Constructibility and Decidability versus Domain Independence and Absoluteness

Arnon Avron

School of Computer Science, Tel Aviv University, Israel

Abstract

We develop a unified framework for dealing with constructibility and absoluteness in set theory, decidability of relations in effective structures (like the natural numbers), and domain independence of queries in database theory. Our framework and results suggest that domain-independence and absoluteness might be the key notions in a general theory of constructibility, predicativity, and computability.

1 Introduction: Absoluteness and Constructibility

As is well-known, Church Thesis (CT) identifies the intuitive, imprecise notions of computability and decidability with the precise mathematical notion of recursiveness. Accordingly, CT might be useful for two different goals. First, the only known way to provide a precise mathematical proof that a certain relation is *not* decidable, is to show that it is not recursive. Second, to become convinced that a certain function (or relation) *is* recursive, it suffices by CT to give an intuitive argument why it should be computable (or decidable), allowing one to leave out most of the tedious details involved in a direct proof of recursiveness (in principle such an informal argument can always be translated into a full, rigorous one, but people seldom bother to do so).

Now the notion of computation to which CT applies is connected with countable discrete structures (like the natural numbers, or strings of symbols from some alphabet). However, we believe that it is an instance of a more general notion: the notion of *construction*, which is central in constructive mathematics, but is also heavily used in all areas of classical mathematics (from

Email address: aa@cs.tau.ac.il (Arnon Avron).
URL: http://www.cs.tau.ac.il/ aa/ (Arnon Avron).

Euclidean Geometry, where "construction problems" play a decisive role, to set theory).

The ultimate goal of the research to which this paper and its planned continuations are devoted, is to develop a unified general logical framework for studying the notions of construction and constructibility, with an eye to find a corresponding generalization of CT. The present paper describes what we believe to be promising steps in this direction. Its main focus is on constructions with *sets*. This is justified by the central role that sets and set theories have in modern mathematics. However, we do not want to commit ourselves here to the platonic concept of "an arbitrary set". Instead we take here the logical view of a set as the extension of a property which is *defined* by some "acceptable" (or "safe") formula in some, intuitively meaningful, formal language. [1]. Accordingly, the main question is: what formulas can be taken as defining a *construction* of a set from given objects (including other sets)? To get a possible reasonable answer, we combine ideas from three sources.

Set Theory Gödel classical work [11] on the constructible universe L is best known for its applications in pure set theory, especially consistency and independence proofs. However, it is of course of great interest also for the study of the general notion of constructibility. Thus for characterizing the "constructible sets" Gödel identified a set of basic operations on sets (which we may call "computable operations"), that may be used for "effectively" constructing new sets from given ones (in the process of creating the universe of "constructible" sets). For example, binary union and intersection are "effective" in this sense, while the powerset operation is not. Gödel has even provided a finite list of basic set operations, from which all other "effective" (for his purposes) constructions can be obtained through compositions. Another very important idea which was introduced in [11] is the notion of absoluteness of formulas. Roughly, a formula in the language of set theory is absolute if its truth value in a transitive class M, for some assignment v of objects from M to its free variables, depends only on v, but not on M (i.e. the truth value is the same in every transitive class M, in which v is legal). Absoluteness turned out to be a key property of formulas which are used for defining "constructible sets".

Formal arithmetic Absoluteness is not a decidable property. The following set Δ_0 of absolute formulas is therefore extensively used in set theory as a syntactically defined approximation:

- Every atomic formula is in Δ_0.
- If φ and ψ are in Δ_0, then so are $\neg\varphi$, $\varphi \vee \psi$, and $\varphi \wedge \psi$.

---

[1] I am the first to admit that this is somewhat vague. But the goal of this type of research is exactly to try to develop precise counterparts for the vague notions and intuitions with which one starts.

- If x and y are two different variables, and φ is in Δ_0, then so are $\exists x \in y\varphi$ and $\forall x \in y\varphi$.

Now a set of Δ_0 formulas (also called in [16] "bounded formulas" or "Σ_0-formulas") which has *exactly the same definition* (but of course in a different signature) is used in formal arithmetic in order to characterize the computable and the semi-computable (r.e.) relations on the natural numbers. This obvious analogy between the roles in set theory of absolute formulas and of set-theoretical Δ_0 formulas, and the roles in formal arithmetic and computability theory of decidable formulas and of arithmetical Δ_0 formulas, has indeed been noticed and exploited in the research on set theory.

Relational database theory: The importance of computations with sets to this area is obvious: to provide an answer to a query in a relational database, a computation should be made in which the input is a finite set of finite sets of tuples (the "tables" of the database), and the output should also be a finite set of tuples. In other words: the computation is done with (finite) sets. Accordingly, for *effective* computations with finite relations some finite set of basic operations has been identified in database theory, and this basic set defines (via composition) what is called there "the relational algebra" ([1,18]). Interestingly, there is a lot of similarity between the list of operations used in the relational algebra and Gödel's list of basic operations for constructing sets.

It may be less obvious that also the idea of absoluteness is very important for database theory. However, we shall see that domain independence ([13,18,1]), which is the key property that "acceptable" queries should have, is strongly related to the property of absoluteness.

In what follows we reveal strong connections between the notions of constructibility, decidability, domain independence and absoluteness, and develop a unified framework for dealing with them. Our framework and results suggest that a certain general notion of domain independence (of which absoluteness is a special case) is the really fundamental notion (while the others are special cases of a sort, in some particular types of structures).

2 Domain Independence and Computability in Databases

2.1 The Concept of Domain Independence

From a logical point of view, a relational database DB of a scheme $\{P_1, \ldots, P_k\}$ is just a tuple $\langle \underline{P_1}, \ldots, \underline{P_k} \rangle$ of *finite* interpretations (called "tables") of the predicate symbols $P_1, \ldots, P_k$. DB can be turned into a structure S for a first-order language L with equality, the signature of which includes $\{P_1, \ldots, P_k\}$

and perhaps also constants, by specifying a domain D, and an interpretation of the constants of L in it [2]. The domain D should be at most countable (and usually it is finite), and should of course include the union of the domains of the tables in DB. A query for DB is simply a formula ψ of L. If ψ has n free variables, then the answer to ψ in S is the set of n-tuples which satisfy it in S. If ψ is closed, then the answer to the query is either "yes" or "no", depending on whether ψ holds in S or not (The "yes" and "no" can be interpreted as $\{\langle\rangle\}$ and $\emptyset$, respectively. Here $\langle\rangle$ is the unique 0-tuple, and like in set theory, it might be identified with $\emptyset$). Now not every formula ψ of a L can serve as a query. Acceptable is only a query the answer to which is a function of $\langle \underline{P_1}, \ldots, \underline{P_k} \rangle$ alone (and does not depend on the exact identity of the domain D, which might be unknown). Such queries are called *domain independent* ([13,18,1]). The exact definition is reproduced below.

Definition 1 Let σ be a first-order signature, and let S_1 and S_2 be two structures for σ. S_1 is a *weak substructure* of S_2 (notation: $S_1 \subseteq_\sigma S_2$) if the domain of S_1 is a subset of the domain of S_2, and the interpretations in S_1 and S_2 of the constants of σ are identical.

Definition 2 Let σ be a signature which includes $\overrightarrow{P} = \{P_1, \ldots, P_k\}$.

(1) Let S_1 and S_2 be two structures for σ. S_1 is a $\overrightarrow{P}-$*substructure* of S_2 (and S_2 is a $\overrightarrow{P}-$extension of S_1) if $S_1 \subseteq_\sigma S_2$, and the interpretations in S_1 and S_2 of $P_1, \ldots, P_k$ are identical (i.e.

$$S_2 \models P_i(a_1, \ldots, a_n) \;\Leftrightarrow a_1 \in S_1 \wedge \ldots \wedge a_n \in S_1 \wedge S_1 \models P_i(a_1, \ldots, a_n)$$

for every P_i in $\overrightarrow{P}$ and for all $a_1 \in S_2, \ldots, a_n \in S_2$).

(2) A formula $\varphi(x_1 \ldots, x_n)$ in σ is $\overrightarrow{P}-d.i.$ *($\overrightarrow{P}-$domain-independent)* [3], if whenever S_1 is a $\overrightarrow{P}-$substructure of S_2 then for all $a_1 \in S_2, \ldots, a_n \in S_2$:

$$S_2 \models \varphi(a_1, \ldots, a_n) \quad \leftrightarrow \quad a_1 \in S_1 \wedge \ldots \wedge a_n \in S_1 \wedge S_1 \models \varphi(a_1, \ldots, a_n)$$

Note 1 The last condition can be reformulated as follows:

$$\{\overrightarrow{a} \in S_2^n \mid S_2 \models \varphi(\overrightarrow{a})\} = \{\overrightarrow{a} \in S_1^n \mid S_1 \models \varphi(\overrightarrow{a}\})$$

This implies that if there are no predicate symbols in φ besides $\{P_1, \ldots, P_k\}$ (and equality), and the interpretations of the constants are fixed, then the val-

[2] Usually it is demanded in databases to have different interpretations for different constants. This is known as the unique name assumption. This assumption is not important for us here.

[3] This is a slight generalization of the definition in [17], which in turn is a generalization of the usual one ([13,18]). The latter applies only to free Herbrand structures which are generated by adding to σ some new set of constants.

ues of the function $F^S_\varphi = \lambda \underline{P_1}, \ldots, \underline{P_k}.\{\langle a_1, \ldots, a_n \rangle \in S^n \mid S \models \varphi(a_1, \ldots, a_n)\}$ indeed do not depend on the choice of S.

