From Constructibility and Absoluteness to Computability and Domain Independence

Arnon Avron

School of Computer Science
Tel Aviv University, Tel Aviv 69978, Israel
aa@math.tau.ac.il

Abstract. Gödel's main contribution to set theory is his proof that GCH is consistent with ZFC (assuming that ZF is consistent). For this proof he has introduced the important ideas of constructibility of sets, and of absoluteness of formulas. In this paper we show how these two ideas of Gödel naturally lead to a simple unified framework for dealing with computability of functions and relations, domain independence of queries in relational databases, and predicative set theory.

1 Introduction: Absoluteness and Constructibility

Gödel classical work [6] on the constructible universe L is best known for its applications in pure set theory, especially consistency and independence proofs. Its relevance to computability theory was mostly ignored. Still, in this work Gödel introduced at least two ideas which are quite important from a computational point of view:

Computations with Sets The notion of computation is usually connected with discrete structures, like the natural numbers, or strings of symbols from some alphabet. In this respect [6] is important, first of all, in being the first comprehensive research on (essentially) computability within a completely different framework (technically, the name Gödel used was "constructibility" rather than "computability", but the difference is not really significant). No less important (as we shall see) is the particularly important data structure for which computability issues were investigated in [6]: sets. Specifically, for characterizing the "constructible sets" Gödel identified operations on sets (which we may call "computable"), that may be used for "effectively" constructing new sets from given ones (in the process of creating the universe of "constructible" sets). Thus, 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" constructions can be obtained through compositions.

Absoluteness 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 all structures M, in which v is legal). Absoluteness is a property of formulas which was crucial for Gödel consistency proof. However, it is not a decidable property. The following set Δ_0 of absolute formulas is therefore extensively used 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 $\varphi \wedge \psi$.
- If x and y are two different variables, and φ is in Δ_0, then so is $\exists x \in y\varphi$.

Now there is an obvious analogy between the roles in set theory of absolute formulas and of Δ_0 formulas, and the roles in formal arithmetic and computability theory of decidable formulas and of arithmetical Δ_0 formulas (i.e. Smullyan's "bounded" formulas). This analogy was noticed and exploited in the research on set theory. However, the reason for this analogy remains unclear, and beyond this analogy the importance and relevance of these two ideas of Gödel to other areas have not been noticed. As a result, strongly related ideas and theory have been redeveloped from scratch in relational database theory.

2 Domain Independence and Computability in Databases

From a logical point of view, a relational database DB of a scheme $\{P_1, \ldots, P_n\}$ is just a tuple $\langle \underline{P_1}, \ldots, \underline{P_n} \rangle$ of *finite* interpretations (called "tables") of the predicate symbols $P_1, \ldots, P_n$. DB can be turned into a structure S for a first-order language L with equality, the signature of which includes $\{P_1, \ldots, P_n\}$ and constants, by specifying a domain D, and an interpretation of the constants of L in it (different interpretations for different constants). 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 free variables, then the answer to ψ in S is the set of 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 $\{\emptyset\}$ and $\emptyset$, respectively). Now not every formula ψ of a L can serve as a query. Acceptable are only those the answer for which is a *computable function* of $\langle \underline{P_1}, \ldots, \underline{P_n} \rangle$ alone (and does not depend on the identity of the intended domain D. This in particular entails that the answer should be finite). Such queries are called *domain independent* ([8, 11, 1]). The exact definition is:

Definition 1. [1] *Let σ be a signature which includes $\overrightarrow{P} = \{P_1, \ldots, P_n\}$, and optionally constants and other predicate symbols (but no function symbols). A query $\varphi(x_1 \ldots, x_n)$ in σ is called $\overrightarrow{P}$*–d.i. ($\overrightarrow{P}$–domain-independent)*, if whenever S_1 and S_2 are structures for σ, S_1 is a substructure of S_2, and the interpretations of $\{P_1, \ldots, P_n\}$ in S_1 and S_2 are identical, then for all $a_1 \in S_2, \ldots, a_n \in S_2$:*

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

[1] This is a slight generalization of the definition in [Su98], which in turn is a generalization of the usual one ([Ki88,Ul88]). The latter applies only to free Herbrand structures which are generated by adding to σ some new set of constants.

