Acta Cybernetica, Vol. 11, No. 1-2, Szeged, 1993

On the characterization of the integers: The hidden function problem revisited

R. Berghammer*

Abstract

In this paper the hidden function problem studied so far only for equational (e.g., in [9] and [11]) or conditional equational (e.g., in [3]) algebraic specifications is considered for arbitrary first-order theories. It is shown that a unique characterization of the integers with zero, successor and predecessor as term-generated model of a finite first-order theory needs at least one hidden function or relation.
Keywords: Hidden function problem, algebraic specifications, first-order theories.

1 Introduction

In mathematical logic, a structure for a first-order language is said to be a model for a set T of sentences over the same language, if each sentence of T holds in it. The algebraic specification approach of computer science uses a restricted definition. Here it is often additionally demanded that each element of the carrier sets can finitely be "described" by a closed term, i.e., that the model of the specification is term-generated (see e.g., [1], [12], [6], or [13]). The main reason for the restriction to term-generated models of specifications is the necessity of finite descriptions of algorithms. As an essential advantage one obtains the proof principle of term induction. Furthermore, by using only term-generated models one is able to extend the expressiveness of first-order theories (resp. algebraic specifications).

In this paper we deal with the question, whether and how the structure $Z := (\mathbb{Z}, 0, \text{succ}, \text{pred})$ can – up to isomorphism – be characterized as the only term-generated model of a set of first-order sentences over a first-order language with symbols for 0, the successor function $\text{succ}(u) := u + 1$, and the predecessor function $\text{pred}(u) := u - 1$. First, we give a positive answer using an infinite set of sentences. Then we show, and this is the main result of the paper, that there is no finite set of first-order sentences with the same property. Finally, we extend the language by a symbol for the "usual" ordering relation on the integers $\mathbb{Z}$ and present a finite set of sentences, which has the structure $\overline{Z} := (\mathbb{Z}, 0, \text{succ}, \text{pred}, \leq)$ as – up to isomorphism – only term-generated model.

The relation $\leq$ simplifies the specification of the constant and operations of interest 0, succ, and pred. In the terminology of algebraic specifications it is called

*Fakultät für Informatik, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, D-85579 Neubiberg

a "hidden function", since the way to specify Z is structured by first specifying $\overline{Z}$ and then to forget or hide the auxiliary relation $\leq$.

Strictly speaking, $\leq$ is a hidden relation. The term "hidden function" (which we will use in the remainder of the paper, too) results from the fact that the algebraic specification approach considers relations as functions to the truth values.

Given a class C of first-order formulae and a semantic mechanism S which determines the meaning of a specification, the so-called hidden function problem for C and S asks whether the use of hidden functions extends the expressiveness of specifications. All the known examples deal with the following question: Is there a structure that fails to possess a unique characterization (this notion depends on S) using finite subsets of C only, but the same is not true if auxiliary functions may be used? In the case of C being the class of universally quantified equations and S being initial algebra semantics, a solution – the first example which requires the use of a hidden function – can be found in [9]. This paper contains no formal proof, but based on Majster's example in [11] a simple structure, called "toy stack", is constructed and carefully proved that it cannot be specified using initial algebra semantics and finitely many equations unless hidden functions are permitted. This proof is mainly based on regular sets and their properties. Independently of [11], in [2] another solution of the hidden function problem for equational specifications and initial algebra semantics is given. It is shown that the structure $N := (I\!N, 0, \text{succ}, \text{sqr})$, where succ is again the successor function and $\text{sqr}(u) := u^2$, does not possess a finitary equational specification without the use of hidden machinery. The (rather complicated) proof can also be found in [3]. Obviously, N admits a very natural finite equational specification involving addition and multiplication as auxiliary functions. Using the so-called sparsity property of predicates on natural numbers, in the same paper [3] the hidden function problem is also solved for conditional equational specifications and initial algebra semantics.