Note 2 In Definition 2 we did not assume that the interpretations of the predicates in $\overrightarrow{P}$ should be finite. This assumption is needed only when we want to connect d.i. to the computability of the function F^S_φ defined in the previous Note: If L contains no function symbols, S is a structure for $L - \{P_1, \ldots, P_k\}$, and the interpretations of the predicate symbols of this language in S are all decidable, then the value of $F^S_\varphi(\underline{P_1}, \ldots, \underline{P_k})$ for *finite* $\underline{P_1}, \ldots, \underline{P_k}$ can be computed by switching to the finite substructure S' of S induced by the union of the domains of $\underline{P_1}, \ldots, \underline{P_k}$, together with the interpretations in S of the constants mentioned in φ. The $\overrightarrow{P}$-d.i. of φ ensures that $F^S_\varphi(\underline{P_1}, \ldots, \underline{P_k}) = F^{S'}_\varphi(\underline{P_1}, \ldots, \underline{P_k})$, and the latter is of course finite and computable.

Practical database query languages are designed so that only d.i. queries can be formulated in them. Unfortunately, it is undecidable which formulas are d.i. and which are not ([6]). Therefore all commercial query languages (like SQL) allow to use as queries only formulas from some syntactically defined class of d.i. formulas. Many explicit proposals of decidable, syntactically defined classes of safe formulas have been made in the literature. The simplest among them (and the closer to what has actually been implemented) is perhaps the following class $SS(\overrightarrow{P})$ ("syntactically safe" formulas for a database scheme $\overrightarrow{P}$) from [18]: [4]

(1) $P_i(t_1, \ldots, t_{n_i}) \in SS(\overrightarrow{P})$ in case P_i (of arity n_i) is in $\overrightarrow{P}$ (recall that each t_i is here either a variable or a constant).
(2) $x = c$, $c = x$ and $x \neq x$ are in $SS(\overrightarrow{P})$ (where x is a variable and c is a constant). [5]
(3) $\varphi \vee \psi \in SS(\overrightarrow{P})$ if $\varphi \in SS(\overrightarrow{P})$, $\psi \in SS(\overrightarrow{P})$, and $Fv(\varphi) = Fv(\psi)$ (where $Fv(\varphi)$ denotes the set of free variables of φ).
(4) $\exists x \varphi \in SS(\overrightarrow{P})$ if $\varphi \in SS(\overrightarrow{P})$.
(5) If $\varphi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_k$, then $\varphi \in SS(\overrightarrow{P})$ if the following conditions are met:
    (a) For each $1 \leq i \leq k$, either φ_i is atomic, or φ_i is in $\mathcal{SS}(\overrightarrow{P})$, or φ_i is a negation of a formula of either type.
    (b) Every free variable x of φ is limited in φ. This means that there exists $1 \leq i \leq k$ such that x is free in φ_i, and either $\varphi_i \in SS(\overrightarrow{P})$, or there exists y which is already limited in φ, and $\varphi_i \in \{x = y, y = x\}$.

[4] What we present below is both a generalization and a simplification of Ullman's original definition.
[5] $x \neq x$ was not mentioned by Ullman, but it is obviously d.i.

The set $SS(\overrightarrow{P})$ does not seem to resemble much the set Δ_0. Thus the latter is closed under negation, while $SS(\overrightarrow{P})$ is not. Nevertheless, in the next subsection a strong connection will be revealed, when we introduce in the context of databases a common generalization of d.i. and absoluteness. It should also be noted that there is one clause in the definition of $SS(\overrightarrow{P})$ which is somewhat strange and complicated: the last one, which treats conjunction. In the unified framework described in the next subsection this problematic clause is replaced by a simpler one (which at the same time is more general).

2.2 Partial Domain Independence and Absoluteness in Databases

To see the connection between absoluteness and d.i., we start by recalling the most basic notion of absoluteness as given in [14] (Definition 3.1 (1)). For simplicity, we assume from now on that σ is a first-order signature with equality and *no function symbols*.

Definition 3 Let $S_1 \subseteq_\sigma S_2$. $\varphi(x_1, \ldots, x_n)$ is *absolute* for S_1 and S_2 if

$$\forall a_1 \in S_1, \ldots, a_n \in S_1.\ S_2 \models \varphi(a_1, \ldots, a_n) \quad \leftrightarrow \quad S_1 \models \varphi(a_1, \ldots, a_n)$$

Absoluteness of formulas in the context of databases can most naturally be defined now as follows:

Definition 4 Let σ be a signature like in Definition 2. A formula φ in σ is called $\overrightarrow{P}-$*absolute* if φ is absolute for S_1 and S_2 whenever S_1 and S_2 are structures for σ such that S_1 is a $\overrightarrow{P}$-substructure of S_2.

There is an obvious similarity between the concepts of d.i. and absoluteness as defined above. However, the two notions are not identical. Thus, the formula $x = x$ is not d.i., although it is clearly absolute. In order to provide a common generalization, the formula *property* of d.i. was turned in [3] into the following *relation* between a formula φ and finite subsets of $Fv(\varphi)$ (recall that $Fv(\varphi)$ denotes the set of free variables of φ):

Definition 5

(1) Let $S_1 \subseteq_\sigma S_2$. A formula $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ in σ is d.i. for S_1 and S_2 with respect to $\{x_1, \ldots, x_n\}$ (notation: $\varphi \succ^{S_1;S_2} \{x_1, \ldots, x_n\}$), if for all $a_1, \ldots, a_n \in S_2$ and $b_1 \ldots, b_m \in S_1$:

$$S_2 \models \varphi(\overrightarrow{a}, \overrightarrow{b}) \quad \leftrightarrow \quad a_1 \in S_1 \wedge \ldots \wedge a_n \in S_1 \ \wedge S_1 \models \varphi(\overrightarrow{a}, \overrightarrow{b})$$

(2) Let σ be like in Definition 2. A formula $\varphi(x_1,\ldots,x_n,y_1,\ldots,y_m)$ in σ is $\overrightarrow{P}-$d.i. with respect to $\{x_1,\ldots,x_n\}$ if $\varphi \succ^{S_1;S_2} \{x_1,\ldots,x_n\}$ whenever S_1 is a $\overrightarrow{P}$-substructure of S_2.

Note that φ is $\overrightarrow{P}-$d.i. iff it is $\overrightarrow{P}-$d.i. with respect to $Fv(\varphi)$. On the other hand φ is $\overrightarrow{P}-$absolute iff it is $\overrightarrow{P}-$d.i. with respect to $\emptyset$. Note also that the formula $x = y$ is only partially d.i.: it is d.i. with respect to $\{x\}$ and with respect to $\{y\}$, but not with respect to $\{x, y\}$.

Again the condition on φ in Definition 5(1) can be reformulated as follows:

$$\forall b_1 \ldots, b_m \in S_1.\{\overrightarrow{a} \in S_2^n \mid S_2 \models \varphi(\overrightarrow{a}, \overrightarrow{b})\} = \{\overrightarrow{a} \in S_1^n \mid S_1 \models \varphi(\overrightarrow{a}, \overrightarrow{b}\})$$

This now implies that if $\varphi(x_1,\ldots,x_n,y_1,\ldots,y_m)$ is $\overrightarrow{P}-$d.i. with respect to $\{x_1,\ldots,x_n\}$, all the predicate symbols in φ are included in $\overrightarrow{P} \cup \{=\}$, and the interpretations of the constants are fixed, then the values of the function [6]

$$F_\varphi^S = \lambda \underline{P_1},\ldots,\underline{P_k}.\lambda y_1,\ldots,y_m \in S.\{\langle a_1,\ldots,a_n\rangle \in S^n \mid S \models \varphi(a_1,\ldots,a_n,\overrightarrow{y})\}$$

do not depend on the exact choice of the structure S to which $y_1,\ldots,y_m$ all belong, but only on the interpretations of $\{P_1,\ldots,P_k\}$ in it, and on the identity of $y_1,\ldots,y_m$. Note that for given $\underline{P_1},\ldots,\underline{P_k}$ and S, $F_\varphi^S(\underline{P_1},\ldots,\underline{P_k})$ is a function from S^m to the set of subsets of S^n. By an argument similar to that given in Note 2, if $\underline{P_1},\ldots,\underline{P_k}$ are all finite then the values of this function are finite sets, and the function itself is computable [7]. Note that in case $n = 0$ the possible values of this function are $\{\langle\rangle\}$ and $\emptyset$, which again can be taken as "true" and "false", respectively. Hence in this particular case what we get is a computable m-ary predicate on S. From this point of view m-ary predicates on a set S should be viewed as a special type of functions from S^m to the set of finite sets of S-tuples, rather than as a special type of functions from S^m to S, with arbitrary chosen two elements from S serving as the two classical truth values (while like in set theory, functions from S^m to S should be viewed as a special type of $(m+1)$-ary predicates on S. Alternatively, one may identify functions from S^m to S with functions from S^m to the set of singletons of elements of S).

Given $\overrightarrow{P}$, let $\varphi \succ X$ abbreviate that φ is $\overrightarrow{P}-$d.i. with respect to X. It is not difficult to see (see Theorem 2 below) that $\succ$ has the following properties:

[6] For brevity, we use again the notation F_φ^S, even though the function here might depend also on the choice of of the subset $\{x_1,\ldots,x_n\}$ of $Fv(\varphi)$, with respect to which φ is $\overrightarrow{P}-$d.i. (there may be more than one possible choice).