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. (or "safe" according to any other reasonable notion of safety of queries, like "finite and computable"). 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 [11] (originally designed for languages with no function symbols) [2]:

1. $P_i(t_1, \ldots, t_{n_i}) \in SS(\overrightarrow{P})$ in case P_i (of arity n_i) is in $\overrightarrow{P}$.
2. $x = c$ and $c = x$ are in $SS(\overrightarrow{P})$ (where x is a variable and c is a constant).
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 $\varphi_i \in \{x = y, y = x\}$.

It should be noted that there is one clause in this definition which is somewhat strange: the last one, which treats conjunction. The reason why this clause does not simply tell us (like in the case of disjunction) when a conjunction of *two* formulas is in $SS(\overrightarrow{P})$, is the desire to take into account the fact that once the value of y (say) is known, the formula $x = y$ becomes domain independent. In the unified framework described in the next section this problematic clause is replaced by a more concise one (which at the same time is more general).

A more important fact is that given $\{\underline{P_1}, \ldots, \underline{P_n}\}$, the set of relations which are answers to some query in $SS(\overrightarrow{P})$ is exactly the closure of $\{\underline{P_1}, \ldots, \underline{P_n}\}$ under a finite set of basic operations called "the relational algebra" ([1, 11]). This set is quite similar to set of basic operations used by Gödel in [6] for constructing the constructible universe.

3 Partial Domain Independence and Absoluteness

There is an obvious similarity between the concepts of d.i. in databases, and absoluteness in Set Theory. However, the two notions are not identical. Thus, the formula $x = x$ is not d.i., although it is clearly absolute. To exploit the similarity,

[2] What we present below is both a generalization and a simplification of Ullman's original definition.

the formula *property* of d.i. was turned in [2] into the following *relation* between a formula φ and finite subsets of $Fv(\varphi)$:

Definition 2. *Let σ be like in Definition 1. A formula $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_k)$ in σ is $\overrightarrow{P}$−d.i. with respect to $\{x_1, \ldots, x_n\}$, if whenever S_1 and S_2 are structures as in Definition 1, then for all $a_1 \in S_2, \ldots, a_n \in S_2$ and $b_1 \in S_1, \ldots, b_k \in S_1$:*

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

Note that φ is d.i. iff it is d.i. with respect to $Fv(\varphi)$. On the other hand the formula $x = y$ is only partially d.i.: it is d.i. with respect to $\{x\}$ and $\{y\}$, but not with respect to $\{x, y\}$. Note also that a formula φ is d.i. with respect to $\emptyset$ if whenever S_1 and S_2 are structures as in Definition 1 then for all $b_1, \ldots, b_k \in S_1$ $S_2 \models \varphi(\overrightarrow{b}) \;\; \leftrightarrow \;\; S_1 \models \varphi(\overrightarrow{b})$. Under not very different conditions concerning S_1 and S_2, this is precisely Gödel's idea of absoluteness. We'll return to this below.

Another important observation is that given a domain S for the database, if $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_k)$ is $\overrightarrow{P}$−d.i. with respect to $\{x_1, \ldots, x_n\}$ then the function $\lambda y_1, \ldots, y_k.\{\langle x_1, \ldots, x_n\rangle \mid \varphi\}$ is a *computable* function from S^k to the set of finite subsets of S^n, the values of which depend only on the values of the arguments $y_1, \ldots, y_k$, but not on the identity of S. In case $n = 0$ the possible values of this function are $\{\langle\rangle\}$ and $\emptyset$, which can be taken as "true" and "false", respectively. Hence in this particular case what we get is a computable k-ary predicate on S. From this point of view k-ary predicates on a set S should be viewed as a special type of functions from S^k to the set of finite sets of S-tuples, rather than as a special type of functions from S^k to S, with arbitrary chosen two elements from S serving as the two classical truth values (while like in set theory, functions from S^k to S should be viewed as a special type of $(k+1)$-ary predicates on S).

Now it is easy to see that partial d.i. has the following properties (where $\varphi \succ X$ means that φ is $\overrightarrow{P}$−d.i. with respect to X):

0. If $\varphi \succ X$ and $Z \subseteq X$, then $\varphi \succ Z$.
1. $\varphi \succ Fv(\varphi)$ if φ is $p(t_1, \ldots, t_n)$ (where $p \in \overrightarrow{P}$).
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 for defining a syntactic approximation $\succ_P$ of the semantic $\overrightarrow{P}$-d.i. relation. It can easily be checked that the set $\{\varphi \mid \varphi \succ_P Fv(\varphi)\}$ 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!).