Our examples Z and $\overline{Z}$ solve also a hidden function problem for certain C and S. In comparison to the papers just mentioned, we do not restrict the class of formulae and consider all term-generated models. This means that C is the class of all first-order formulae and that a structure M is (uniquely) characterized by a set T of sentences under S if and only if M is a term-generated model of T and all these models are isomorphic. Furthermore, we use proof principles from "classical" model theory, viz. the use of the compactness theorem and elimination of quantifiers.

2 Preliminaries

Throughout this paper we use first-order logic with the equality symbol $\approx$ as a logical symbol. In this section, we briefly recall some basic definitions of first-order logic. Further details can be found in, for instance, [7] or [10].

Assume L to be a first-order language. A *structure* M for L (also called L-structure) consists of a non-empty carrier set $|M|$, an n-ary function $f_M : |M|^n \to |M|$ for every n-place function symbol f, and an n-ary predicate $p_M : |M|^n \to I\!B$ for every n-place predicate symbol p, where $I\!B$ denotes the set $\{0, 1\}$ representing truth values. If $n = 0$, then f_M is an element of $|M|$ and p_M is a truth value.

Assume M and N to be two structures for the same first-order language. A bijective function $\Phi : |M| \to |N|$ is said to be an *isomorphism* from M to N, if

$$\Phi(f_M(u_1,\ldots,u_n)) = f_N(\Phi(u_1),\ldots,\Phi(u_n))$$

for all n-place function symbols f and all $u_1,\ldots,u_n \in |M|$ and

$$p_M(u_1,\ldots,u_n) = 1 \;\Leftrightarrow\; p_N(\Phi(u_1),\ldots,\Phi(u_n)) = 1$$

for all n-place predicate symbols p and all $u_1,\ldots,u_n \in |M|$. If there is an isomorphism from M to N, then we say that M and N are *isomorphic*.

Let M be a structure for a first-order language L and $\Psi : V \to |M|$ be an assignment for the variables $x \in V$ with values from $|M|$. Furthermore, let t be a term and A be a formula built up over L. By t^M_Ψ we denote the value of t in M under Ψ; by $M \models A[\Psi]$ we denote that A holds in M under Ψ. Both notations are inductively defined as usual. In particular, we have $M \models t_1 \approx t_2[\Psi]$ if and only if $t^M_{1\Psi} = t^M_{2\Psi}$. If both t and A are closed, then t^M_Ψ as well as $M \models A[\Psi]$ do not depend on the assignment Ψ. Therefore, in this case we use the notations t^M and $M \models A$ instead. The notation $M \models A$ is also used to indicate that $M \models A[\Psi]$ for every assignment Ψ.

Let L be a first-order language. A set T of sentences (i.e., closed formulae) built up over L is called a *theory* over L. A structure M for L is said to be a *model* of T, if $M \models A$ for all sentences $A \in T$. In addition, M is called *term-generated,* if for every element $u \in |M|$ there exists a closed term t (also built up over L) such that $u = t^M$.

3 An infinite characterization of the integers without hidden functions

In the following, we give a characterization of the integers with 0, succ, and pred as – up to isomorphism – only term-generated model of an infinite first-order theory. This result will also be used in the next section.

Let L_Z be the first-order language consisting of a 0-place function symbol (constant symbol) z and two 1-place function symbols s, p, and let T_Z denote the following infinite theory:

$$\begin{array}{ll}
(1) & \forall x(s(p(x)) \approx x) \\
(2) & \forall x(p(s(x)) \approx x) \\
(3.1) & \forall x(\neg(s(x) \approx x)) \\
(3.2) & \forall x(\neg(s(s(x)) \approx x)) \\
 & \cdots\cdots \\
(3.n) & \forall x(\neg(s(s(\ldots s(s(x))\ldots)) \approx x) \quad \text{(exactly } n \text{ occurrences of } s\text{)}. \\
 & \cdots\cdots
\end{array}$$

Obviously, we have: The structure $Z := (\mathbb{Z}, 0, \text{succ}, \text{pred})$ is a term-generated model of T_Z. We call Z the *standard model* of the theory T_Z. In the following, we show that it is – up to isomorphism – the only term-generated model of T_Z. To this end, we assume for the rest of this Section 3 an arbitrarily chosen (but fixed) term-generated model $M := (|M|, z_M, s_M, p_M)$ of T_Z and construct an isomorphism from M to the standard model.

