimes}$-definable (up to isomorphism) *as a graph transformer*. Just take $\mathcal{T} = \mathrm{TC}(T \cup \{\langle T; \ldots\rangle\})$, define it and $\langle \mathcal{T}; <_{\mathcal{T}}, \in_{\mathcal{T}}, \in^*_{\mathcal{T}}, \langle T; \ldots\rangle\rangle$ by means of FO in terms of $\langle T; \ldots\rangle$, compose this FO-definition with FO$^{\otimes}$-definition of $\langle \mathcal{T}'; \ldots, \langle T'; \ldots, \lambda\tau \in T.q(\tau)\rangle\rangle$ and, finally, extract T', $T = \mathbf{dom}(\lambda\tau \in T.q(\tau))$, etc. by means of FO from $\langle \mathcal{T}'; \ldots, \langle T'; \ldots, \lambda\tau \in T.q(\tau)\rangle\rangle$. In particular, T is FO$^{\otimes}$-definably embedded in T'. Therefore, graph transformer

$$Q : \langle T; <|_T, \in|_T, \in^*_T, \tau\rangle \mapsto \langle T'; <|_{T'}, \in|_{T'}, \in^*_{T'}, q(\tau)\rangle,$$

(which is almost the same as $\tilde{Q}$ and FO-definable relative to $\tilde{Q}$) is inflationary FO$^{\otimes}$-definable, and it also evidently computes q.

following considerations to use in this definition corresponding equivalence relations $\equiv_{G_1+G_2}$ and $\equiv_{G'_1+G'_2}$ instead of $\leq_{G_1+G_2}$ and $\leq_{G'_1+G'_2}$. (Cf. also Section 13 below.) Note, that the linear orders $<_{G_1}$ and $<_{G_2}$ on each of these graphs may be recovered from $\leq_{G_1+G_2}$ by restricting it, respectively, to $|G_1|$ and $|G_2|$.

[11] The same holds also if to replace here 'FO$^{\otimes}$' by 'FO$^{\odot}$' or 'FO*', or even by 'FO', and also take Δ^* and $\mathcal{EAG}^*$'s instead of $\Delta^*_<$ and $\mathcal{EAG}^*_<$'s, and, respectively, replace $\leq$ by $\equiv$ in the definition of coherence property; cf. also Section 12.

By analogous consideration we can show that Q is coherent. To this end consider the following transformer

$$\tilde{\tilde{Q}} : \langle T_1 + T_2; \leq_{T_1+T_2}, \in_{T_1}, \in_{T_2}, \in^*_{T_1}, \in^*_{T_2}\rangle \mapsto \langle T'_1 + T'_2; \leq_{T'_1+T'_2}\rangle$$

definable in $\Delta^*_< + q \subseteq \Delta'$ with the operation $T \mapsto T'$ the same as above. Again, $\tilde{\tilde{Q}}$ must be computable by some FO$^{\otimes}$-definable transformer between $\mathcal{EAG}^*_<$'s:

$$\langle \mathcal{T}; <_{\mathcal{T}}, \in_{\mathcal{T}}, \in^*_{\mathcal{T}}, \langle T_1 + T_2; \leq_{T_1+T_2}, \ldots\rangle\rangle \mapsto \\ \langle \mathcal{T}'; <_{\mathcal{T}'}, \in_{\mathcal{T}'}, \in^*_{\mathcal{T}'}, \langle T'_1 + T'_2; \leq_{T'_1+T'_2}, \ldots\rangle\rangle$$

with $\langle T_1 + T_2; \ldots\rangle \in \mathcal{T}$ and $\langle T'_1 + T'_2; \ldots\rangle \in \mathcal{T}'$. By taking $\mathcal{T} = \mathrm{TC}(T_1 \cup T_2 \cup \{\langle T_1 + T_2; \leq_{T_1+T_2}, \ldots\rangle\})$, etc., this imply that $\tilde{\tilde{Q}}$ is also FO$^{\otimes}$-definable *as a graph transformer*. This is just the coherence property of Q.

□

That is why it is inevitable to consider here inflationary and coherent FO$^{\otimes}_<$-definable ((N)LOGSPACE-computable) transformers

It is not difficult to prove the following auxiliary

Proposition 2 *(a) The class of* IC-FO$^{\otimes}$*-transformers is closed under compositions.*

(b) The class of one-place $\mathcal{IC}$-$\mathcal{NL}$-operations is closed under compositions.

(c) SIMPLE$^{\otimes}$ $\subseteq$ IC-FO$^{\otimes}$.

(d) $\Delta^{\otimes}_<$*-definable operations over* HF *are contained in $\mathcal{IC}$-$\mathcal{NL}$ (cf. (6) and (c)).*

□

The clauses (b) and (d) are also partial cases of the statement (12) below which means, in particular, that $\mathcal{IC}$-$\mathcal{NL}$ is closed under $\Delta^{\otimes}_<$-constructs and therefore under *arbitrary* compositions (not only for one-place operations, as in (b)).

The following proposition plays a crucial role in our searching for a reasonable notion of NLOGSPACE-computability over HF and in proving the corresponding Main Result (the statement (12)) below.

Proposition 3 *Any* IC-FO$^{\otimes}$*-transformer*

$$F : \langle T, <; \bar{\tau}\rangle \mapsto \langle F_0(T), <; F_1(T, \bar{\tau}), F_2(T, \bar{\tau}), \ldots\rangle : \mathrm{HF} \to \mathrm{HF}$$

is an $\mathcal{IC}$-$\mathcal{NL}$-operation (i.e. such operation which is itself computable by a similar transformer).

The class of such operations F over HF is denoted as $\mathcal{IC}$-$\mathcal{NL}_0$.[12]

[12]Essentially, $\mathcal{IC}$-$\mathcal{NL}_0$ coincides with IC-FO$^{\otimes}$ = IC-NL, if to consider that input and output graphs of any IC-FO$^{\otimes}$-transformer are just HF-sets of a special kind. However, even in this case we must carefully use different notations for them because they play different roles: as a special class of set-theoretic operations and as a class of graph transformers which may compute some other set-theoretic operations.