Note: For convenience, we are taking here $\wedge, \vee, \neg$ and $\exists$ as our primitives. Moreover: we take $\neg(\varphi \to \psi)$ as an abbreviation for $\varphi \wedge \neg\psi$, and $\forall x_1, \ldots, x_k\varphi$ as an abbreviation for $\neg\exists x_1, \ldots, x_k \neg\varphi$. This entails the following important property of "bounded quantification": *If $\succ$ is a relation satisfying the above properties, and $\varphi \succ \{x_1, \ldots, x_n\}$, while $\psi \succ \emptyset$, then $\exists x_1 \ldots x_n(\varphi \wedge \psi) \succ \emptyset$ and $\forall x_1 \ldots x_n(\varphi \rightarrow \psi) \succ \emptyset$* (recall that $\varphi \succ \emptyset$ is our counterpart of absoluteness).

4 Partial Domain Independence in Set Theory

We return now to set theory, to see how the idea of partial d.i. applies there. In order to fully exploit it, we use a language with abstraction terms *for sets.* However, we allow only terms which are known to be d.i. in a sense we now explain. For simplicity of presentation, we assume the accumulative universe V of ZF, and formulate our definitions accordingly.

Definition 3. *Let $\mathcal{M}$ be a transitive class. Define the relativization to $\mathcal{M}$ of terms and formulas recursively as follows:*

- $t_{\mathcal{M}} = t$ *if t is a variable or a constant.*
- $\{x \mid \varphi\}_{\mathcal{M}} = \{x \mid x \in \mathcal{M} \wedge \varphi_{\mathcal{M}}\}$.
- $(t = s)_{\mathcal{M}} = (t_{\mathcal{M}} = s_{\mathcal{M}}) \quad (t \in s)_{\mathcal{M}} = (t_{\mathcal{M}} \in s_{\mathcal{M}})$.
- $(\neg\varphi)_{\mathcal{M}} = \neg\varphi_{\mathcal{M}} \quad (\varphi \vee \psi)_{\mathcal{M}} = \varphi_{\mathcal{M}} \vee \psi_{\mathcal{M}}. \quad (\varphi \wedge \psi)_{\mathcal{M}} = \varphi_{\mathcal{M}} \wedge \psi_{\mathcal{M}}$.
- $(\exists x\varphi)_{\mathcal{M}} = \exists x(x \in \mathcal{M} \wedge \varphi_{\mathcal{M}})$.

Definition 4. *Let T be a theory such that $V \models T$.*

1. *Let t be a term, and let $Fv(t) = \{y_1, \ldots, y_n\}$. We say that t is T-*d.i.*, if the following is true (in V) for every transitive model $\mathcal{M}$ of T:*

$$\forall y_1 \ldots \forall y_n . y_1 \in \mathcal{M} \wedge \ldots \wedge y_n \in \mathcal{M} \rightarrow t_{\mathcal{M}} = t$$

2. *Let φ be a formula, and let $Fv(\varphi) = \{y_1, \ldots, y_n, x_1, \ldots, x_k\}$. We say that φ is T-*d.i. for* $\{x_1, \ldots, x_k\}$ if $\{\langle x_1, \ldots, x_k\rangle \mid \varphi\}$ is a set for all values of the parameters $y_1, \ldots, y_n$, and the following is true (in V) for every transitive model $\mathcal{M}$ of T:*

$$\forall y_1 \ldots \forall y_n . y_1 \in \mathcal{M} \wedge \ldots \wedge y_n \in \mathcal{M} \rightarrow [\varphi \leftrightarrow (x_1 \in \mathcal{M} \wedge \ldots \wedge x_k \in \mathcal{M} \wedge \varphi_{\mathcal{M}})]$$

Thus, a term is T-d.i. if it has the same interpretation in all transitive models of T which contains the values of its parameters, while a formula is T-d.i. for $\{x_1, \ldots, x_k\}$ if it has the same extension (which should be a set) in all transitive models of T which contains the values of its other parameters. In particular: φ is T-d.i. for $\emptyset$ iff it is absolute relative to T in the original sense of set theory, while φ is T-d.i. for $Fv(\varphi)$ iff it is domain-independent in the sense of database theory (see Definition 1) for transitive models of T.