Define s_M^n (resp. p_M^n) as n^{th} power of s_M (resp. p_M). Fundamental for the construction of the just mentioned isomorphism is the following representation of the elements of $|M|$.

Lemma 3.1 *Let $u \in |M|$. Then there exists exactly one natural number $n \in \mathbb{N}$ such that $u = s_M^n(z_M)$ or $u = p_M^n(z_M)$.*

Proof. a) In the first step we prove the existence of the number n.

As the model M is term-generated, for all $u \in |M|$ there exists a closed term t built up over L_Z such that $t^M = u$. Thus, it suffices to show that for all closed terms t built up over L_Z there exists a natural number $n \in \mathbb{N}$ such that $t^M = s_M^n(z_M)$ or $t^M = p_M^n(z_M)$. This can be done by term induction.

Induction base: The case of t being the symbol z is trivial; choose $n = 0$.

Induction step: By the induction hypothesis, $t^M = s_M^n(z_M)$ or $t^M = p_M^n(z_M)$. First, suppose $t^M = s_M^n(z_M)$. Then we have

$$s(t)^M = s_M(t^M) = s_M(s_M^n(z_M)) = s_M^{n+1}(z_M).$$

Furthermore, due to the validity of sentence (2) in M,

$$p(t)^M = p_M(t^M) = p_M(s_M^n(z_M)) = p_M(s_M(s_M^{n-1}(z_M))) = s_M^{n-1}(z_M),$$

provided $n > 0$. Finally, in the case $n = 0$ we obtain

$$p(t)^M = p_M(t^M) = p_M(z_M).$$

This shows that also $s(t)^M$ and $p(t)^M$ have the stated representation.

The remaining case $t^M = p_M^n(z_M)$ is handled similarly using the validity of (1) in M.

b) In a second step, now we prove the uniqueness of the representation. To this end, suppose $u = s_M^m(z_M) = s_M^n(z_M)$ and $m \neq n$. W.l.o.g, let $m < n$. Then there exists a positive natural number k fulfilling the equation $m + k = n$. Sentence (2) is true in M. Thus,

$$s_M^m(z_M) = s_M^n(z_M) = s_M^m(s_M^k(z_M)) \Leftrightarrow z_M = s_M^k(z_M).$$

However, $s_M^k(z_M) = z_M$ contradicts the validity of sentence (3.k) in M. In the same manner one deals with the remaining cases. □

With the help of this lemma, we are able to define a function Φ from the carrier set $|M|$ to the integers by

$$\Phi : |M| \to \mathbb{Z} \qquad \Phi(u) := \begin{cases} n & \text{if } u = s_M^n(z_M) \\ -n & \text{if } u = p_M^n(z_M). \end{cases}$$

We then have the following property:

Lemma 3.2 *The function Φ is an isomorphism from the fixed model M to the standard model Z.*

Proof. Bijectivity of Φ is obvious; the inverse Φ^{-1} from the integers to $|M|$ is given as

$$\Phi^{-1} : \mathbb{Z} \to |M| \qquad \Phi^{-1}(n) := \begin{cases} s_M^n(z_M) & \text{if } n \geq 0 \\ p_M^n(z_M) & \text{if } n \leq 0. \end{cases}$$

It remains to prove that Φ preserves the interpretations of the three symbols z, s, and p. This is done in the following. Note, that we have $z_Z = 0, s_Z = \text{succ}$, and $p_Z = \text{pred}$.