Proof We only outline how the operation $T \mapsto T' = F_0(T)$ is computable by some inflationary FO$^{\otimes}$-transformer

$$\langle \mathcal{T}, <; T\rangle \mapsto \langle \tilde{\mathcal{T}}, <; T'\rangle$$

with transitive $\mathcal{T}$, $\tilde{\mathcal{T}}$, $T \in \mathcal{T}$, $T' \in \tilde{\mathcal{T}}$ and $\mathcal{T} \subseteq \tilde{\mathcal{T}}$. In fact, we could show that this transformer is also FO$^{\otimes}$-coherent and, moreover, can be extended to the required one which computes the *whole* operation F. We left this (and also recovering other lacking details) to the reader as an exercise.

So, we must FO$^{\otimes}$-define in terms of $\langle \mathcal{T}; \in\!|_{\mathcal{T}}, <\!|_{\mathcal{T}}, =\!|_{\mathcal{T}}\rangle$ (in a suitable cartesian power of $\mathcal{T}$, up to a generalized isomorphism) the transitive set $\tilde{\mathcal{T}}$ together with FO$^{\otimes}$-definable relations $\in\!|_{\tilde{\mathcal{T}}}$, $<\!|_{\tilde{\mathcal{T}}}$, $=\!|_{\tilde{\mathcal{T}}}$ and embedding $\mathcal{T} \hookrightarrow \tilde{\mathcal{T}}$. First, consider transitive sets $\mathcal{T}' = F_0(\mathcal{T}) \supseteq \mathcal{T}$ and all $T' = F_0(T) \supseteq T$, for $T \in \mathcal{T}$ any transitive set of cardinality ≥ 2. Then take

$$\hat{\mathcal{T}} \rightleftharpoons \mathcal{T}' \cup (\bigcup \{T' : T \in \mathcal{T} \text{ transitive set of cardinality} \geq 2\})$$

and

$$\tilde{\mathcal{T}} \rightleftharpoons \hat{\mathcal{T}} \cup \{\{T'\} : T \in \mathcal{T} \text{ transitive set of cardinality} \geq 2\}$$

as the ordinary set-unions of HF-sets. Both transitive sets $\hat{\mathcal{T}}$ and $\tilde{\mathcal{T}}$ are FO$^{\otimes}$-definable as follows.

First, by IC-FO$^{\otimes}$-definability of the given transformer F, some isomorphic copies of $\mathcal{T}'$ and of each T' may be considered as FO$^{\otimes}$-definable, *uniformly on* $\langle \mathcal{T}; \in\!|_{\mathcal{T}}, <\!|_{\mathcal{T}}, =\!|_{\mathcal{T}}, T\rangle$, $T \in \mathcal{T}$, subsets of the same finite power $\mathcal{T}^k$ of $\mathcal{T}$:

$$\mathcal{T}' \hookrightarrow \mathcal{T}^k \text{ and } T' \hookrightarrow \mathcal{T}^k, \text{ due to } T \subseteq \mathcal{T} \text{ and } T' \hookrightarrow T^k \subseteq \mathcal{T}^k.$$

Moreover, we may consider that some unique FO$^{\otimes}$-definable k-tuple $\bar{s} \in \mathcal{T}^k$ lies *outside* all of these subsets.

Then represent $\hat{\mathcal{T}}$ as (a quotient of) the disjoint union

$$\mathcal{T}^{\Sigma} \rightleftharpoons \mathcal{T}' + \sum \{T' : T \in \mathcal{T} \text{ transitive set of cardinality} \geq 2\}.$$

We may consider that $\mathcal{T}^{\Sigma} \subseteq \mathcal{T}^{k+1}$, according to the natural embeddings $\mathcal{T}' \hookrightarrow \mathcal{T}^k \times \{\emptyset\}$ and $T' \hookrightarrow \mathcal{T}^k \times \{T\}$ for each considered transitive set $T \in \mathcal{T}$. Due to FO$^{\otimes}$-coherency of the transformer F we can FO$^{\otimes}$-define a linear preorder $\preceq$ on $\mathcal{T}^{\Sigma}$ such that corresponding quotient of this linearly preordered disjoint sum will be isomorphic to $\langle \hat{\mathcal{T}}; \in\!|_{\hat{\mathcal{T}}}, \leq\!|_{\hat{\mathcal{T}}}\rangle$. Then $\tilde{\mathcal{T}}$ may be represented as the union of this subset $\mathcal{T}^{\Sigma} \subseteq \mathcal{T}^{k+1}$ with the set of all $(k+1)$-tuples of the kind $\langle \bar{s}, T\rangle$, each representing the set T' as an element of $\tilde{\mathcal{T}}$. Linear preorder $\preceq$ on $\mathcal{T}^{\Sigma}$ may be easily further extended by an FO$^{\otimes}$-formula to these new elements of $\mathcal{T}^{k+1}$ so that the corresponding quotient will be isomorphic to $\langle \tilde{\mathcal{T}}; \in\!|_{\tilde{\mathcal{T}}}, \leq\!|_{\tilde{\mathcal{T}}}\rangle$. Finally, we can express in FO$^{\otimes}$ the correspondence between $T \in \mathcal{T}$ and $T' = F_0(T) \in \tilde{\mathcal{T}}$ in terms of $\mathcal{T}$ by the formula

$$\varphi_0(\bar{x}, y; \bar{u}, v) \rightleftharpoons$$
$$(\mathrm{Tran}(v) \;\&\; \text{`}\bar{x} \text{ in } \mathcal{T}^k \text{ represents } v \text{ in } \mathcal{T}\text{'} \;\&\; y = \emptyset \;\&\; \bar{u} = \bar{s})$$

with the lists of variables $\bar{x}, y$ and $\bar{u}, v$ corresponding to T and T', respectively. $\square$

Note 1 The above definition of the class of all $\mathcal{IC}$-$\mathcal{NL}$-operations has a 'semantical' flavour. It is required that the result of any $\mathcal{IC}$-$\mathcal{NL}$-operation computed by corresponding IC-FO$^{\otimes}$-transformer does not depend on the transitive set containing its arguments. A priori, it is unclear how to keep this *second-order* property syntactically by the allowed first-order means. Nevertheless, we shall find corresponding syntactic description of the class $\mathcal{IC}$-$\mathcal{NL}$.