The set-theoretical notion of d.i., we have just introduced, is again a semantic notion that one cannot characterize in a constructive manner, and so a syntactic approximation of it should be used in practice. The key observation for this is that the transitive classes are the structures for which the atomic formula $x \in y$ (where y is different from x) is d.i. with respect to $\{x\}$. Accordingly, an appropriate approximation is most naturally obtained by adapting the definition of $\succ_P$ above to the present language, taking into account this key observation:

Definition 5. *The relation $\succ_{RST}$ is inductively defined as follows:*

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* $\varphi \succ_{RST} X$.

Note: It can easily be proved by induction on the complexity of formulas that the clause 0 in the definition of $\succ_P$ is also satisfied by $\succ_{RST}$: if $\varphi \succ_{RST} X$ and $Z \subseteq X$, then $\varphi \succ_{RST} Z$.

A first (and perhaps the most important) use of $\succ_{RST}$ is for defining the set of legal terms of the corresponding system RST (Rudimentary Set Theory). Unlike the languages for databases (in which the only terms were variables and constants), the language of RST used here has a very extensive set of terms. It is inductively defined as follows:

- Every variable is a term.
- If x is a variable, and φ is a formula such that $\varphi \succ_{RST} \{x\}$, then $\{x \mid \varphi\}$ is a term (and $Fv(\{x \mid \varphi\}) = Fv(\varphi) - \{x\}$).

(Actually, the relation $\succ_{RST}$, the set of terms of RST, and the set of formulas of RST are defined together by a simultaneous induction).

A second use of $\succ_{RST}$ is that the set $\{\varphi \mid \varphi \succ_{RST} \emptyset\}$ is a natural extension of the set Δ_0 of bounded formulas. Moreover, we have:

Theorem 1. *Let RST be the theory consisting of the following axioms:*

Extensionality: $\forall y(y = \{x \mid x \in y\})$
Comprehension: $\forall x(x \in \{x \mid \varphi\} \leftrightarrow \varphi)$

Then given an extension T of RST, any valid term t of RST is T-d.i., and if $\varphi \succ_{RST} X$, then φ is T-d.i. for X.

The following theorem connects $\succ_{RST}$ with the class of rudimentary set functions (introduced independently by Gandy ([5]) and Jensen ([7]). See also [4]) — a refined version of Gödel basic set functions:

Theorem 2.

1. *If F is an n-ary rudimentary function, then there exists a formula φ s. t.:*
   (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 1. *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.*

4.1 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, independently of any "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 in the formula (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 all the predicatively acceptable means of set construction. In [3] we suggest that an adequate language for this is obtained by adding to the the language of RST an operation TC for transitive closure of binary relations, and then replacing $\succ_{RST}$ by the relation $\succ_{PZF}$, which 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 $\{x, y\} \cap X \neq \emptyset$. See [3] for more details.

5 Domain Independence: a General Framework

In this section we introduce a general abstract framework for studying domain independence and absoluteness (originally introduced in [2]).

Definition 6. *A* d.i.-signature *is a pair (σ, F), where σ is an ordinary first-order signature, and F is a function which assigns to every n-ary symbol s from σ (other than equality) a subset of $\mathcal{P}(\{1, \ldots, n\})$.*

Definition 7. *Let (σ, F) be a d.i.-signature. Let S_1 and S_2 be two structures for σ s.t. $S_1 \subseteq S_2$. S_2 is called a $(\sigma, F)-$*extension *of S_1 if the following conditions are satisfied:*

- *If $p \in \sigma$ is a predicate symbol of arity n, $I \in F(p)$, and $a_1, \ldots, a_n$ are elements of S_2 such that $a_i \in S_1$ in case $i \notin I$, then $S_2 \models p(a_1, \ldots, a_n)$ iff $a_i \in S_1$ for* all *i, and $S_1 \models p(a_1, \ldots, a_n)$.*
- *If $f \in \sigma$ is a function symbol of arity n, $a_1, \ldots, a_n \in S_1$, and b is the value of $f(a_1, \ldots, a_n)$ in S_2, then $b \in S_1$, and b is the value of $f(a_1, \ldots, a_n)$ in S_1. Moreover: if $I \in F(f)$, and $a_1, \ldots, a_n$ are elements of S_2 such that $a_i \in S_1$ in case $i \notin I$, then $S_2 \models b = f(a_1, \ldots, a_n)$ iff $a_i \in S_1$ for* all *i, and $S_1 \models b = f(a_1, \ldots, a_n)$.*