Obviously, $\Phi(z_M) = 0$ holds. Now, assume $u \in |M|$. For a proof of $\Phi(s_M(u)) = \text{succ}(\Phi(u))$ we distinguish two cases. If $u = s_M^n(z_M)$, then we obtain

$$\Phi(s_M(u)) = \Phi(s_M^{n+1}(z_M)) = n + 1 = \Phi(s_M^n(z_M)) + 1 = \Phi(u) + 1 = \text{succ}(\Phi(u)).$$

In the case $u = p_M^n(z_M)$ we have

$$\Phi(s_M(u)) = \Phi(p_M^{n-1}(z_M)) = -n + 1 = \Phi(p_M^n(z_M)) + 1 = \Phi(u) + 1 = \text{succ}(\Phi(u)),$$

provided $n > 0$ (here we have used that sentence (1) is true in M), and

$$\Phi(s_M(u)) = \Phi(s_M(z_M)) = 1 = \Phi(z_M) + 1 = \Phi(u) + 1 = \text{succ}(\Phi(u)),$$

provided $n = 0$. Equation $\Phi(p_M(u)) = \text{pred}(\Phi(u))$ is proved analogously to the latter one. □

Summing up, we have the desired result that the structure Z is characterized by the theory T_Z:

Theorem 3.3 *The standard model Z is - up to isomorphism - the only term-generated model of T_Z.* □

4 There is no finite characterization of the integers without hidden functions

In this section we show (Theorem 4.3 below) that there is no finite theory of arbitrary sentences built up over the language L_Z of Section 3 which has Z as - up to isomorphism - only term-generated model. The crucial point of this proof is the use of the compactness theorem of first-order logic which implies that a theory T has a model if every finite subset of T has a model. However, to conclude the proof it is additionally necessary to get a term-generated model for the chosen theory. Here elimination of quantifiers plays an important role.

A theory T over a first-order language L admits *elimination of quantifiers* if and only if for every formula A built up over L there is a quantifier-free formula B built up over the same language such that $M \models A \leftrightarrow B$ for every model M of T. In model theory elimination of quantifiers is one of the methods for proving theories decidable. Some examples can e.g., be found in [10], Section 13. The next lemma shows that the theory T_Z of Section 3 admits elimination of quantifiers, whereby no additional free variables are introduced.

Lemma 4.1 *Assume A to be a formula built up over the language L_Z. Then there exists a quantifier-free formula B, also built up over L_Z, such that $M \models A \leftrightarrow B$ for every model M of T_Z and, furthermore, the set of the free variables of B is contained in the set of the free variables of A.*

Proof. a) In a first step we prove the existence of a quantifier-free formula B over L_Z such that $M \models A \leftrightarrow B$ for every model M of T_Z.

We are allowed to assume the given formula A to be of the form $\exists x(A_1 \wedge \ldots \wedge A_m)$, where each $A_i, 1 \leq i \leq m$, is an atomic formula or the negation of an atomic formula. A proof of this well-known fact can e.g., be found in [7], Section 3.1. Furthermore, we may suppose that the variable x occurs in each A_i. For, if x does not occur in some A_{i_0}, then we use the equivalence of $\exists x(A_1 \wedge \ldots \wedge A_m)$ and $A_{i_0} \wedge \exists x(A_1 \wedge \ldots \wedge A_{i_0-1} \wedge A_{i_0+1} \wedge \ldots \wedge A_m)$.

Assume $y_i, 1 \leq i \leq k$, to denote the free variables of $\exists x(A_1 \wedge \ldots \wedge A_m)$. For a being an element from $\{z, x, y_1, \ldots, y_k\}$, we abbreviate the term $s(\ldots s(a)\ldots)$ (resp. $p(\ldots p(a)\ldots)$) with $n \geq 0$ occurrences of s (resp. p) by $s^n(a)$ (resp. $p^n(a)$). Particularly, we have $s^0(a) := p^0(a) := a$.

Now, suppose M to be a model of the theory T_Z. Each atomic sub-formula of A is an equation $t_1 \approx t_2$, where the terms $t_i, 1 \leq i \leq 2$, are built up using the variables $y_i, 1 \leq i \leq k$, the variable x, and the function symbols z, s, and p. Since the variable x occurs in at least one of the terms and the sentences (1) and (2) are true in M, there exist natural numbers m and n and $a \in \{z, x, y_1, \ldots, y_k\}$ such that $t_1 \approx t_2$ is equivalent to one of the following equations:

$$\begin{array}{llll} \text{(i)} & s^m(x) \approx s^n(a) & \text{(ii)} & s^m(x) \approx p^n(a) \\ \text{(iii)} & p^m(x) \approx s^n(a) & \text{(iv)} & p^m(x) \approx p^n(a). \end{array}$$

In the case $m \leq n$, the first equation is equivalent to $x \approx s^{n-m}(a)$; otherwise it is equivalent to $s^{m-n}(x) \approx a$, i.e., to $x \approx p^{m-n}(a)$. The proofs that also for the remaining equations there exist equivalent formulae of this specific form are identical and follow likewise from the validity of (1) and (2) in M.