[7] In fact, it remains computable even if there are other predicate symbols in φ besides $\{P_1,\ldots,P_m\}$, provided that their interpretations in S are all decidable.

1. $\varphi \succ X$ if φ is $t_1 = t_2$ or $p(t_1, \ldots, t_n)$ (where $p \in \overrightarrow{P}$), and $X \subseteq Fv(\varphi)$.
2. $x \neq x \succ \{x\}$, $t = x \succ \{x\}$, and $x = t \succ \{x\}$ if $x \notin Fv(t)$.
3. $\neg\varphi \succ \emptyset$ if $\varphi \succ \emptyset$.
4. $\varphi \vee \psi \succ X$ if $\varphi \succ X$ and $\psi \succ X$.
5. $\varphi \wedge \psi \succ X \cup Y$ if $\varphi \succ X$, $\psi \succ Y$, and $Y \cap Fv(\varphi) = \emptyset$.
6. $\exists y\varphi \succ X - \{y\}$ if $y \in X$ and $\varphi \succ X$.

These properties can be used to define a syntactic approximation $\succ^s_{\overrightarrow{P}}$ of the $\overrightarrow{P}$-d.i. relation. It can easily be checked that the set $\{\varphi \mid \varphi \succ^s_{\overrightarrow{P}} Fv(\varphi)\}$ strictly extends $SS(\overrightarrow{P})$ (but note how the complicated last clause in the definition of $SS(\overrightarrow{P})$ is replaced here by a concise clause concerning conjunction!).

3 A General Framework for D. I. and Absoluteness

Although the notion of $\overrightarrow{P}$-absoluteness is closely related to the set-theoretical notion of absoluteness, it is not really a generalization of that notion as it is usually used in set theory. In addition to $=$, the language of set theory has only one binary predicate symbol: $\in$. Now the notion of $\{\in\}$-absoluteness is useless, since if S_2 is a model of $\forall x \exists y. x \in y$ then S_1 can be an $\{\in\}$-substructure of S_2 iff S_1 is identical with S_2. The notion of $\emptyset$-absoluteness, in contrast, *is* identical to the most general notion of absoluteness as defined e.g. in [14], but that notion is of little use in set theory. Thus Δ_0-formulas are not $\emptyset$-absolute. Indeed, in order for Δ_0-formulas to be absolute for structures S_1 and S_2 (where S_1 is a substructure of S_2), we should assume that S_1 is a *transitive* substructure of S_2. This means that if b is an element of S_1, and $S_2 \models a \in b$, then a belongs to S_1, and $S_1 \models a \in b$. In other words: the formula $x \in y$ should be d.i. for S_1 and S_2 with respect to $\{x\}$ (but not with respect to $\{y\}$). This observation leads to the following general framework for domain independence and absoluteness (originally introduced in [3]):

Definition 6 A *d.i.-signature* is a pair (σ, F), where σ is an ordinary first-order signature with equality and no function symbols, and F is a function which assigns to every n-ary predicate symbol from σ (other than equality) a subset of $\mathcal{P}(\{1, \ldots, n\})$. [8]

Definition 7 Let (σ, F) be a d.i.-signature, and let $S_1 \subseteq_\sigma S_2$. S_2 is called a $(\sigma, F)-$*extension* of S_1 (and S_1 is called a $(\sigma, F)-$substructure of S_2) if $p(x_1, \ldots, x_n) \succ^{S_1;S_2} \{x_{i_1}, \ldots, x_{i_k}\}$ whenever p is an n-ary predicate of σ, $x_1, \ldots, x_n$ are n distinct variables, and $\{i_1, \ldots, i_k\} \in F(p)$.

[8] In [3] a more general notion of a d.i.-signature was introduced, in which function symbols are allowed, and a corresponding condition for them is given.

Definition 8 Let (σ, F) be a d.i.-signature.

(1) A formula φ of σ is called $(\sigma, F)-$*d.i. w.r.t.* X (notation: $\varphi \succ_{(\sigma,F)} X$) if $\varphi \succ^{S_1;S_2} X$ whenever S_2 is a $(\sigma, F)-$extension of S_1.
(2) A formula φ of σ is called $(\sigma, F)-$*d.i.* if $\varphi \succ_{(\sigma,F)} Fv(\varphi)$.
(3) A formula φ of σ is called $(\sigma, F)-$*absolute* if $\varphi \succ_{(\sigma,F)} \emptyset$.

Note 3 We assume that we are talking only about first-order languages with equality, and so we do not include the equality symbol in our first-order signatures. Had it been included then we would have demanded $F(=)$ to be $\{\{1\}, \{2\}\}$ (or $\{\{1\}, \{2\}, \emptyset\}$, which is equivalent). The reason is that $x_1 = x_2$ is always d.i. w.r.t. both $\{x_1\}$ and $\{x_2\}$, but usually not w.r.t. $\{x_1, x_2\}$.

Note 4 It is easy to see that if $\varphi \succ_{(\sigma,F)} X$ and $Z \subseteq X$, then $\varphi \succ_{(\sigma,F)} Z$. In particular: if $\varphi \succ_{(\sigma,F)} X$ for some X then φ is (σ, F)-absolute. [9]

Examples.

- Let σ include $\overrightarrow{P} = \{P_1, \ldots, P_k\}$. Assume that the arity of P_i is n_i, and define $F(P_i) = \{\{1, \ldots, n_i\}\}$, and $F(Q) = \emptyset$ in case $Q \notin \overrightarrow{P}$. Then φ is $(\sigma, F)-$d.i. w.r.t. X iff it is $\overrightarrow{P}-$d.i. w.r.t. X in the sense of Definition 5.
- Let $\sigma_{ZF} = \{\in\}$ and let $F_{ZF}(\in) = \{\{1\}\}$. Then S_2 is a $(\sigma_{ZF}, F_{ZF})-$extension of S_1 iff $S_1 \subseteq_{\sigma_{ZF}} S_2$, and $x_1 \in x_2 \succ^{S_1;S_2} \{x_1\}$. The latter condition means that S_1 is a transitive substructure of S_2 (In particular, the universe V is a $(\sigma_{ZF}, F_{ZF})-$extension of the transitive sets and classes). Therefore $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_k) \succ_{(\sigma_{ZF},F_{ZF})} \{x_1, \ldots, x_n\}$ iff the following holds whenever S_1 is a transitive substructure of S_2, and $y_1, \ldots, y_k \in S_1$:

$$\{\langle x_1, \ldots, x_n\rangle \mid S_1 \models \varphi\} = \{\langle x_1, \ldots, x_n\rangle \mid S_2 \models \varphi\}$$

  In particular, a formula is (σ_{ZF}, F_{ZF})-absolute iff it is absolute in the usual sense this notion is used in set theory.
- Assume that $F(p)$ is nonempty for every p in σ. Then (see Note 4) S_1 is a substructure of S_2 (in the usual sense of model theory) whenever S_2 is a (σ, F)-extension of S_1.

Theorem 1 *The property of* (σ, F)*-absoluteness is in general undecidable* [10] .

Proof: We show it in the case where σ has no constants, and $F(p) = \{\emptyset\}$ for all p. Let φ be any sentence. We prove that the formula $x = x \wedge \varphi$ is (σ, F)-

[9] In [3] the values of the function F were demanded to be closed under subsets. By the present note, this condition is not really necessary.

[10] A similar result was proved in [6] for d.i. in databases. However, this was done under the assumption that the interpretations of all the predicate symbols are finite. Here we do not assume this.

absolute iff φ is either logically valid or a logical contradiction. Assume that it is neither. Then there are structure S_t and S_f for σ such that φ is true in S_t and false in S_f. Without loss in generality, we may assume that S_t and S_f are disjoint. Let S be the structure for σ whose domain is the union of the domains of S_t and S_f, and the interpretation of any predicate p in it is the union of the interpretations of p in S_t and S_f. It is easy to see that S is a (σ, F)-extension of both S_t and S_f. Obviously, it is impossible that both $x = x \wedge \varphi \succ^{S_t;S} \emptyset$ and $x = x \wedge \varphi \succ^{S_f;S} \emptyset$. Hence $x = x \wedge \varphi$ is not (σ, F)-absolute in this case. On the other hand it is easy to see that it is (σ, F)-absolute if φ is logically valid or a logical contradiction. It follows that had (σ, F)-absoluteness been decidable, logical validity of formulas in σ would have been decidable. This is not always the case, of course. □

It follows from Theorem 1 that the semantic relation of $(\sigma, F)-$d.i. is undecidable. Hence again it should be replaced in practice by a useful syntactic approximation. Now the most natural way to define a syntactic approximation of a semantic logical relation concerning formulas is by a structural induction. Such an inductive definition should be based on the behavior of the atomic formulas and of the logical connectives and quantifiers with respect to the original semantic relation. The next theorem lists the most obvious and useful relevant properties of $\succ_{(\sigma,F)}$:

Theorem 2 $\succ_{(\sigma,F)}$ *has the following properties:*

(1) $p(t_1, \ldots, t_n) \succ_{(\sigma,F)} X$ *in case* p *is an* n*-ary predicate symbol of* σ*, and there is* $I \in F(p)$ *such that:*
  (a) For every $x \in X$ *there is* $i \in I$ *such that* $x = t_i$*.*
  (b) $X \cap Fv(t_j) = \emptyset$ *for every* $j \in \{1, \ldots, n\} - I$*.*