Definition 8. *Let (σ, F) be as in Definition 7. A formula φ of σ is called $(\sigma, F)-$*d.i. w.r.t. *X ($\varphi \succ^{di}_{(\sigma,F)} X$) if whenever S_2 is a $(\sigma, F)-$extension of S_1,* *and φ^* results from φ by substituting values from S_1 for the free variables of φ that are not in X, then the sets of tuples which satisfy φ^* in S_1 and in S_2 are identical.* [3] *A formula φ of σ is called $(\sigma, F)-$*d.i. *if $\varphi \succ^{di}_{(\sigma,F)} Fv(\varphi)$, and $(\sigma, F)-$*absolute *if $\varphi \succ^{di}_{(\sigma,F)} \emptyset$.*

Note. 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 defined $F(=) = \{\{1\}, \{2\}\}$ (meaning that $x_1 = x_2$ is d.i. w.r.t. both $\{x_1\}$ and $\{x_2\}$, but not w.r.t. $\{x_1, x_2\}$).

Examples

- Let σ be a signature which includes $\overrightarrow{P} = \{P_1, \ldots, P_n\}$, and optionally constants and other predicate symbols (but no function symbols). Assume that the arity of P_i is n_i, and define $F(P_i) = \{\{1, \ldots, n_i\}\}$. 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 2.
- Let $\sigma_{ZF} = \{\in\}$ and let $F_{ZF}(\in) = \{\{1\}\}$. In this case the universe V is a $(\sigma_{ZF}, F_{ZF})-$ extension of the transitive sets and classes. Therefore a formula is σ_{ZF}-absolute iff it is absolute in the usual sense of set theory.

Again the relation of $(\sigma, F)-$d.i. is a semantic notion that in practice should be replaced by a syntactic approximation. The following definition generalizes in a very natural way the relations $\succ_P$ and $\succ_{RST}$:

Definition 9. *The relation $\succ_{(\sigma,F)}$ is inductively defined as follows:*

0. *If $\varphi \succ_{(\sigma,F)} X$ and $Z \subseteq X$, then $\varphi \succ_{(\sigma,F)} Z$.*

1*a*. *If p is an n-ary predicate symbol of σ; $x_1, \ldots, x_n$ are n distinct variables, and $\{i_1, \ldots, i_k\}$ is in $F(p)$, then $p(x_1, \ldots, x_n) \succ_{(\sigma,F)} \{x_{i_1}, \ldots, x_{i_k}\}$.*

1*b*. *If f is an n-ary function symbol of σ; $y, x_1, \ldots, x_n$ are $n+1$ distinct variables, and $\{i_1, \ldots, i_k\} \in F(f)$, then $y = f(x_1, \ldots, x_n) \succ_{(\sigma,F)} \{x_{i_1}, \ldots, x_{i_k}\}$.*

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

Again it is easy to see that if $\varphi \succ_{(\sigma,F)} X$, then $\varphi \succ^{di}_{(\sigma,F)} X$. The converse fails, of course. However, we suggest the following conjecture (that for reasons to become clear in the next section, may be viewed as a generalized Church Thesis):

Conjecture. Given a d.i. signature (σ, F), a formula is upward (σ, F)-absolute iff it is logically equivalent to a formula of the $\exists y_1, \ldots, y_n \psi$, where $\psi \succ_{(\sigma,F)} \emptyset$ ($\varphi(x_1, \ldots, x_n)$ is upward (σ, F)-absolute if whenever S_2 is a $(\sigma, F)-$extension of S_1, and $\models_{S_1} \varphi(a_1, \ldots, a_n)$, then $\models_{S_2} \varphi(a_1, \ldots, a_n)$).

[3] φ^* is a formula only in a generalized sense, but the intention should be clear.

6 Absoluteness and Computability in $\mathcal{N}$

Finally, we turn to the connections between the above ideas and computability in the Natural numbers.