Hence, we may suppose that every atomic formula occurring in A is of the form $x \approx s^n(a)$ or $x \approx p^n(a)$, where $a \in \{z, x, y_1, \ldots, y_k\}$. However, we may further suppose that a is different from x. This is due to the fact that $x \approx s^n(x)$ as well as $x \approx p^n(x)$ can be replaced by $z \approx z$ if $n = 0$, and by $\neg(z \approx z)$ if $n \neq 0$, and that the latter closed formulae can again be moved out-side of quantification.

Summing up, we may assume the given formula A to be of the form $(1 \leq m, 1 \leq j \leq m)$

$$\exists x(x \approx t_1 \wedge \ldots \wedge x \approx t_{j-1} \wedge \neg(x \approx t_j) \wedge \ldots \wedge \neg(x \approx t_m)),$$

where the terms $t_i, 1 \leq i \leq m$, are of the form $s^n(a)$ or $p^n(a)$ and $a \in \{z, y_1, \ldots, y_k\}$. Now, we distinguish three cases:

Case 1: $j = 1$, i.e, the formula A has the form $\exists x(\neg(x \approx t_1) \wedge \ldots \wedge \neg(x \approx t_m))$. It can easily be shown that the carrier set of each model of the theory T_Z is infinite. Now

$$\begin{array}{rcl} M \text{ is a model of } T_Z & \Rightarrow & |M| \text{ is infinite} \\ & \Rightarrow & M \models \forall y_1 \ldots \forall y_m(\exists x(\neg(x \approx y_1) \wedge \ldots \wedge \neg(x \approx y_m))) \\ & \Rightarrow & M \models \exists x(\neg(x \approx t_1) \wedge \ldots \wedge \neg(x \approx t_m)) \end{array}$$

implies that A is true in M. Since $M \models z \approx z$ holds, too, we may choose B as $z \approx z$ and obtain, thus, $M \models A \leftrightarrow B$.

Case 2: $j > 1$ and $m = 1$, i.e., A has the form $\exists x(x \approx t_1)$. Then A is also valid in M and we may again choose B as formula $z \approx z$.

Case 3: $j > 1$ and $m \geq 2$, i.e., A contains an equation and there is at last a further equation and/or negation of an equation:

$$\exists x(x \approx t_1 \wedge \ldots \wedge x \approx t_{j-1} \wedge \neg(x \approx t_j) \wedge \ldots \wedge \neg(x \approx t_m)).$$

In this case, first, we delete the equation $x \approx t_1$ from A and then replace in the resulting formula every occurrence of the variable x by the term t_1. Since x does not occur in the terms t_i, $1 \leq i \leq m$, this leads to

$$\exists x(t_1 \approx t_2 \wedge \ldots \wedge t_1 \approx t_{j-1} \wedge \neg(t_1 \approx t_j) \wedge \ldots \wedge \neg(t_1 \approx t_m)),$$

a formula, which is equivalent to the original one. (Note, that the matrix of the original formula A is quantifier-free.) We have now a formula in the matrix of which x no longer occurs, so the quantifier may be omitted. Now, we choose B as formula

$$t_1 \approx t_2 \wedge \ldots \wedge t_1 \approx t_{j-1} \wedge \neg(t_1 \approx t_j) \wedge \ldots \wedge \neg(t_1 \approx t_m).$$

With this choice, we have again that $M \models A \leftrightarrow B$ holds.

b) The additional property is an immediate consequence of the construction of B. Either B is closed (cases 1 and 2) or the sets of the free variables of A and B are identical (case 3). □

Let L be a first-order language with at least one constant symbol. Furthermore, let T be a theory over L such that each sentence of T is a *prenex universal formula,* i.e., of the form $\forall x_1 \ldots \forall x_n A$, where $n \geq 0$ and A (the "matrix" of the formula) is quantifier-free. If T has a model, then it has also a term-generated one. For a logic without equality a proof of this well-known fact can e.g., be found in [8], p. 19; the generalization of this proof to a logic with equality is trivial.