Note 2 In contrast to $\mathcal{IC}$-$\mathcal{NL}$, we can present *immediately* an effective syntax for its subclass $\mathcal{IC}$-$\mathcal{NL}_0$ considered in Proposition 3. Just augment any 'potential' FO$^{\otimes}$-definition of the inflationary and coherent transformers by additional FO$^{\otimes}$-formulas checking correctness of this definition so that we always could get as the output an actual $\mathcal{EAG}$ with the required congruence relation, isomorphic embedding of $\mathcal{EAG}$'s and canonical linear (pre)order, etc. By using this fact and despite the first impression on $\mathcal{IC}$-$\mathcal{NL}$, implied by Note 1, we show below in (12) that $\mathcal{IC}$-$\mathcal{NL}$ *also has an effective syntax*, just by proving $\mathcal{IC}$-$\mathcal{NL} \sim_{\mathrm{HF}} \Delta^{\otimes}_{<} + \mathcal{IC}$-$\mathcal{NL}_0$.

Now we are ready to prove our

Main Result on NLOGSPACE**-computability over** HF

$$\mathcal{IC}\text{-}\mathcal{NL} \quad \sim_{\mathrm{HF}} \quad \Delta^{\otimes}_{<} + \mathcal{IC}\text{-}\mathcal{NL}_0 \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}_{<} \xrightarrow[\text{IC-NL}]{\text{IC-FO}^{\otimes}} \mathcal{EAG}_{<}\right) \tag{12}$$

Proof goes essentially along that of Theorems 2–3 and statements (5,6) with some modifications. First, we must show that any $\mathcal{IC}$-$\mathcal{NL}$-operation $f : \mathrm{HF}^n \to \mathrm{HF}$ is definable in the language $\Delta^{\otimes}_{<} + \mathcal{IC}$-$\mathcal{NL}_0$. By Proposition 3 some $\mathcal{IC}$-$\mathcal{NL}_0$-transformer

$$F : \langle T; \bar{\tau}\rangle \mapsto \langle F_0(T); F_1(T, \bar{\tau})\rangle$$

computes f, i.e. $f(\bar{\tau}) = F_1(T, \bar{\tau})$. Evidently, F_1 is Δ-definable from F by a projection operation. Take $f(\bar{\tau}) = F_1(\mathrm{TC}(\bar{\tau}), \bar{\tau})$.

As in the proof of Theorem 2 it suffices to prove the corresponding

General Statement. *For any list* $\bar{t}(\bar{y}) = t_1(\bar{y}), \ldots, t_n(\bar{y})$ *of* $\Delta^{\otimes}_{<} + \mathcal{IC}$-$\mathcal{NL}$-*terms/formulas there exists corresponding* IC-FO$^{\otimes}$-*transformer*

$$\langle T, <; \bar{\tau}\rangle \mapsto \langle \tilde{T}, <; \bar{t}(\bar{\tau})\rangle.$$

(We need no λ-terms here as in the above versions of the General Statement because we can use, instead, the fact that $\tilde{T}$ does not depend on $\bar{\tau} \in T$ and that $\bar{t}(\bar{\tau}) \in \tilde{T}$ for all $\bar{\tau} \in T$.)

Proof of General Statement. We shall consider only three cases.

In the case if each $t_i(\bar{y})$ is a variable y_{j_i} take suitable elementary transformation of the second kind (with $\tilde{T} = T$).

Let $t_1(\bar{y})$ be $f(a_1(\bar{y}), \ldots, a_n(\bar{y}))$. Then, by induction hypothesis, there exists an IC-FO$^{\otimes}$-transformer

$$\langle T, <; \bar{\tau}\rangle \mapsto \langle T', <; a_1(\bar{\tau}), \ldots, a_n(\bar{\tau}), t_2(\bar{\tau}), t_3(\bar{\tau}), \ldots]\rangle.$$

By $f \in \mathcal{IC}\text{-}\mathcal{NL}$ there exists also an IC-FO$^{\otimes}$-transformer

$$\langle T', <; a_1, \ldots, a_n, t_2, t_3, \ldots\rangle \mapsto \langle \tilde{T}, <; f(a_1, \ldots, a_n), t_2, t_3, \ldots\rangle.$$

The required transformer is obtained by composing these two.

Let $t_1(\bar{y})$ be $\{s(\bar{y}, x) : x \in a(\bar{y}) \& \varphi(\bar{y}, x)\}$. Then, by induction hypothesis, there exist IC-FO$^{\otimes}$-definable transformers

$$\langle T, <; \bar{\tau}\rangle \mapsto \langle T', <; a(\bar{\tau}), t_2(\bar{\tau}), t_3(\bar{\tau}), \ldots\rangle,$$

$$\langle T', <; \bar{\tau}, x\rangle \mapsto \langle T'', <; a(\bar{\tau}), s(\bar{\tau}, x), \varphi(\bar{\tau}, x), t_2(\bar{\tau}), t_3(\bar{\tau}), \ldots\rangle$$

and therefore

$$\langle T, <; \bar{\tau}, x\rangle \mapsto \langle T', <; \bar{\tau}, x\rangle \text{ and } \langle T, <; \bar{\tau}\rangle \mapsto \langle T'', <; \bar{\tau}\rangle$$

Here $T \subseteq T' \subseteq T'' \hookrightarrow T^k$ so that there exist also corresponding FO$^{\otimes}$-formula $\Omega(\bar{x}, \bar{y}, \bar{z})$ over $\langle T, \in, <, =\rangle$ having the natural meaning '$\bar{x}$ in T^k represents some element x in $a(\bar{y})$ such that $\varphi(\bar{y}, x)$ is true and $\bar{z}$ in T^k represents $s(\bar{y}, x)$'. By composing the last transformer with elementary transformer of the first kind

$$[\![\bar{z}; \bar{y}.\exists\bar{x}\Omega(\bar{x}, \bar{y}, \bar{z})]\!]$$

(applied instead of a transitive set in HF to an $\mathcal{EAG}$ with vertices in T^k) and suitable elementary transformation of the second kind we obtain (up to isomorphism) the required transformation

$$\langle T <; \bar{\tau}\rangle \mapsto \langle \tilde{T}, <; \{s(\bar{\tau}, x) : x \in a(\bar{\tau}) \& \varphi(\bar{\tau}, x)\}, t_2(\bar{\tau}), t_3(\bar{\tau}), \ldots\rangle.$$

$\square$

A goal for further research. It would be important to replace in this theorem the countable family $\mathcal{IC}\text{-}\mathcal{NL}_0$ by some finite number of $\mathcal{IC}\text{-}\mathcal{NL}$-operations or schemes like $\{s(x) : x \in a \& \varphi(x)\}$. Then we could say that we really have a good version of Δ-language describing $\mathcal{IC}\text{-}\mathcal{NL}$, i.e. language corresponding to a natural notion of NLOGSPACE-computability over HF. Cf. also Section 11 for some examples of $\mathcal{IC}\text{-}\mathcal{NL}$-operations.