Definition 10. *The d.i. signature $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$ is defined as follows:*

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

Definition 11. *The standard structure $\mathcal{N}$ for $\sigma_{\mathcal{N}}$ has the set of natural numbers as its domain, with the usual interpretations of 0, 1, 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 now 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 $\mathcal{N}$). Accordingly, if $\varphi \succ_{(\sigma_{\mathcal{N}}, F_{\mathcal{N}})} \emptyset$, then for any assignment in N it gets the same truth value in all initial segments of $\mathcal{N}$ (including $\mathcal{N}$ itself) which contain the values assigned to its free variables. Now the set of formulas φ such that $\varphi \succ_{(\sigma_{\mathcal{N}}, F_{\mathcal{N}})} \emptyset$ is a straightforward extension of Smullyan's set of bounded formulas ([10]). This set is defined of course using the relation $\succ_N = \succ_{(\sigma_{\mathcal{N}}, F_{\mathcal{N}})}$. From definitions 9 and 10 it easily follows that this relation can be characterized as follows (compare with Definition 5!):

1. $\varphi \succ_N \emptyset$ if φ is atomic.
2. $\varphi \succ_N \{x\}$ if $\varphi \in \{x \neq x, x = t, t = x, x < t\}$, and $x \notin Fv(t)$.
3. $\neg\varphi \succ_N \emptyset$ if $\varphi \succ_N \emptyset$.
4. $\varphi \vee \psi \succ_N X$ if $\varphi \succ_N X$ and $\psi \succ_N X$.
5. $\varphi \wedge \psi \succ_N X \cup Y$ if $\varphi \succ_N X$, $\psi \succ_N Y$, and $Y \cap Fv(\varphi) = \emptyset$.
6. $\exists y \varphi \succ_N X - \{y\}$ if $y \in X$ and $\varphi \succ_N X$.

Now the crucial connection between Gödel's work on absoluteness in set theory, and computability in the natural numbers, is given in the following Theorem:

Theorem 3. *The following conditions are equivalent for a relation R on N:*

1. *R is semi-decidable.*
2. *R is definable by a formula of the form $\exists y_1, \dots, y_n \psi$, where $\psi \succ_N \emptyset$.*
3. *R is definable by a formula of the form $\exists y_1, \dots, y_n \psi$, where the formula ψ is $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$-absolute.*

Proof. 2. follows from 1. by the Thesis of Church and Smullyan's characterization in [10] of the r.e. subsets of N using his set of bounded formulas (recall that if ψ is bounded, then $\psi \succ_N \emptyset$). That 3. follows from 2. is immediate from the fact that if $\psi \succ_N \emptyset$, then ψ is $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$-absolute. To show that 3. entails 1., assume that R is definable by a formula of the form $\exists y_1, \dots, y_n \psi$, where the formula $\psi(x_1, \ldots, x_k, y_1, \ldots, y_n)$ is $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$-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 $(\sigma_{\mathcal{N}}, F_{\mathcal{N}})$-absolute, it easily follows that this procedure halts with the correct answer in case $R(n_1, \ldots, n_k)$, and never halt otherwise.

The last theorem shows a very close connection between (semi)-computability and (upward) absoluteness. However, further research is needed in order to understand the full connection between these notions. A key problem that one has to solve in order to provide a general computability theory based on d.i. relations and absoluteness, is what is so special about the standard interpretations in $\mathcal{N}$ of P_+ and $P_\times$ that makes the last theorem possible. We suspect that in order to provide a satisfactory answer (and develop the desired theory), one should go beyond first-order languages (most probably to first-order language with a transitive closure operation). We leave that for future investigations.

References

1. S. Abiteboul, R. Hull and V. Vianu, **Foundations of Databases**, Addison-Wesley, 1995.
2. 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.
3. A. Avron, *A New Approach to Predicative Set Theory*, to appear.
4. K. J. Devlin, **Constructibility**, Perspectives in Mathematical Logic, Springer-Verlag, 1984.
5. Gandy, R. O., em Set-theoretic functions for elementary syntax, In **Axiomatic set theory, Part 2**, AMS, Providence, Rhode Island, 1974, 103-126.
6. K. Gödel, **The Consistency of the Continuum Hypothesis**, Annals of Mathematical Studies, No. 3, Princeton University Press, Princeton, N.J., 1940.
7. R. B. Jensen, *The Fine Structure of the Constructible Hierarchy*, Annals of Mathematical Logic 4 (1971), 229-308, AMS, Providence, Rhode Island, 1974, 143-176.
8. M. Kiffer, *On Safety Domain independence and capturability of database queries*, Proc. International Conference on database and knowledge bases, 405-414, Jerusalem 1988.
9. K. Kunen, **Set Theory, An Introduction to Independence Proofs**, North-Holland, 1980.
10. R. M. Smullyan, **The Incompleteness Theorems**, Oxford University Press, 1992.
11. J.D. Ullman, **Principles of database and knowledge-base systems**, Computer Science Press, 1988.