(2) (a) $\varphi \succ_{(\sigma,F)} \{x\}$ *if* $\varphi \in \{x \neq x, x = t, t = x\}$*, and* $x \notin Fv(t)$*.*
  (b) $t = s \succ_{(\sigma,F)} \emptyset$*.*

(3) $\neg\varphi \succ_{(\sigma,F)} \emptyset$ *if* $\varphi \succ_{(\sigma,F)} \emptyset$*.*

(4) $\varphi \vee \psi \succ_{(\sigma,F)} X$ *if* $\varphi \succ_{(\sigma,F)} X$ *and* $\psi \succ_{(\sigma,F)} X$*.*

(5) $\varphi \wedge \psi \succ_{(\sigma,F)} X \cup Y$ *if* $\varphi \succ_{(\sigma,F)} X$*,* $\psi \succ_{(\sigma,F)} Y$*, and* $Y \cap Fv(\varphi) = \emptyset$*.*

(6) $\exists y\varphi \succ_{(\sigma,F)} X - \{y\}$ *if* $y \in X$ *and* $\varphi \succ_{(\sigma,F)} X$*.*

(7) If $\varphi \succ_{(\sigma,F)} \{x_1, \ldots, x_n\}$*, and* $\psi \succ_{(\sigma,F)} \emptyset$*, then* $\forall x_1 \ldots x_n(\varphi \to \psi) \succ_{(\sigma,F)} \emptyset$*.*

Proof: Most of the proofs of the various properties are straightforward. We do (5),(6), and (7) as examples. In the following, we assume that S_2 is a (σ, F)-extension of S_1.

For property 5, assume that $\theta = \varphi \wedge \psi$, where $\varphi \succ_{(\sigma,F)} X$, $\psi \succ_{(\sigma,F)} Y$, and $Y \cap Fv(\varphi) = \emptyset$. To simplify notation, assume that $Fv(\varphi) = \{x, z\}$, $Fv(\psi) = Fv(\theta) = \{x, y, z\}$, $X = \{x\}$, $Y = \{y\}$. Let $Z(c) = \{x \in S_2 \mid S_2 \models \varphi(x, c)\}$ for $c \in S_1$. Since $\varphi \succ_{(\sigma,F)} X$, $Z(c) = \{x \in S_1 \mid S_1 \models \varphi(x, c)\}$ as well. Hence $Z(c) \subseteq S_1$. This and the fact that $\psi \succ_{(\sigma,F)} Y$ imply that if $d \in Z(c)$

then $\{y \in S_2 \mid S_2 \models \psi(d,y,c)\} = \{y \in S_1 \mid S_1 \models \psi(d,y,c)\}$. Denote this set by $W(c,d)$. Now both of the sets $\{\langle x,y\rangle \in S_2^2 \mid S_2 \models \theta(x,y,c)\}$ and $\{\langle x,y\rangle \in S_1^2 \mid S_1 \models \theta(x,y,c)\}$ equal the union of the sets $\{d\} \times W(c,d)$ for $d \in Z(c)$. Hence these two sets are the same, and so $\theta \succ_{(\sigma,F)} \{x,y\}$, which is what we need to prove.

For property 6, assume that $\psi = \exists y\varphi$, where $\varphi \succ_{(\sigma,F)} X$, and $y \in X$. To simplify notation, assume that $Fv(\varphi) = \{x,y,z\}$, and $X = \{x,y\}$. Now for $c \in S_1$, $\{\langle x,y\rangle \in S_2^2 \mid S_2 \models \varphi(x,y,c)\}= \{\langle x,y\rangle \in S_1^2 \mid S_1 \models \varphi(x,y,c)\}$, since $\varphi \succ_{(\sigma,F)} X$. This immediately implies that also $\{x \in S_2 \mid S_2 \models \psi(x,c)\} = \{x \in S_1 \mid S_1 \models \psi(x,c)\}$, since these two sets are just the projections on the second component of the above equal sets. Hence $\psi \succ_{(\sigma,F)} \{x\}$, which is what we need to prove.

Finally, property 7 follows from properties 3, 5, and 6, since $\forall x_1 \ldots x_n(\varphi \rightarrow \psi)$ is equivalent to $\neg\exists x_1 \ldots x_n(\varphi \wedge \neg\psi)$. $\square$

Note 5 Using exactly the same argument, we can actually prove a stronger result: For every S_1 and S_2 such that S_1 is a (σ, F)-substructure of S_2, the relation $\succ^{S_1;S_2}$ has the properties 1–7 from Theorem 2.

Note 6 Theorem 2 remains true for languages which include more complex terms (not just variables and constants), provided that $x = t \succ_{(\sigma,F)} \{x\}$ whenever $x \notin Fv(t)$.

Now Theorem 2 naturally leads to the following syntactic relation:

Definition 9 $\succ^s_{(\sigma,F)}$ is the least relation which has the properties of $\succ_{(\sigma,F)}$ listed in Theorem 2 [11].

Corollary 1 *If $\varphi \succ^s_{(\sigma,F)} X$ then $\varphi \succ_{(\sigma,F)} X$. The converse might fail.*

Proof: Immediate from Theorems 2 and 1. $\square$

Note that $\succ^s_{(\sigma,F)}$ is a direct generalization of $\succ^s_{\overrightarrow{P}}$. On the other hand, the set of Δ_0-formulas of σ_{ZF} is obviously a subset of $\{\varphi \mid \varphi \succ^s_{(\sigma_{ZF},F_{ZF})} \emptyset\}$.

[11] Property 7 is easily derivable from the others. Hence if $\forall$ and $\rightarrow$ are taken as defined in terms of the other logical constants, then the same relation is obtained if we omit property 7 from the list in Theorem 2.

4 The Role of Absoluteness in Effective Structures

We turn in this section to explore the relations between d.i. and absoluteness on one hand, computability and decidability on the other. For this we assume in most of the theorems and proofs an intuitive understanding of the notions of "effectivity", "computability", and "decidability" (of the type assumed in textbooks when proving that all recursive functions are "computable"). By Church thesis, valid exact mathematical theorems are obtained from our results whenever these notions are translated (perhaps via some "effective" coding) into an appropriate, precisely defined, notion of recursiveness (and our proofs can then easily, though tediously, be converted into full proofs of these theorems). However, our results remain true for stronger (or even weaker) notions of "effectiveness" (e.g. relative ones) which satisfy certain simple closure conditions (which are implicit in our proofs).

4.1 Absoluteness and Decidability

Definition 10 Let (σ, F) be a d.i.-signature, and let S be a structure for σ. A formula φ is called $(S, F)-$*d.i. w.r.t.* X ($\varphi \succ_{(S,F)} X$) if $\varphi \succ^{S';S} X$ whenever S' is a $(\sigma, F)-$substructure of S. φ is $(S, F)-$*absolute* if $\varphi \succ_{(S,F)} \emptyset$.

Obviously, if S is a structure for σ, then $\succ^{s}_{(\sigma,F)} \subseteq \succ_{(\sigma,F)} \subseteq \succ_{(S,F)}$. In particular: if $\varphi \succ^{s}_{(\sigma,F)} \emptyset$ then φ is $(S, F)-$absolute. The converses are not true (by Theorem 1 and the example concerning $\mathcal{N}$ which is given below). It is also easy to show that $\succ_{(S,F)}$ has the properties 1–7 from Theorem 2 (using the same arguments as in the proof of that theorem).

Definition 11 Let (σ, F) be a d.i.-signature, and let S be a structure for σ.

(1) S is *effectively* $(\sigma, F)-$*locally finite* if for every finite subset A of S one can effectively find a finite $(\sigma, F)-$substructure S' of S s. t. $A \subseteq S'$.
(2) S is $(\sigma, F)-$*effective* if S is effectively $(\sigma, F)-$locally finite, and all the interpretations in S of the predicates of σ are decidable.
(3) S is *strongly* $(\sigma, F)-$*effective* if it is $(\sigma, F)-$effective, and the elements of S can effectively be enumerated.

Examples.

Databases Assume that σ is finite. Then every structure S for σ, in which the interpretations of all the predicate symbols (except equality) are finite, is $(\sigma, F)-$effective: Given A, let S' be the substructure of S whose domain is the union of A, the set of the interpretations in S of the constants of σ, and all the domains of the interpretations in S of the predicates of σ.

The Natural Numbers Define the d.i. signature $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$ as follows:

- $\sigma_{\mathcal{N}}$ is the first-order signature which includes the constant 0, the binary predicate $<$, and the ternary relations P_+ and $P_\times$.
- $F_{\mathcal{N}}(<) = \{\{1\}\}$, $F_{\mathcal{N}}(P_+) = F_{\mathcal{N}}(P_\times) = \{\emptyset\}$.