As an immediate consequence, we obtain:

Lemma 4.2 *Assume A to be a sentence built up over the language L_Z'. If there is a model of the theory $T_Z \cup \{A\}$, then there is also a term-generated one.*

Proof. We use Lemma 4.1 and obtain that for every sentence A over L_Z there exists a quantifier-free sentence B over the same language such that the class of all models of $T_Z \cup \{A\}$ equals the class of all models of $T_Z \cup \{B\}$. Each sentence of T_Z is a prenex universal formula. Since B is a prenex universal formula, too, the above mentioned property of the class of these formulae applies. □

After these preparations, we are now able to prove the desired result.

Theorem 4.3 *There is no finite theory over the first-order language L_Z which has the structure Z as - up to isomorphism - only term-generated model.*

Proof. Suppose, for a contradiction, that we are given a finite theory $\{A_1, \ldots, A_m\}$ over the language L_Z which has - up to isomorphism - the structure Z as only term-generated model. We define the sentence A by $A := A_1 \wedge \ldots \wedge A_m$.

Claim: Each finite subset S of the theory $T_Z \cup \{\neg A\}$ has a model.

Proof: If $\neg A \notin S$, then Z is a model. Otherwise, let $k := \max\{n : (3.n) \in S\}$. We define a structure M for the language L_Z as a "loop of size $k+1$", i.e., by $|M| := \{0, \ldots, k\}$ and

$$z_M := 0 \qquad s_M(u) := \begin{cases} u+1 & \text{if } u \neq k \\ 0 & \text{if } u = k \end{cases} \qquad p_M(u) := \begin{cases} u-1 & \text{if } u \neq 0 \\ k & \text{if } u = 0. \end{cases}$$

It is obvious that the sentences (1), (2) and (3.n), where $1 \leq n \leq k$, are true in M. Also $M \models \neg A$ holds. Otherwise, we would have $M \models A$ which implies (the structure M is term-generated) that M and Z were isomorphic. Thus, we have a contradiction. Summing up, M is a model of S.

Now, we use the compactness theorem of first-order logic to deduce that the theory $T_Z \cup \{\neg A\}$ has a model. In combination with Lemma 4.2 this implies the existence of a term-generated model M of $T_Z \cup \{\neg A\}$. M is also a term-generated model of T_Z. From this fact and Theorem 3.3 we obtain that the two models M and Z of T_Z are isomorphic. As a consequence, $M \models A$ holds. But this is a contradiction to $M \models \neg A$. □

Consider the sub-theory of T_Z containing the two sentences (1) and (2) only. It can be shown that each term-generated model of this theory is either isomorphic to Z or to a "loop of size n". In the manner of speaking of algebraic specifications or universal algebra, Z is initial in the class of all term-generated models of $\{(1), (2)\}$. To obtain this model as – up to isomorphism – only term-generated model, one has to extend the theory in such a way that loops are prevented, i.e., infinitely many inequalities can be derived. Theorem 4.3 states that the language used so far is too "poor" to do this in a finite manner.

5 A finite characterization of the integers using a hidden function

As just mentioned, a finite extension of the theory $\{(1), (2)\}$ which prevents loops requires an extension of the language L_Z, i.e., the use of hidden machinery. In this section we show, that a symbol for the usual ordering on the integers suffices. To this end, we extend the language L_Z to $\overline{L}_Z := L_Z \cup \{\ll\}$, where $\ll$ is a 2-place predicate symbol. Furthermore, we consider the three sentences (the symbol $\ll$ is used in infix notation)

$$(3)\ \forall x(\neg(s(x) \ll x)) \quad (4)\ \forall x(x \ll x) \quad (5)\ \forall x \forall y(s(x) \ll y \rightarrow x \ll y).$$

And, finally, we define the finite theory $\overline{T}_Z$ over $\overline{L}_Z$ to consist of the sentences (1) and (2) of T_Z and the sentences (3), (4), and (5).