The following simple Proposition confirms that there is some concordance in our approaches to PTIME- and NLOGSPACE-computability over HF.

Proposition 4 $\mathcal{PTIME} = \mathcal{IC}\text{-}\mathcal{NL}$ *iff* PTIME = NLOGSPACE.

Proof If PTIME = NLOGSPACE then, evidently, PTIME = IC-NL as the classes of graph transformers. Therefore, for corresponding set-theoretic versions $\mathcal{PTIME} = \mathcal{IC}\text{-}\mathcal{NL}$ holds.

Conversely, $\mathcal{PTIME} = \mathcal{IC}\text{-}\mathcal{NL}$ implies that arbitrary PTIME-computable predicate $P(x)$ over HF is also computable in NLOGSPACE. In particular, this is true for the predicate $P(x) \rightleftharpoons$ 'HF-set x represents an $\mathcal{AG}$ with two distinguished vertices which denote the same set according to the general collapsing γ'. By PTIME-completeness of the corresponding problem for $\mathcal{AG}$'s [6] this actually gives PTIME = NLOGSPACE.

□

10 On Definability of the Canonical Linear Ordering

Here we present technical results on *partial* definability of the canonical linear ordering which played the crucial role above. This will illuminate the general problem of defining this ordering by the allowed means and also will be useful later in Section 11 to give some examples.

For any transitive class $K \subseteq \mathrm{HF}$ let $\leq_k$ denote the restriction of the canonical linear ordering $\leq_{\mathrm{HF}}$ to the elements of any set $k \in K$.

Proposition 5 *The relation of canonical linear order $k \leq x$ between any $k \in K$ and $x \in \mathrm{HF}$ is uniformly definable in $\Delta^{\otimes}$ modulo any transitive $K \subseteq \mathrm{HF}$ and $\leq_k$. Therefore, if K and $\leq_k$ are $\Delta^{\otimes}$-definable (with $k \in K$ a parameter) then $k \leq x$ is also $\Delta^{\otimes}$-definable.*

Analogously, we may consider, in place of $\leq_{\mathrm{HF}}$, corresponding *canonical preorder* relation $\leq_G$ on arbitrary acyclic finite graph G with its *canonical equivalence relation* $u \equiv v$ ($\Leftrightarrow u \leq v \& v \leq u$) induced from HF by the general collapsing operation. (We usually omit subscript G in the relation symbols $\equiv_G$, $\leq_G$ and $\in_G$.) Moreover, the proof of the above Proposition may be easily obtained from the proof of the following Proposition on acyclic graphs G and a relative $\mathrm{FO}^{\otimes}$-definability of a restricted $\leq_G$ on such graphs. For any subgraph $K \subseteq G$ let $\equiv_{K,G}$ and $\leq_{K,G}$ denote the restricted relations $k \equiv x$ and $k \leq x$, respectively, for $k \in |K|$ and $x \in |G|$. Also let $\leq_x$ be the restriction of the canonical linear preordering $\leq_G$ on any $x \in |G|$, in fact, on the set $\{z \in |G| : z \in_G x\}$.

Proposition 6 *The restricted canonical preordering $\leq_{K,G}$ is $\mathrm{FO}^{\otimes}$-definable in terms of $\equiv_{K,G}$, $\in_G$, K, and $\leq_k$, where K is any transitive subgraph of G.*

Proof Let corresponding *generalized* membership and subset relations $\sqsubseteq$ and $\sqsubseteq$ and also set-minus operation $\setminus\!\!\setminus$ be defined as

$$x \mathrel{\underline{\in}} y \quad \rightleftharpoons \quad \exists x' \in y(x \equiv x'),$$

$$
\begin{aligned}
u \sqsubseteq v &\rightleftharpoons \forall x \in u(x \mathrel{\sqsubseteq\!\!\!\in} v) \quad \text{and}\\
u \mathbin{\backslash\!\!\backslash} v &\rightleftharpoons \{x \in u : x \not\mathrel{\in} v\}.
\end{aligned}
$$

(In the case of a transitive set T or an extensional acyclic graph G these relations $\equiv$, $\mathrel{\sqsubseteq\!\!\!\in}$, $\sqsubseteq$, $\mathbin{\backslash\!\!\backslash}$ coincide, respectively, with the "ordinary" equality, membership, subset relations $=, \in, \subseteq$ and set-minus operation $\setminus$. Then we evidently have $x \sqsubseteq y \Rightarrow x \leq y$, $x \in y \Rightarrow x < y$ and also

$$
\begin{aligned}
x < k &\Leftrightarrow \exists k' \in k \mathbin{\backslash\!\!\backslash} x \forall v(v \mathrel{\sqsubseteq\!\!\!\in} x \& k' < v \Leftrightarrow v \mathrel{\sqsubseteq\!\!\!\in} k \& k' < v)\\
&\Leftrightarrow \exists k' = \max(k \mathbin{\backslash\!\!\backslash} x) \forall x' \in x \mathbin{\backslash\!\!\backslash} k(x' < k'),\\
k \leq x &\Leftrightarrow \forall k' = \max(k \mathbin{\backslash\!\!\backslash} x) \exists x' \in x \mathbin{\backslash\!\!\backslash} k(k' \leq x')\\
&\Leftrightarrow k \sqsubseteq x \vee \forall k' = \max(k \mathbin{\backslash\!\!\backslash} x) \exists x' \in x \mathbin{\backslash\!\!\backslash} k(k' \leq x')\\
&\Leftrightarrow k \sqsubseteq x \vee \exists k' = \max(k \mathbin{\backslash\!\!\backslash} x) \exists x' \in x \mathbin{\backslash\!\!\backslash} k(k' \leq x')\\
&\Leftrightarrow k \sqsubseteq x \vee \exists k' \in k \exists x' \in x(k' = \max(k \mathbin{\backslash\!\!\backslash} x) \& x' \not\mathrel{\sqsubseteq\!\!\!\in} k \& k' \leq x').
\end{aligned}
$$