The standard structure $\mathcal{N}$ for $\sigma_{\mathcal{N}}$ has the set N of natural numbers as its domain, with the usual interpretations of 0 and $<$, and the (graphs of the) operations $+$ and $\times$ on N (viewed as ternary relations on N) as the interpretations of P_+ and $P_\times$, respectively. It is easy to see that $\mathcal{N}$ is a $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$-extension of a structure S for $\sigma_{\mathcal{N}}$ iff the domain of S is an initial segment of $\mathcal{N}$ (where the interpretations of the relation symbols are the corresponding reductions of the interpretations of those symbols in $\mathcal{N}$). The same will be true if we replace $<$ by the binary predicate $Succ$ (with "y is the successor of x" as the intended interpretation of $Succ(x, y)$), and let $F_{\mathcal{N}}(Succ) = \{\{1\}\}$, or if we delete $<$ altogether, and let $F_{\mathcal{N}}(P_+) = \{\{1, 2\}\}$. It follows that $\mathcal{N}$ and its variants are strongly $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})-$effective. Note that φ is $(\mathcal{N}, F_{\mathcal{N}})$-absolute if for any assignment v in N, φ gets the same truth value in all initial segments of $\mathcal{N}$ (including $\mathcal{N}$ itself) which contain the values assigned by v to its free variables. Thus $\forall y(y = 0 \vee \exists z Succ(z, y))$ is $(\mathcal{N}, F_{\mathcal{N}})$-absolute, even though it is clearly not $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$-absolute.

S-expressions Following [8], let V_0 be the set of Lisp's S-expressions, i.e.: all the expressions generated from 0 (or *nil*) using the pairing operation. [12] Construct a corresponding d.i. signature (σ_V, F_V) by letting σ_V have a constant 0 and a ternary relation *pair* (where $pair(x, y, z)$ is interpreted in V_0 as "$z = \langle x, y\rangle$"), with $F_V(pair) = \{\{1, 2\}\}$. It is easy to see that V_0 is strongly (σ_V, F_V)-effective (Similar treatments can be given to other data structures used for computations, like strings of Symbols from some finite alphabet).

Hereditarily finite sets Obviously, the structure $\mathcal{H}F$ of hereditarily finite sets is strongly $(\sigma_{ZF}, F_{ZF})-$effective. This example is particularly interesting, because of Corollary 3 below.

Next we present some results which connect d.i. to computability, and absoluteness with decidability.

Theorem 3 *If S is $(\sigma, F)-$effective, and*

$$\varphi(x_1, \ldots, x_n, y_1, \ldots, y_k) \succ_{(S,F)} \{x_1, \ldots, x_n\}$$

then $G_\varphi = \lambda y_1, \ldots, y_k.\{\langle x_1, \ldots, x_n\rangle \in S^n \mid S \models \varphi(x_1, \ldots, x_n, y_1, \ldots, y_k)\}$ is a computable function from S^k to the set of finite *subsets of S^n.*

---

[12] V_0 was suggested in [8] as a framework for computability theory and metamathematics which is superior to $\mathcal{N}$.

Proof: Let $b_1, \ldots, b_k$ be elements of S. To compute $G_\varphi(b_1, \ldots, b_k)$, find a finite $(\sigma, F)-$substructure S_1 of S whose domain S_1 include $b_1, \ldots, b_k$, and compute $\{\langle x_1, \ldots, x_n \rangle \in S_1^n \mid S_1 \models \varphi(x_1, \ldots, x_n, b_1, \ldots, b_k)\}$. This is possible, and the result is finite, because S_1 is finite, and the interpretations of the predicates of σ in S are decidable. Now the result is $G_\varphi(b_1, \ldots, b_k)$, because $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_k) \succ_{(S,F)} \{x_1, \ldots, x_n\}$, and $b_1, \ldots, b_k \in S_1$. $\square$

Corollary 2 *Let S be like in Theorem 3, and let φ be $(S, F)-$absolute, with $Fv(\varphi) = \{x_1, \ldots, x_n\}$. Then $\{\langle a_1, \ldots, a_n \rangle \in S^n \mid S \models \varphi(a_1, \ldots, a_n)\}$ is a decidable n-ary relation on S. In particular, this is true for φ if $\varphi \succ^s_{(\sigma,F)} \emptyset$.*

One application of theorem 3 is the well-known fact (see Note 2) that given a database DB with scheme $\overrightarrow{P}$, the answer to any $\overrightarrow{P}$-d.i. query is finite and can effectively be computed in any structure for DB.

Other important applications are for the structures $\mathcal{N}$, V_0, and $\mathcal{HF}$. Thus it follows from Corollary 2 that every formula φ such that $\varphi \succ^s_{(\sigma_V, F_V)} \emptyset$ defines a decidable relation on V_0. Similar results obtain of course for $\mathcal{N}$ and $\mathcal{HF}$. For the latter we have also:

Corollary 3 *If $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_k) \succ^s_{(\sigma_{ZF}, F_{ZF})} \{x_1, \ldots, x_n\}$ then $G_\varphi = \lambda y_1, \ldots, y_k. \{\langle x_1, \ldots, x_n \rangle \in \mathcal{HF}^n \mid \mathcal{HF} \models \varphi(x_1, \ldots, x_n, y_1, \ldots, y_k)\}$ is a computable function from $\mathcal{HF}^k$ to $\mathcal{HF}$.*

The relations $\succ^s_{(\sigma_\mathcal{N}, F_\mathcal{N})}$ and $\succ^s_{(\sigma_{ZF}, F_{ZF})}$ are quite interesting. For convenience, denote them by $\succ^s_\mathcal{N}$ and $\succ^s_{\mathcal{HF}}$, respectively. It can easily be seen that $\succ^s_\mathcal{N}$ is the minimal relation which satisfies the following conditions:

(1) $\varphi \succ^s_\mathcal{N} \emptyset$ if φ is atomic.
(2) $\varphi \succ^s_\mathcal{N} \{x\}$ if $\varphi \in \{x \neq x, x = t, t = x, x < t\}$, and $x \notin Fv(t)$.
(3) $\neg\varphi \succ^s_\mathcal{N} \emptyset$ if $\varphi \succ^s_\mathcal{N} \emptyset$.
(4) $\varphi \vee \psi \succ^s_\mathcal{N} X$ if $\varphi \succ^s_\mathcal{N} X$ and $\psi \succ^s_\mathcal{N} X$.
(5) $\varphi \wedge \psi \succ^s_\mathcal{N} X \cup Y$ if $\varphi \succ^s_\mathcal{N} X$, $\psi \succ^s_\mathcal{N} Y$, and $Y \cap Fv(\varphi) = \emptyset$.
(6) $\exists y \varphi \succ^s_\mathcal{N} X - \{y\}$ if $y \in X$ and $\varphi \succ^s_\mathcal{N} X$.
(7) $\forall y_1, \ldots, y_n(\varphi \rightarrow \psi) \succ^s_\mathcal{N} \emptyset$ if $\varphi \succ^s_\mathcal{N} \{y_1, \ldots, y_n\}$ and $\psi \succ^s_\mathcal{N} \emptyset$.

$\succ^s_{\mathcal{HF}}$ has an almost identical characterization. The only difference is that in the second clause $<$ should be replaced by $\in$.[13]

It follows from the above characterization of $\succ^s_\mathcal{N}$ that the set of formulas φ such that $\varphi \succ^s_\mathcal{N} \emptyset$ is a straightforward extension of Smullyan's set of bounded formulas ([16]).

[13] Note again that in both cases condition (7) is superfluous if we take $\forall$ and $\rightarrow$ as defined in terms of the other connectives and $\exists$.

Problem For a set of formulas A, let $R(A)$ denote the set of relations on N which are defined by some formula in A. From our results it follows that

$$R(\{\varphi \mid \varphi \succ^s_{\mathcal{N}} \emptyset\}) \subseteq R(\{\varphi \mid \varphi \succ_{(\sigma_{\mathcal{N}},F_{\mathcal{N}})} \emptyset\}) \subseteq R(\{\varphi \mid \varphi \succ_{(\mathcal{N},F_{\mathcal{N}})} \emptyset\}) \subseteq DEC$$

where DEC is the set of decidable relations on N. Now the problem is: which of these three inclusions is actually an equality, and which of them is proper? (since it is easy to show that the first set in this chain is a proper subset of the last, at least one of the three inclusions is proper). A similar problem exists concerning $\mathcal{H}F$ and (σ_{ZF}, F_{ZF}).

With the aid of Church Thesis, things become much clearer when we consider semi-decidability rather than decidability. According to the thesis, the semi-decidable relations on N (or HF) are precisely the recursively enumerable (r.e.) ones, and for the latter we have the following theorem (which provides another crucial connection between absoluteness and decidability):

Theorem 4 *The following conditions are equivalent for a relation R on N:*

(1) R is r.e.
(2) R is definable by a formula of the form $\exists y_1, \ldots, y_n\psi$, where $\psi \succ^s_{\mathcal{N}} \emptyset$.
(3) R is definable by a formula of the form $\exists y_1, \ldots, y_n\psi$, where the formula ψ is $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$-absolute.
(4) R is definable by a formula of the form $\exists y_1, \ldots, y_n\psi$, where the formula ψ is $\succ_{(\mathcal{N},F_{\mathcal{N}})}$-absolute.