Clearly, the structure $\overline{Z} := (\mathbb{Z}, 0, \text{succ}, \text{pred}, \leq)$ for $\overline{L}_Z$ is a term-generated model of $\overline{T}_Z$. In the rest of this section we prove that each other term-generated model is isomorphic to this model. As in Section 3, therefore, we assume in the sequel an arbitrarily chosen (but fixed) term-generated model $M := (|M|, z_M, s_M, p_M, \ll_M)$ of $\overline{T}_Z$. In the following, we write $u \ll_M v$ (resp. $u \not\ll_M v$) instead of $\ll_M (u, v) = 1$ (resp. $\ll_M (u, v) = 0$). As in the case of T_Z we obtain:

Lemma 5.1 *Let $u \in |M|$. Then there exists exactly one natural number $n \in \mathbb{N}$ such that $u = s_M^n(z_M)$ or $u = p_M^n(z_M)$.*

Proof. As the existence of n follows from the validity of (1) and (2) in M (cf. the proof of Lemma 3.1), it remains to show uniqueness.

If $u = s_M^m(z_M) = s_M^n(z_M)$ and $m + k = n$ (where $k > 0$), then

$$\begin{aligned} s_M^m(z_M) = s_M^m(s_M^k(z_M)) \quad &\Rightarrow \quad s_M^k(Z_M) = Z_M \Rightarrow s_M^k(z_M) \ll_M z_M \\ &\Rightarrow \quad s_M(z_M) \ll_m z_M, \end{aligned}$$

since (2), (4), and (5) are true in M. However, $s_M(z_M) \ll_M z_M$ is a contradiction to the validity of formula (3) in M.

The remaining cases are handled similarly. □

The proof of the fact that the function Φ of the third section is an isomorphism from M to $\overline{Z}$, too, is prepared by a simple

Lemma 5.2 *If $u \in |M|$ and $n \in \mathbb{N}$, then $n > 0$ implies $s_M^n(u) \not\ll_M u$.*

We use induction on n. The induction base $n = 1$ holds since sentence (3) is true in M; the induction step proceeds as follows: From the validity of (5) in M we obtain

$$s_M^{n+1}(u) \ll_M u \Rightarrow s_M^n(u) \ll_M u$$

and, thus, contraposition in conjunction with the induction hypothesis applies. □

Now, we are able to prove:

Lemma 5.3 *The function Φ of Section 3 is also an isomorphism from the fixed model M to the model $\overline{Z}$.*

Proof. Due to Lemma 3.2 of the third section, we have only to prove that Φ preserves the two interpretations $\ll_M$ and $\leq$ of the predicate symbol $\ll$, i.e., that for all $u, v \in |M|$

$$u \ll_M v \Leftrightarrow \Phi(u) \leq \Phi(v).$$

Assume $u = s_M^n(z_M)$ and $v = s_M^m(z_M)$. For a proof of direction "$\Rightarrow$" we show that $\Phi(u) \not\leq \Phi(v)$ implies $u \not\ll_M v$. From $\Phi(u) \not\leq \Phi(v)$ we get $m > n$, hence $m = k + n$, where $k > 0$. Thus,

$$u = s_M^{k+n}(z_M) = s_M^k(s_M^n(z_M)) = s_M^k(v).$$

Due to this result, $u \not\ll_M v$ is equivalent to $s_M^k(v) \not\ll_M v$ and Lemma 5.2 applies. Now, we prove direction "$\Leftarrow$". From $\Phi(u) \leq \Phi(v)$ we obtain that $m \leq n$ holds, i.e., $k + m = n$, where $k \geq 0$. This shows the equation

$$s_M^k(u) = s_M^k(s_M^m(z_M)) = s_M^{k+m}(z_M) = v.$$

In combination with the validity of (4) in M, this result yields $s_M^k(u) \ll_M v$ which in turn implies (since (5) is true in M) that $u \ll_M v$.