Let us define relations $\lhd$ and φ on the set $|K| \times |G|$:

$$
\begin{aligned}
(k', x') \lhd (k, x) &\rightleftharpoons k \sqsubseteq x \vee (k' \in k \& x' \in x \& k' = \max(k \mathbin{\backslash\!\!\backslash} x) \& x' \not\mathrel{\sqsubseteq\!\!\!\in} k),\\
\varphi(k, x) &\rightleftharpoons \exists x_0[(\emptyset, x_0) \lhd^{\otimes} (k, x)].
\end{aligned}
$$

These are FO$^{\otimes}$-formulas in terms of $\equiv_{K,G}, \in_G, K$ and $\leq_k$. The last three-place predicate $k_1 \leq_k k_2$ (with $k, k_1, k_2 \in K$ and $k_1, k_2 \in_G k$) is used in the formula $k' = \max(k \mathbin{\backslash\!\!\backslash} x)$. It is assumed that variables k, k' range over K.

Then we can prove by $\in_K$-induction on k that for all x

$$
k \leq x \Leftrightarrow \varphi(k, x)
$$

what gives an FO$^{\otimes}$-definition of $k \leq x$ (and $x < k \Leftrightarrow \neg k \leq x$) we need.

Note, that the case $k \sqsubseteq x$ and, in particular, the base case $k = \emptyset$ are evident. So let $k \not\sqsubseteq x$. Then we have the following equivalences (by using induction hypothesis for $k' \in k$ in the second line):

$$
\begin{aligned}
k \leq x &\Leftrightarrow \exists k' \in k \exists x' \in x(k' = \max(k \mathbin{\backslash\!\!\backslash} x) \& x' \not\mathrel{\sqsubseteq\!\!\!\in} k \& k' \leq x')\\
&\Leftrightarrow \exists k' \in k \exists x' \in x(k' = \max(k \mathbin{\backslash\!\!\backslash} x) \& x' \not\mathrel{\sqsubseteq\!\!\!\in} k \& \varphi(k', x'))\\
&\Leftrightarrow \exists k' \in k \exists x' \in x(k' = \max(k \mathbin{\backslash\!\!\backslash} x) \& x' \not\mathrel{\sqsubseteq\!\!\!\in} k \&\\
&\qquad\qquad \exists x_0[(\emptyset, x_0) \lhd^{\otimes} (k', x')])\\
&\Leftrightarrow \exists x_0 \exists k' \in k \exists x' \in x(k' = \max(k \mathbin{\backslash\!\!\backslash} x) \& x' \not\mathrel{\sqsubseteq\!\!\!\in} k \&\\
&\qquad\qquad (\emptyset, x_0) \lhd^{\otimes} (k', x'))\\
&\Leftrightarrow \exists x_0[(\emptyset, x_0) \lhd^{\otimes} (k, x)] \rightleftharpoons \varphi(k, x).
\end{aligned}
$$

The last equivalence is proved as follows.

($\Rightarrow$): If for some $k' \in k$ and $x' \in x$ we have $k' = \max(k \mathbin{\backslash\!\!\backslash} x)$, $x' \not\mathrel{\sqsubseteq\!\!\!\in} k$ and $(\emptyset, x_0) \lhd^{\otimes} (k', x')$ then $(k', x') \lhd (k, x)$ and therefore $(\emptyset, x_0) \lhd^{\otimes} (k, x)$.

($\Leftarrow$): Let $(\emptyset, x_0) \lhd^{\otimes} (k, x))$ holds. Then two cases are possible.

1. If $(\emptyset, x_0) \lhd (k, x)$ then it follows by the definition of $\lhd$ and by $k \not\sqsubseteq x$ that $k' \rightleftharpoons \emptyset \in k$, $x' \rightleftharpoons x_0 \in x$, $k' = \emptyset = \max(k \setminus\setminus x)$, $x' = x_0 \notin k$ and $(\emptyset, x_0) \lhd^{(\otimes)} (k', x') = (\emptyset, x_0)$ by $k' = \emptyset \sqsubseteq x_0 = x'$.

2. If $(\emptyset, x_0) \lhd^{\otimes} (k', x') \lhd (k, x)$ then $k' \in k \& x' \in x \& k' = \max(k \setminus\setminus x) \& x' \notin k$ by $k \not\sqsubseteq x$, as required.

$\square$

Note 3 A simple example of Δ^*-definable $K \subseteq \mathrm{HF}$ (and $\leq_k$ for $k \in K$) is the class of ordinals or quasi-ordinals. It follows from the Proposition 5 that $k \leq x$ is definable in $\Delta^{\otimes}$ where k ranges over (quasi)ordinals.

Analogously, instead of $\Delta^{\otimes}$ and HF, we could consider definability in $\mathrm{FO}^{\otimes}$ of the corresponding notions in any $\mathcal{EAG}$ or even in any disjoint sum $G = G_1 + G_2$ of $\mathcal{EAG}$'s G_i. According to Proposition 6 we must only show that the relation $\equiv_{K,G}$ is $\mathrm{FO}^{\otimes}$-definable in terms of G only. It suffices to note that $n_1 \equiv n_2 \Leftrightarrow \mathrm{Next}^{\otimes}(0_1, 0_2; n_1, n_2)$ for (quasi)ordinals $n_1 \in G_1$ and $n_2 \in G_2$ in the sorts G_1 and G_2 where (for the case of quasi-ordinals) $\mathrm{Next}(x, y, x', y') \rightleftharpoons \{x\} = x' \;\&\; \{y\} = y'$.

11 Some examples of $\mathcal{IC}$-$\mathcal{NL}$ operations over HF

Here we demonstrate an example of $\mathcal{IC}$-$\mathcal{NL}$-operation over HF which is not expressible in $\Delta^{\otimes}_{\leq}$-language. This is the natural *addition* operation $m + n$ on quasi-ordinals m and n (considered as natural numbers). We can show that this operation on quasi-ordinals is in $\mathcal{IC}$-$\mathcal{NL}$ by the following steps:

- $\mathrm{FO}^{\otimes}$-define in terms of $\langle T; \in\!|_T\rangle$ the minimal extension operation $T \mapsto \tilde{T}$, with $T \subseteq \tilde{T}$ transitive sets, such that for any two quasi-ordinals $m, n \in T$ their sum $m + n$ lies in $\tilde{T}$ and $\mathrm{FO}^{\otimes}$-define the corresponding embedding of T into $\tilde{T}$.