A similar result obtains if instead of $\mathcal{N}$, $\sigma_{\mathcal{N}}$, and $F_{\mathcal{N}}$ we consider $\mathcal{H}F$, $\sigma_{\mathcal{Z}F}$, and $F_{\mathcal{Z}F}$, respectively,

Proof: We do the proof in the case of $\mathcal{N}$. In this case (2) follows from (1) by Smullyan's characterization in [16] of the r.e. subsets of N using his set of bounded formulas (recall that if ψ is bounded, then $\psi \succ^s_{\mathcal{N}} \emptyset$). Obviously, (3) follows from (2), and (4) follows from (3). To show that (4) entails (1), assume that R is definable by a formula of the form $\exists y_1, \ldots, y_n\psi$, where the formula $\psi(x_1, \ldots, x_k, y_1, \ldots, y_n)$ is $(\mathcal{N}, F_{\mathcal{N}})$-absolute. Given numbers $n_1, \ldots, n_k$ we search whether $R(n_1, \ldots, n_k)$ by examining all the finite initial segments of N that contain $n_1, \ldots, n_k$, and return "true" if we find in one of them numbers $m_1, \ldots, m_n$ such that $\psi(n_1, \ldots, n_k, m_1, \ldots, m_n)$ is true in it. From the fact that ψ is $(\mathcal{N}, F_{\mathcal{N}})$-absolute, it easily follows that this procedure halts with the correct answer in case $R(n_1, \ldots, n_k)$, and never halt otherwise. It follows that R is semi-decidable, and so it is r.e. (by Church Thesis or by a direct translation of this argument to a precise proof). $\square$

It follows from the last theorem that according to Church Thesis, the semi-decidable relations on N are precisely the projections of the absolute relations on $\mathcal{N}$ (where a relation on $\mathcal{N}$ is absolute iff it is definable by a $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$-absolute formula). A similar result (which uses a more natural language) obtains for $\mathcal{H}F$. These are purely model-theoretic consequences of the Thesis.

4.2 Upward Absoluteness and Semi-decidability

Theorem 4 suggests that formulas of the form $\exists y_1, \ldots, y_n \psi$, where ψ is (σ, F)-absolute, may have a special interest in general. Next we turn our attention to an obvious property that these formulas have, and which might be crucial for their connection with semi-decidability.

Definition 12 Let (σ, F) be a d.i.-signature, and let S' and S be structures for σ such that S' is a (σ, F)-substructure of S. A formula $\varphi(x_1, \ldots, x_n)$ in σ is upward (σ, F)-absolute (notation: $(\sigma, F)-\uparrow$) with respect to S' and S if for all $a_1, \ldots, a_n \in S'$, if $S' \models \varphi(a_1, \ldots, a_n)$ then $S \models \varphi(a_1, \ldots, a_n)$. φ is upward (S, F)-absolute (notation: $(S, F)-\uparrow$) if it is $(\sigma, F)-\uparrow$ with respect to S' and S for every S' which is a (σ, F)-substructure of S. φ is $(\sigma, F)-\uparrow$ (upward (σ, F)-absolute) if it is $(S, F)-\uparrow$ for every structure S for σ.

Theorem 5 *Upward (σ, F)-absoluteness has the following properties:*

(1) If φ is (σ, F)-absolute then φ is $(\sigma, F)-\uparrow$.
(2) If φ and ψ are $(\sigma, F)-\uparrow$ then so are $\varphi \vee \psi$ and $\varphi \wedge \psi$.
(3) If φ is $(\sigma, F)-\uparrow$ then so is $\exists x \varphi$.
(4) If $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_k) \succ_{(S,F)} \{y_1, \ldots, y_k\}$, and $\psi(\overrightarrow{x}, \overrightarrow{y})$ is $(\sigma, F)-\uparrow$, then $\forall y_1, \ldots, y_k(\varphi \to \psi)$ is $(\sigma, F)-\uparrow$.

The same is true for upward (S, F)-absoluteness.

Proof: We prove the last property as an example (the rest are straightforward). So assume that φ and ψ have the relevant properties, S' is a (σ, F)-substructure of S, and $a_1, \ldots, a_n$ are elements of S' such that

$$(*) \quad S' \models \forall y_1, \ldots, y_k(\varphi(a_1, \ldots, a_n, y_1, \ldots, y_k) \to \psi(a_1, \ldots, a_n, y_1, \ldots, y_k))$$

We show that also

$$S \models \forall y_1, \ldots, y_k(\varphi(a_1, \ldots, a_n, y_1, \ldots, y_k) \to \psi(a_1, \ldots, a_n, y_1, \ldots, y_k))$$

Let $b_1, \ldots, b_k$ be arbitrary elements of S. We should show that

$$(@) \quad S \models \varphi(a_1, \ldots, a_n, b_1, \ldots, b_k) \to \psi(a_1, \ldots, a_n, b_1, \ldots, b_k)$$

This is obvious if $S \not\models \varphi(a_1, \ldots, a_n, b_1, \ldots, b_k)$. So assume the opposite. Since $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_k) \succ_{(S,F)} \{y_1, \ldots, y_k\}$, and $a_1, \ldots, a_n$ are elements of S', this assumption implies that $b_1, \ldots, b_k$ are all elements of S' too, and that $S' \models \varphi(a_1, \ldots, a_n, b_1, \ldots, b_k)$. From this and $(*)$ it follows that S' is a model of $\psi(a_1, \ldots, a_n, b_1, \ldots, b_k)$. Hence $S \models \psi(a_1, \ldots, a_n, b_1, \ldots, b_k)$ as well (because ψ is $(\sigma, F)- \uparrow$), and so (@) is true. $\square$

Theorem 5 naturally leads to the following Definition and Corollary:

Definition 13 Let (σ, F) be a d.i.-signature. $\Sigma(\sigma, F)$ is the least set of formulas which has the properties of $(\sigma, F)- \uparrow$ listed in Theorem 5.

Corollary 4 *Every formula in $\Sigma(\sigma, F)$ is upward (σ, F)-absolute.*

Problems: Is every formula which is upward (σ, F)-absolute logically equivalent to some formula in $\Sigma(\sigma, F)$? Or even to a formula of the form $\exists y_1, \ldots, y_n\psi$, where ψ is (σ, F)-absolute? And given a structure S for σ, is every formula which is upward (S, F)-absolute equivalent in S to some formula in $\Sigma(\sigma, F)$? Or to a formula of the form $\exists y_1, \ldots, y_n\psi$, where ψ is (σ, F)-absolute? Or to a formula of the form $\exists y_1, \ldots, y_n\psi$, where ψ is (S, F)-absolute?

It is very easy to see that the set $\Sigma(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$ is a superset of the set Σ of formulas in the language $\sigma_{\mathcal{N}}$ (as defined e.g. in [16]). The latter is used in [16] to characterize (using Church Thesis) the semi-decidable subsets of N. The next theorem suggests a general strong analogy between semi-decidable formulas and upward absolute formulas (Again we assume in its proof only an intuitive understanding of "semi-decidable": $\varphi(x_1, \ldots, x_n)$ is semi-decidable in a structure S if there exists an effective procedure, that given a tuple $\langle a_1, \ldots, a_n\rangle \in S^n$ halts iff that tuple satisfies φ in S).

Theorem 6 *If S is strongly $(\sigma, F)-$effective then the set of formulas which are semi-decidable in S (i.e. define semi-decidable relations on S) has all the properties of $(\sigma, F)- \uparrow$ listed in Theorem 5.*

Proof: That if φ is (S, F)-absolute then φ defines a semi-decidable relation on S immediately follows from Corollary 2.

Assume that φ and ψ are both semi-decidable in S. Given a tuple $\overrightarrow{a}$, to decide whether $\overrightarrow{a}$ satisfies $\varphi \wedge \psi$, check first whether it satisfies φ. If it does, check whether it satisfies ψ. To decide whether $\overrightarrow{a}$ satisfies $\varphi \vee \psi$, check in parallel (or by a fair interleaving) whether it satisfies φ, and whether it satisfies ψ. Halt when one of them succeeds.

Assume that $\varphi(\overrightarrow{x}, y)$ is semi-decidable in S. To decide whether $\overrightarrow{a}$ satisfies $\exists y\varphi$, check (by a fair interleaving) for every $b \in S$ whether $\langle a_1, \ldots, a_n, b\rangle$ satisfies φ (this is possible by the strong effectivity of S). Halt once one succeeds.

Finally, assume that $\varphi(x_1,\ldots,x_n,y_1,\ldots,y_k) \succ_{(S,F)} \{y_1,\ldots,y_k\}$, and that $\psi(x_1,\ldots,x_n,y_1,\ldots,y_k)$ is semi-decidable in S. To decide whether $\overrightarrow{a}$ satisfies $\forall y_1,\ldots,y_k(\varphi \rightarrow \psi)$, find first all the tuples in $\{\overrightarrow{b} \in S^k \mid S \models \varphi(\overrightarrow{a},\overrightarrow{b})\}$. By Theorem 3 this set is finite and computable. Now check in parallel (or by a fair interleaving) for each $\overrightarrow{b}$ in this set whether $S \models \psi(\overrightarrow{a},\overrightarrow{b})\}$. □

Corollary 5 *If S is strongly $(\sigma, F)-$effective then every formula in $\Sigma(\sigma, F)$ is semi-decidable in S.*

Corollary 6 *A relation on N is semi-decidable iff it is definable by some formula in $\Sigma(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$.*

Proof: This is immediate from the previous Corollary and Theorem 4. □

Problems: It follows from Corollaries 4 and 6 that every semi-decidable relation on N is definable by an upward $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$-absolute formula. It is not clear whether the converse is also true. It is also not known whether semi-decidability implies upward (σ, F)-absoluteness in every strongly (σ, F)-effective structure.

Despite the intimate relationship that the results of this section suggest, further research is needed in order to understand the full connection between upward absoluteness and semi-decidability. Following the basic idea of [2], it seems to us very likely that in order to provide a satisfactory answer (and develop an appropriate general theory), one should go beyond first-order languages by introducing into the language an operation TC for the transitive closure of (definable) predicates. We leave that for future investigations.