- Show that the predicate $m + n = k$ is also $\mathrm{FO}^{\otimes}$-definable, actually in terms of $\langle \tilde{T}, \in\!|_{\tilde{T}}\rangle$.

- Extend by $\mathrm{FO}^{\otimes}$-means the canonical linear order from T to $\tilde{T}$.

- Show coherency property by demonstrating the $\mathrm{FO}^{\otimes}$-definable extension of the canonical linear preordering from disjoint union $T_1 + T_2$ to $\tilde{T}_1 + \tilde{T}_2$ where T_i and $\tilde{T}_i \supset T_i$ are transitive sets in HF.

Let Q_T be (isomorphic copy of) the set of all quasi-ordinals in T considered simultaneously as a substructure of T. Take $\tilde{T} \rightleftharpoons T + Q_T$. The evident embeddings $T \hookrightarrow \tilde{T} \hookrightarrow T + T \hookrightarrow T^2$ together with identifying the maximal quasi-ordinal in T with 0 in Q_T give the required FO$^{\otimes}$-representation of $\tilde{T}$ and embedding of T into $\tilde{T}$. The resulting $\in$-relation on $\tilde{T}$ is just the union of those on T and on Q_T.

The sum predicate $m + n = k$ for the quasi-ordinals m, n, k is defined as $\mathrm{Next}^{\otimes}(0, m, n, k) \vee (0 = n \& m = k)$ where $\mathrm{Next}(x, y, x', y') \rightleftharpoons \{x\} = x' \;\&\; \{y\} = y'$. Strictly speaking, we must suitably use also the FO$^{\otimes}$-formula

$$\mathrm{Quasiord}(x) \rightleftharpoons \forall y((y \in^* x \vee y = x) \& \exists z(z \in y) \Rightarrow \exists! z(z \in y)).$$

To extend the canonical linear order from T to the resulting $\tilde{T} \supseteq T$ it suffices to define the relation $k \leq x$ (and $x \leq k \rightleftharpoons \neg(k \leq x) \vee k = x$) between any quasi-ordinal k and arbitrary set x, both in $\tilde{T}$, because $\tilde{T} \setminus T$ contains only quasi-ordinals. In fact, this can be done for arbitrary transitive set (T or $\tilde{T}$) by using *only* $\in$-relation (and even for the case k and x are ranging over different sorts). Cf. Proposition 6 and Note 3.

There is also more simple FO$^{\otimes}$-definition than in Proposition 6 for this special case of quasi-ordinals in acyclic graphs:

$$k \leq x \Leftrightarrow k = 0 \vee \exists z \in^* x((\emptyset, z) \lhd^{\otimes} (k, x))$$

where now

$$(k', x') \lhd (k, x) \rightleftharpoons (k' \in k \& x' \in x).$$

To extend the canonical linear preordering from $T_1 + T_2$ to $\tilde{T}_1 + \tilde{T}_2$ it suffices to define the required relation $x \leq_{\tilde{T}_1 + \tilde{T}_2} y$ when x or y is a quasi-ordinal, because $\tilde{T}_i \setminus T_i$ contains only quasi-ordinals. To this end we use here the above defined relation $k \leq x$ in FO$^{\otimes}$ (cf. also Proposition 6 and Note 3) and its symmetric counterpart $x \leq k \rightleftharpoons \neg(k \leq x) \vee k \equiv x$ and also FO$^{\otimes}$-definability of the relation $k \equiv x$ (cf. Note 3).

$\square$

For any $\Delta^{\otimes}_{<}$-definable operation $f : \mathrm{HF}^k \to \mathrm{HF}$ we have the inequality $\mathrm{rank}(f(x_1, \ldots, x_k)) \leq \max(\mathrm{rank}(x_i)) + \mathrm{Const}$, with a constant independent on $\bar{x}$. The operation $+$ does not have this property, and therefore is not $\Delta^{\otimes}_{<}$-definable.

It is easy to extend the above considerations to the case of *multiplication* operation on quasiordinals. Analogously, we have in $\mathcal{IC}$-$\mathcal{NL}$ the operations of addition and multiplication on (finite) ordinals. Only for $\equiv$ and $+$ on the ordinals we must *redefine*

$$\mathrm{Next}(x, y, x', y') \rightleftharpoons x \cup \{x\} = x' \& y \cup \{y\} = y'.$$

Also, the operation $\mathbf{rank}(x) = k$, giving a quasi-ordinal (or ordinal) k of the same rank (i.e. the depth of "nesting") as the set x, is in $\mathcal{IC}$-$\mathcal{NL}$. Note, that FO$^{\otimes}$-definition of the corresponding extension $\tilde{T}$ uses the canonical linear order on T. This rank operation is also not definable in $\Delta^{\otimes}_{<}$ because any such definable operation must be based on a *fixed* number k of applications of restricted powerset operation. Therefore, it adds no more than k of new (quasi)ordinals to the initial transitive set T. However, T may have an arbitrary high rank, but contain a few (even only two) (quasi)ordinals.

12 Deterministic Case

The above Main Result may be easily adopted for the case of DLOGSPACE-computability over HF. First we have to replace the "horizontal" transitive closure construct $\otimes$ by its *deterministic* version $\odot$. Further, we have to consider graphs from the class $\mathcal{EAG}^*_<$ (instead of $\mathcal{EAG}_<$) as codes of HF-sets because the construct $\odot$ seems does not allow to express the "vertical" transitive closure $\in^*_G$ which is used for DLOGSPACE-computability of the Δ-operation TC : HF $\to$ HF.

Note 4 Fortunately, we can keep up (as in the Note 2) the correctness of the input and output codes, i.e. to check the membership relation "$G \in \mathcal{EAG}^*_<$" by FO$^\odot$ and even FO-means. (This is used to show that the class $\mathcal{IC}$-$\mathcal{DL}_0$ of graph transformers considered below has an effective syntax.) Extensionality and all axioms of a strict linear and canonical ordering $<$ are evidently expressible by FO-formulas. Then acyclicity of any finite graph G follows from an additional axiom $x \in y \Rightarrow x < y$ which, of course, must hold on $\mathcal{EAG}_<$'s. This allows to use $\in$-induction in proving that $<$ is uniquely defined by these axioms on any $\mathcal{EAG}_<$. Finally, the relation $\in^*_G$ is completely characterized on all the structures satisfying these axioms by the formula $x \in^* y \Leftrightarrow x \in y \vee \exists z \in y(x \in^* z)$ also by using $\in$-induction (on y).

As above, we can define (identical) "deterministic" classes of transformers IC-FO$^\odot$ = IC-DL of $\mathcal{EAG}^*_<$'s and their set-theoretic version $\mathcal{IC}$-$\mathcal{DL}_0$ and also the class $\mathcal{IC}$-$\mathcal{DL} \supseteq \mathcal{IC}$-$\mathcal{DL}_0$ of operations over HF computable by these transformers. Also, we can prove

Main Result on DLOGSPACE-**computability over** HF

$$\mathcal{IC}\text{-}\mathcal{DL} \quad \sim_{\mathrm{HF}} \quad \Delta^\odot_< + \mathcal{IC}\text{-}\mathcal{DL}_0 \quad \sim_{\mathrm{HF}} \quad \left(\mathcal{EAG}^*_< \xrightarrow[\text{IC-DL}]{\text{IC-FO}^\odot} \mathcal{EAG}^*_<\right). \tag{13}$$

Note, that the operation + for quasi-ordinals which was shown to be in $\mathcal{IC}$-$\mathcal{NL}$ actually lies in $\mathcal{IC}$-$\mathcal{DL}$. Essentially, we must change the formal FO$^\otimes$-definition of the canonical preorder relations $k \le x$ (for k quasi-ordinal) in any $\mathcal{EAG}$ (or disjoint sum of $\mathcal{EAG}$'s) from Section 11. Just replace $\otimes$ by $\odot$ and $x' \in x$ in $\lhd$ by "x' is the maximal wrt $\le$ element of the set x". This makes new version of $\lhd$ *single-valued* relation ($D(\lhd) = \lhd$) and, on the other hand, *reduces* the definition of $k \le x$ to three place relation $x_1 \le_x x_2$ where $x_1, x_2 \in x$ and $\le_x$ is the restriction of $\le$ to the set x. It proves that such a reducing suffices.

Finally note, that this Main Result holds also if to replace in its formulation '$\mathrm{FO}^{\odot}$' by 'FO^{*}', or even by 'FO', and/or omit $<$. In particular, $\mathcal{IC}$-$\mathcal{DL}_{(0)}$ is replaced by $\mathcal{IC}$-$\mathcal{FO}^{*}_{(0)}$, or even by $\mathcal{IC}$-$\mathcal{FO}_{(0)}$. Cf. also footnote to Theorem 5. However, it seems that $+$ is not in $\mathcal{IC}$-$\mathcal{FO}^{*}$.

13 Relations to Finite Model Theory

Computability and definability over HF-sets were discussed in this paper in terms of finite graph transformers. We have found several interesting corresponding questions in finite model theory which are listed below for the convenience.

1. *Is it possible to uniformly define in* $\mathrm{FO}^{\otimes}$ *canonical linear ordering on* $\mathcal{EAG}_{(\prec)}$*'s?*
2. *Is it possible to uniformly define in* $\mathrm{FO}^{\otimes}$ *any linear ordering on* $\mathcal{EAG}$*'s?* If 'YES' then the following question also has affirmative answer.
3. *Are the classes of* $\mathrm{FO}^{\otimes}$*-definable and* NLOGSPACE*-computable queries over* $\mathcal{EAG}$*'s the same?*
4. *Is* $\equiv_G$ *or, more strongly,* $\leq_G$ *uniformly definable in* $\mathrm{FO}^{\otimes}$ *on* AG*'s?* Note, that for the case of $\mathrm{AG}_{\prec}$'s such question would be equivalent to "PTIME= NLOGSPACE?".
5. *Does any* (N)LOGSPACE*-transformer:* $\mathcal{EAG}_{<} \to \mathcal{EAG}_{<}$ *belong to the class* IC-(N)LOGSPACE*?* Evidently, this question may be reformulated in terms of $\mathrm{FO}^{\otimes}$-definability.
6. *Is* $\leq_{G_1+G_2}$ $\mathrm{FO}^{\otimes}$*-definable in* $\langle G_1, G_2, <_{G_1}, <_{G_2}, \equiv_{G_1+G_2}\rangle$*?* (Cf. the definition of *coherence* in Section 9.)
7. *Is it possible to uniformly define in* $\mathrm{FO}^{\odot}$ *the relation* $\in^{*}_{G}$ *on* $\mathcal{EAG}$*'s?*

14 Acknowledgments

The authors are grateful to the anonymous referee for very useful comments on the first version of the paper.

References

[1] Abiteboul, S., Beeri, C.: On the power of languages for the manipulation of complex objects. INRIA research report **846** (1988). Abstract in Proc. International Workshop on Theory and Applications of Nested Relations and Complex Objects. Darmstadt (1987)

[2] Aczel, P.: Non-Well-Founded Sets. CSLI Lecture Notes. No 14 (1988)

[3] Barwise J.K.: Admissible Sets and Structures. Springer, Berlin, 1975

[4] Blakely, J.A., Larson, P.A., Tompa, F.W.: Efficiently updating materialized view. Proc. ACM-SIGMOD. 1986. Intern. Conf. on Management of Data. Washington D.C. May 1986. 61–71

[5] Buss, S.R.: Bounded Arithmetic. Bibliopolis. Napoli. 1986

[6] Dahlhaus, E.: Is SETL a suitable language for parallel programming—a theoretical approach. Börger, E., Kleine Buning, H., Richter, M.M. ed. CSL'87. LNCS **329** (1987) 56–63

[7] Dahlhaus, E., Makowsky, J.: The Choice of programming Primitives in SETL-like Languages. ESOP'86. LNCS **213** (1986) 160–172

[8] Dahlhaus, E., Makowsky, J.: Query languages for hierarhic databases. Information and Computation **101** (1992) 1–32

[9] Ershov, Yu.L.: Numberings theory. "Nauka". Glavnaja redakcija phisicomatematicheskoi literatury. Moskva. 1977 (in Russian)

[10] Feferman, S.: Predicatively reducible systems of set theory. Proc. Symp. in Pure Math. Vol. **13**, Part II (1974) 11–32

[11] Gandy, R.O.: Set theoretic functions for elementary syntax. Proc. Symp. in Pure Math. Vol. **13**, Part II (1974) 103–126

[12] Garey, M.R. and Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H.Freeman and Company, New York, 1979.