5 Domain Independence and Predicativity

To complete the picture, we return in this section to the area in which the notion of absoluteness has first been introduced: set theory. We do it briefly, leaving most details and discussions (and all proofs) to a future paper.

In Section 3 we have noted that the notion of (σ_{ZF}, F_{ZF})-absoluteness is identical to Gödel's original notion of absoluteness, and that $\{\varphi \mid \varphi \succ^s_{\mathcal{H}F} \emptyset\}$ is a natural extension of the set of Δ_0-formulas in the language of σ_{ZF}. However, in order to fully exploit the power of the idea of partial d.i. in the framework of set theory, we need to use a language which is stronger (and more natural) than the official language of ZF. The main feature of the stronger language is that it employs a rich class of set terms of the form $\{x \mid \varphi\}$. Of course, not every formula φ can be used in such a term. The basic idea is to allow only formulas which are d.i. with respect to $\{x\}$. Intuitively, in such a case the term $\{x \mid \varphi\}$ denotes a *set* with an *absolute* identity. This set is "effectively" *constructed* from the (values of the) parameters of φ and the sets referred to in φ (this is made precise in Theorem 8 below). Since d.i. is a semantic notion, we use instead a formal approximation $\succ_{RST}$. $\succ_{RST}$ is basically the natural extension of $\succ^s_{\mathcal{H}F}$ to our reacher language. However, the definition of that very language depends in turn on that of $\succ_{RST}$. Accordingly, the sets of terms and formulas of our language, and the relation $\succ_{RST}$, are defined together by a simultaneous induction:

Definition 14 The language $\mathcal{L}_{RST}$ is defined as follows:

Terms:
(1) Every variable is a term.
(2) If x is a variable, and φ is a formula such that $\varphi \succ_{RST} \{x\}$, then $\{x \mid \varphi\}$ is a term (and $Fv(\{x \mid \varphi\}) = Fv(\varphi) - \{x\}$).
Formulas:
(1) If t and s are terms than $t = s$ and $t \in s$ are atomic formulas.
(2) If φ and ψ are formulas, and x is a variables, then $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$ $(\varphi \rightarrow \psi)$, $\forall x\varphi$, and $\exists x\varphi$ are formulas.
The d.i. relation $\succ_{RST}$:
(1) $\varphi \succ_{RST} \emptyset$ if φ is atomic.
(2) $\varphi \succ_{RST} \{x\}$ if $\varphi \in \{x \neq x, x = t, t = x, x \in t\}$, and $x \notin Fv(t)$.
(3) $\neg\varphi \succ_{RST} \emptyset$ if $\varphi \succ_{RST} \emptyset$.
(4) $\varphi \vee \psi \succ_{RST} X$ if $\varphi \succ_{RST} X$ and $\psi \succ_{RST} X$.
(5) $\varphi \wedge \psi \succ_{RST} X \cup Y$ if $\varphi \succ_{RST} X$, $\psi \succ_{RST} Y$, and $Y \cap Fv(\varphi) = \emptyset$.
(6) $\exists y\varphi \succ_{RST} X - \{y\}$ if $y \in X$ and $\varphi \succ_{RST} X$.
(7) $\forall y_1, \ldots, y_n(\varphi \rightarrow \psi) \succ_{RST} \emptyset$ if $\varphi \succ_{RST} \{y_1, \ldots, y_n\}$ and $\psi \succ_{RST} \emptyset$.

Note 7 $\succ_{RST}$ is a syntactic approximation of an intuitive set-theoretical relation of "universe-independence" (see part (2) of Theorem 8 below). Note that it is defined using *exactly* the same clauses used to characterize $\succ^s_{\mathcal{H}F}$ in Section 4 (after Corollary 3). However, in the case of $\succ_{RST}$ the first two clauses refer to richer classes of terms and atomic formulas than they do in the case of $\succ^s_{\mathcal{H}F}$.

Here are some examples of valid terms of $\mathcal{L}_{RST}$:

- $\emptyset =_{Df} \{x \mid x \neq x\}$
- $\{t_1, \ldots, t_n\} =_{Df} \{x \mid x = t_1 \vee \ldots \vee x = t_n\}$ (where x is new).
- $\langle t, s\rangle =_{Df} \{\{t\}, \{t, s\}\}$.
- $\{x \in t \mid \varphi\} =_{Df} \{x \mid x \in t \wedge \varphi\}$, provided $\varphi \succ_{RST} \emptyset$. (where $x \notin Fv(t)$).
- $\{t \mid x \in s\} =_{Df} \{y \mid \exists x.x \in s \wedge y = t\}$ (where y is new, and $x \notin Fv(s)$).
- $s \times t =_{Df} \{x \mid \exists a \exists b.a \in s \wedge b \in t \wedge x = \langle a, b\rangle\}$ (where x, a and b are new).
- $\bigcup t =_{Df} \{x \mid \exists y.y \in t \wedge x \in y\}$

The following theorem and its two corollaries determine the expressive power of $\mathcal{L}_{RST}$, and connect it (and $\succ_{RST}$) with the class of rudimentary set functions — a refined version of Gödel basic set functions which was independently introduced by Gandy in [10] and by Jensen in [12] (See also [5]). For simplicity of presentation, we assume in them the platonic universe V of ZF (that a language even stronger than $\mathcal{L}_{RST}$ has a semantics in V was proved in [4]).

Theorem 7

(1) If F is an n-ary rudimentary function, then there exists a formula φ with the following properties:
  (a) $Fv(\varphi) = \{y, x_1, \ldots, x_n\}$
  (b) $\varphi \succ_{RST} \{y\}$
  (c) $F(x_1, \ldots, x_n) = \{y \mid \varphi\}$.

(2) If φ is a formula such that:
  (a) $Fv(\varphi) = \{y_1, \ldots, y_k, x_1, \ldots, x_n\}$
  (b) $\varphi \succ_{RST} \{y_1, \ldots, y_k\}$
  then there exists a rudimentary function F such that:

$$F(x_1, \ldots, x_n) = \{\langle y_1, \ldots, y_k\rangle \mid \varphi\}$$

Corollary 7 *Every term of $\mathcal{L}_{RST}$ with n free variables explicitly defines an n-ary rudimentary function. Conversely, every rudimentary function is defined by some term of $\mathcal{L}_{RST}$.*

Corollary 8 *If $Fv(\varphi) = \{x_1, \ldots, x_n\}$, and $\varphi \succ_{RST} \emptyset$, then φ defines a rudimentary predicate P. Conversely, if P is a rudimentary predicate, then there is a formula φ such that $\varphi \succ_{RST} \emptyset$, and φ defines P.*

Next we introduce the most basic formal set theory in the language $\mathcal{L}_{RST}$:

Definition 15 RST is the first-order theory with equality in the language $\mathcal{L}_{RST}$ [14] which has the following axioms:

- Extensionality: $\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y$
- Comprehension: $\forall x(x \in \{x \mid \varphi\} \leftrightarrow \varphi)$ (where $\varphi \succ_{RST} \{x\}$).

Our final theorem uses RST to clarify the connection between $\succ_{RST}$ and d.i. (and absoluteness) in the context of set theory. Again for simplicity we assume in it the universe V of ZF.

[14] $\mathcal{L}_{RST}$ has richer classes of terms than those allowed in orthodox first-order systems. In particular: a variable can be bound in them within a term. The notion of a term being free for substitution should be extended accordingly. Otherwise the rules/axioms concerning the quantifiers, terms, and equality remain unchanged.

Theorem 8 *Let $\mathcal{M}$ be a transitive (in V) model of RST.*

(1) If t is a term of $\mathcal{L}_{RST}$ with $Fv(t) = \{x_1, \ldots, x_n\}$, then

$$V \models \forall x_1 \ldots \forall x_n . x_1 \in \mathcal{M} \wedge \ldots \wedge x_n \in \mathcal{M} \rightarrow t_{\mathcal{M}} = t$$

(2) Let φ be a formula of $\mathcal{L}_{RST}$ such that $Fv(\varphi) = \{y_1, \ldots, y_k, x_1, \ldots, x_n\}$, and $\varphi \succ_{RST} \{y_1, \ldots, y_k\}$. Then for every $a_1, \ldots, a_n \in \mathcal{M}$, the class $\{\langle y_1, \ldots, y_k\rangle \mid \varphi(y_1, \ldots, y_k, a_1, \ldots, a_n)\}$ is a set, and it equals the class $\{\langle y_1, \ldots, y_k\rangle \in \mathcal{M}^k \mid \mathcal{M} \models \varphi(y_1, \ldots, y_k, a_1, \ldots, a_n)\}$

Note 8 RST can be shown to be equivalent to Gandy's basic set theory ([10]) and to the system called BST_0 in [15]. It is a very weak subsystem of ZF. Even for getting from it the system obtained from ZF by deleting the axioms of infinity and foundations, one should considerably strengthen the relation $\succ_{RST}$ (but then the resulting relation does not reflect d.i. anymore, and the terms do not always have absolute meaning). See [4] for further details.