[13] Grumbach, S., Vianu, V.: Tractable query languages for complex object databases. Rapports de Recherche N**1573**. INRIA. 1991

[14] Gurevich, Y.: Algebras of feasible functions. FOCS **24** (1983) 210–214

[15] Gurevich, Y.: Evolving Algebras 1993: Lipari Guide. Specification and Validation Methods (E. Börger ed.) Oxford University Press (1994)

[16] Immerman, N.: 'Relational queries computable in polynomial time', in: *Proc. 14th. ACM Symp. on Theory of Computing*, (1982) 147–152; cf. also *Inform. and Control* **68** (1986) 86–104.

[17] Immerman, N.: Languages which captures complexity classes. SIAM J. Comput., **16** 4 (1987) 760–778

[18] Immerman, N.: Descriptive and computational complexity. Proceedings of Symposia in Applied Mathematics, **38** (1989) 75–91

[19] Immerman, N., Patnaik, S. and Stemple, D.: The expressiveness of a family of finite set languages. Theoretical Computer Science, **155**, N 1 (1996) 111–140

[20] Jensen, R.B.: The fine structure of the constructible hierarchy. Ann. Math. Logic **4** (1972) 229–308

[21] Jensen, R.B., Karp, C.: Primitive recursive set functions. Proc. Sympos. Pure Math., vol. **13** part I (1971) 143–176

[22] Kuper, G.M., Vardi, M.Y.: A new approach to database logic. Proc. 3rd ACM Symp. on Principles of Database Systems 1984

[23] Levy, A.: A hierarchy of formulas in set theory. Mem. Amer. Math. Soc. No. 57 (1965) 76pp. MR 32 N 7399

[24] Livchak, A.B.: Languages of polynomial queries. *Raschet i optimizacija teplotehnicheskih ob'ektov s pomosh'ju EVM* , Sverdlovsk, 1982, p. 41 (in Russian).

[25] Parikh, R.: Existence and feasibility in arithmetic. JSL, **36** No 3 (1971) 494–508

[26] Red'ko, V.N., Basarab, I.A.: Data bases and information systems. News in the life, science and technique; series: "Mathematics, Kybernetics" N6 Moscow. Znanije. (1987) (in Russian)

[27] Sazonov, V.Yu.: Polynomial computability and recursivity in finite domains. Elektronische Informationsverarbeitung und Kybernetik. **16**, N7 (1980) 319–323

[28] Sazonov, V.Yu.: A logical approach to the problem "P=NP?". Proc., Math. Found. of Comput. Sci. Lect. Not. Comput. Sci. **88** Springer. New York (1980) 562–575 (An important correction to this paper is given in [29], P.490)

[29] Sazonov, V.Yu.: On existence of complete predicate calculus in metamathematics without exponentiation. Lect. Not. Comput. Sci. **118** Springer. New York (1981) 483–490

[30] Sazonov, V.Yu.: Bounded set theory and polynomial computability. All Union Conf. Appl. Logic., Proc. Novosibirsk (1985) 188–191 (In Russian)

[31] Sazonov, V.Yu.: The Collection Principle and the Existential Quantifier. Logiko-matematicheskie problemy MOZ., *Vychislitel'nye sistemy* **107** (1985) 3O–39 (In Russian). (English translation in: *Amer. Math. Soc. Transl.* (2) bf Vol.142 (1989) 1–8)

[32] Sazonov, V.Yu.: Bounded set theory, polynomial computability and Δ-programming. Application aspects of mathematical logic. Computing systems **122** (1987) 110–132 (In Russian) Cf. also a short English version of this paper in: Lect. Not. Comput. Sci. **278** Springer (1987) 391–397

[33] Sazonov, V.Yu.: Bounded set theory and inductive definability. Abstracts of Logic Colloquium'90. JSL **56** Nu.3 (1991) 1141–1142

[34] Sazonov, V.Yu.: Hereditarily-finite sets, data bases and polynomial-time computability. TCS **119** Elsevier (1993) 187–214

[35] Sazonov, V.Yu.: On polynomial regular codings of hereditarily finite sets. *Bulletin of Symbolic Logic*. Abstracts of European Summer Meeting of the Association for Symbolic Logic, Keel, 1993.

[36] Sazonov, V.Yu.: A bounded set theory with anti-foundation axiom and inductive definability. Computer Science Logic, 8th Workshop, CSL'94 Kazimierz, Poland, September 1994, Selected Papers. Lecture Notes in Computer Science **933** Springer (1995) 527–541

[37] Sazonov, V.Yu.: On bounded set theory. Invited talk on the Congress on Logic, Methodology and Philosophy of Sciences, Florence 1995; to appear.

[38] Sazonov, V.Yu., Lisitsa, A.P.: Δ-languages for sets and sub-PTIME graph transformers. Database Theory – ICDT'95, 5th International Conference, Prague, Czech Republic, January 1995, Proceedings. Lecture Notes in Computer Science **893** Springer (1995) 125–138

[39] Sazonov, V.Yu., Leontjev, A.V.: On coding of hereditarily-finite sets, polynomial-time computability and Δ-expressibility. Proc. Conf. Appl. Logic. Novosibirsk, May 1993. Computing systems **146** (1992) 195-198. (In English).

[40] Shoenfield, J. R.: *Mathematical Logic*. Addison-Wesley, Reading, MA, 1967

[41] Szelepcsényi, R.: 1987, The method of forcing for nondeterministic automata. *Bull. European Association Theor. Comp. Sci.* (Oct. 1987) 96–100

[42] Vardi, M.Y.: The complexity of relational query languages. Proc. of the 14th. ACM Symp. on Theory of Computing, (1982) 137–146