On Predicative Set Theory. In his writings Gödel expressed the view that his hierarchy of constructible sets codified the predicatively acceptable means of set construction, and that the only impredicative aspect of the constructible universe L is its being based on the full class On of ordinals. This seems to us to be only partially true. We think that indeed the predicatively acceptable instances of the comprehension schema are those which determine the collections they define in an absolute way, independent of the extension of the "surrounding universe". Therefore a formula ψ is predicative (with respect to x) if the collection $\{x \mid \psi(x, y_1, \ldots, y_n)\}$ is completely and uniquely determined by the identity of the parameters $y_1, \ldots, y_n$, and the identity of other objects referred to (e.g. using constants) in ψ (all of which should be well-determined before). In other words: ψ is predicative (with respect to x) iff it is d.i. (with respect to x). It follows that all the operations used by Gödel are indeed predicatively acceptable, and even capture what is intuitively predicatively acceptable in the language of RST. However, we believe that one should go beyond first-order languages in order to capture at least the most obvious means of set construction which are predicatively acceptable. In [2] we suggest that an adequate language for this might again be obtained by adding to the language of RST an operation TC for transitive closure of *binary predicates.* [15] The idea is to replace $\succ_{RST}$ by a relation $\succ_{PZF}$, which like $\succ_{RST}$ is a syntactic approximation of an intuitive set-theoretical relation of "universe-independence", but this time only with respect to "universes" which contains the set ω of natural

[15] [15] makes some related steps. Thus it considers languages with an operation for forming the transitive closure of a given set or a given relation (when the latter is a set of pairs), or a language in which a predicate symbol $\in^*$, denoting the transitive closure of the special predicate $\in$, is added as an extra symbol to the language. However, none of these extensions forces one to go beyond $\mathcal{HF}$.

numbers (i.e. finite ordinals). $\succ_{PZF}$ is defined like $\succ_{RST}$, but with the following extra clause: $(TC_{x,y}\varphi)(x,y) \succ_{PZF} X$ if $\varphi \succ_{PZF} X$, and $\{x,y\} \cap X \neq \emptyset$. Thus the set ω of the finite ordinals is definable in this extended language by the term $\{y \mid \exists x.x = \emptyset \wedge (TC_{x,y} y = \{z \mid z = x \vee z \in x\})(x,y)\}$.

6 Some Related Work

There were of course plenty of works in the past on generalizing computability theory to more general types of structures. To the best of my knowledge, none of them has concentrated on domain independence as the fundamental notion. However, absoluteness does play an important role in the generalization of computability theory to arbitrary admissible structures. Other points of similarity between works in this area (see e.g. [7]) and the present one is the unification they both suggest between classical computability theory and constructibility in set theory, and the emphasis of both on relations (rather than functions) and on their formal definability. There is still a big difference between the two approaches in that admissible sets (or more generally, admissible structures) are based on Kripke-Platek set theory KP (or some variant of it, like KPU), and this theory is not predicatively justified. Indeed, one of its main principles is the impredicative Δ_0-collection schema. This schema is valid and constructive for $\mathcal{HF}$ and similar structures. However, it is not constructive anymore when infinite sets are allowed as first-class citizens (i.e. as elements of other sets). Moreover, in the general case the identity of the various sets that Δ_0-collection postulates is not always domain-independent. [16]

Another research which has even greater similarity with the present one is the work of Sazonov and his coauthors (see e.g. [15]) on "bounded set theory" (BST). The main points of similarity are the following:

- In BST too the emphasis is on computability with *sets*. In fact, $\mathcal{HF}$ is for BST the fundamental data structure for computability theory. Moreover, the problem of (effective) definability and computability of operations on sets is one of the main goals of this research program.
- Like in the present work, Gödel's constructibility theory provides a great part of the motivation and ideas, and according to both works the rudimentary operations are the most basic (effective) constructions on sets [17].
- BST is explicitly connected to database theory in general, and to query

[16] It is worth noting that KP can be obtained from RST by the addition of Δ_0-collection and (the predicatively acceptable) $\in$-induction.
[17] However, while the approach of the present paper naturally *leads* to the rudimentary operations, it seems to me that taking them as the most adequate starting point was somehow *assumed* by Sazonov.

languages in particular. In fact, this connection is one of the main possible applications suggested for it in [15] (and it is indeed investigated in other papers of Sazonov). It should be noted that BST can be used for query languages for semistructured databases, which are more general than the relational databases dealt with here (the present approach can in fact be extended to such structures, but doing this does not seem to contribute much to the specific goals of this paper).
- In order to provide an adequate treatment of effective set theories, both works introduce languages with complicated terms, including nested abstract set terms. What is more, the basic language of BST, $\Delta(BST_0)$, and our basic language $\mathcal{L}_{RST}$, are equivalent in their expressive power. [18]

Having reviewed the similarities between the work on BST and ours, let us turn to the differences:

- First of all, the goals of the two research programs seem to be different. BST is designed mainly to provide a "*theory of computability* (over sets) *with bounded resources*" ([15]). As far as I understand, the research on BST does not aim to get a generalized CT, nor does it have the much less ambitious specific goal of this paper: the unification of important notions developed in different areas of mathematics and computer science.
- As a result of the difference in goals, the concept of domain-independence (central to the present work) plays no real role in the research on BST (despite its close connections with database theory). Even the notion of absoluteness is connected to the research in BST only in a roundabout way, through the central role that Δ-formulas play in BST. Note however that unlike in our work, Δ-formulas are not taken in BST as a syntactic *approximation* of an important *semantic* notion (absoluteness), but as the obvious appropriate language for dealing with "bounded resources". [19]
- It was noted in [15] that all the constructs considered in the various languages of BST have a natural semantics in "any reasonable universe of sets V, say for Zermelo-Fraenkel set theory ZF". However, it was already emphasized above that none of these constructs can take us out of $\mathcal{HF}$. This is not something peculiar to the languages dealt with in [15], but (so I believe) is crucial to the whole approach. The fundamental notion of BST is "bounded" (operation, formula, resource, ...), and it is an essential feature of this notion that it cannot force us to get out of $\mathcal{HF}$ (Sazonov would almost certainly agree with this, but unlike me, he would take it as an important

[18] It is worth noting that while $\mathcal{L}_{RST}$ is based on just one uniform set-forming constructor (the abstraction $\{x \mid \varphi\}$), $\Delta(BST_0)$ is based on a mixture of complicated forms of abstraction (like $\{t(x) \mid x \in a \wedge \varphi\}$) and operation symbols (like $\bigcup$).

[19] I believe that this view might have prevented Sazonov and his coauthors from trying to extend the language of Δ-formulas in the way it is done here. Instead they have introduced into their languages by brute force various set-theoretic operations.

virtue of this notion and of his approach). In contrast, it was pointed out at the end of Section 5, that by using an appropriate extension of the language, the notion of domain-independence leads to the introduction (as an object) of the set of natural numbers, and to predicative (or countable) set theory.

References

[1] S. Abiteboul, R. Hull and V. Vianu, **Foundations of Databases**, Addison-Wesley, 1995.

[2] Avron A., *Transitive Closure and the mechanization of Mathematics*, In **Thirty Five Years of Automating Mathematics** (F. Kamareddine, ed.), 149-171, Kluwer Academic Publishers, 2003.

[3] A. Avron, *Safety Signatures for First-order Languages and Their Applications*, in **First-Order Logic revisited** (Hendricks et al, eds.), 37-58, Logos Verlag Berlin, 2004.

[4] Avron A., *Formalizing Set Theory as It Is Actually Used*, In **Proceedings of Mathematical Knowledge Management (MKM 2004)** (A. Asperti, G. Bancerek, and A. Trybulec, eds.), 32-43, LNCS 3119, Springer, 2004.

[5] K. J. Devlin, **Constructibility**, Perspectives in Mathematical Logic, Springer-Verlag, 1984.

[6] R.A. Di Paola, *The recursive unsolvability of the decision problem for the class of definite formulas*. J. ACM 16(2) (1969) 324-327.

[7] Y. L. Ershov, **Definability and Computability**, Siberian School of Algebra and Logic, Consultants Bureau, New-York, 1996.

[8] Feferman S., *Finitary Inductively Presented Logics*, In **Logic Colloquium 1988**, Amsterdam, North-Holland, 191-220, 1989. Reprinted in [9], 297-328.

[9] Gabbay D., editor, **What is a Logical System?** Oxford Science Publications, Clarendon Press, Oxford, 1994.

[10] Gandy, R. O., em Set-theoretic functions for elementary syntax, In **Axiomatic set theory, Part 2**, AMS, Providence, Rhode Island, 103-126, 1974.

[11] K. Gödel, **The Consistency of the Continuum Hypothesis**, Annals of Mathematical Studies, No. 3, Princeton University Press, Princeton, N.J., 1940.

[12] R. B. Jensen, *The Fine Structure of the Constructible Hierarchy*, Annals of Mathematical Logic 4, 229-308, 1972.

[13] M. Kiffer, *On Safety Domain independence and capturability of database queries*, Proc. International Conference on database and knowledge bases, 405-414, Jerusalem 1988.

[14] K. Kunen, **Set Theory, An Introduction to Independence Proofs**, North-Holland, 1980.

[15] V. Y. Sazonov, *On Bounded Set Theory*, Proceedings of the 10th International Congress on Logic, Methodology and Philosophy of Sciences, Florence, August 1995, in Volume I: Logic and Scientific Method, Kluwer Academic Publishers, 85-103, 1997.

[16] R. M. Smullyan, **The Incompleteness Theorems**, Oxford University Press, 1992.

[17] D. Suciu, *Domain-independent queries on databases with external functions*, Theoretical Computer Science 190, 279-315, 1998.

[18] J.D. Ullman, **Principles of database and knowledge-base systems**, Computer Science Press, 1988.