Next, let $u = s_M^m(z_M)$ and $v = p_M^n(z_M)$. For a proof of "$\Rightarrow$" we distinguish between $m+n=0$ and $m+n>0$. The first case is trivial. In the second case we use that (1) is true in M and get

$$u \ll_M v \Leftrightarrow s_M^m(s_M^n(p_M^n(z_M))) \ll_M p_M^n(z_M) \Leftrightarrow s_M^{m+n}(p_M^n(z_M)) \ll_M p_M^n(z_M).$$

Now, Lemma 5.2 shows that the premise of the implication to be proven does not hold. A proof of "$\Leftarrow$" is trivial.

The remaining cases can be shown analogously. □

We now have that the structure $\overline{Z}$ is characterized by the theory $\overline{T}_Z$:

Theorem 5.4 *The model $\overline{Z}$ is – up to isomorphism – the only term-generated model of $\overline{T}_Z$.* □

It is obvious that the use of a predicate symbol for the ordering (in combination with an extension of the theory $\{(1),(2)\}$) is not the only way to prevent loops. E.g., one can also extend the langauge L_Z by a predicate symbol n and $\{(1),(2)\}$ by the four sentences

$$\begin{array}{llll} (6) & \neg n(z) & (7) & n(p(z)) \\ (8) & \forall x(n(x) \rightarrow n(p(x))) & (9) & \forall x(n(s(x)) \rightarrow n(x)) \end{array}$$

which specify the interpretation of n to test a given integer for being negative or not. Another possibility is to introduce inductively (using z, s, and p) a 2-place function symbol f that describes the repeated application of the symbols s and p, resp. A natural way to specify f is

$$\begin{array}{ll} (10) & \forall x(f(x,z) \approx x) \\ (11) & \forall x \forall y(f(x,s(y)) \approx s(f(x,y))) \\ (12) & \forall x \forall y(f(x,p(y)) \approx p(f(x,y))). \end{array}$$

We may then substitute in the theory T_Z the infinite set (3.n), $n \geq 1$, of sentences by a single one, viz.

$$(13)\ \forall x \forall y(\neg(y \approx z) \rightarrow \neg(f(x,y) \approx z)).$$

In both cases, the proof of isomorphism is mainly a consequence of (the validity of) Lemma 3.1.

We finish this section with a remark concerning our proof method. Certain, our "model-oriented" approach is not the only way to solve the given problem. For instance, a proof which argues algebraically can proceed as follows: One shows that the initial term-generated model Z of the theory $\{(1),(2)\}$ can be extended by the ordering relation $\leq$ in such a way that the resulting structure $\overline{Z}$ for $\overline{L}_Z$ is initial wrt. T_Z. Since the truth values 0 and 1 are different, the ordering relation cannot identify elements. Now, the desired isomorphism result is an immediate consequence of the initiality of $\overline{Z}$. This remark shows also: For a translation of the proof of this section into the notation of algebraic specifications a specification of the truth values is required which has – up to isomorphism – the two element Boolean algebra as only model.

6 Concluding remarks

From a theoretical point of view, hidding machinery is used to overcome the lack of expressive power. In the present paper we have shown its necessity even in the case of full first-order specifications. To this end, first, we have presented an infinite first-order theory T_Z whose term-generated models are exactly the structures isomorphic to $Z = (\mathbb{Z}, 0, \text{succ}, \text{pred})$. Then we have shown that there is no finite set of first-order sentences which has the same property. And, finally, we have given unique characterizations of Z using hidding machinery.

For the proof of the main result (Theorem 4.3) we have used the argument that the theory $T_Z \cup \{\neg A\}$ has a term-generated model if every of its finite subsets has a model. It seems that this argument (an extension of the compactness theorem of first-order logic) can also be used to prove that there is no finite characterization of more complex data types without hidden functions.

For the description of large structures and systems it is necessary to compose specifications in a modular way from smaller ones to master complexity. Hidding is one of these so-called specification-building operations and contained in almost all modern specification languages; see [13] for an overview. Frequently, its use makes specifications more readable and understandable. Furthermore, in various case studies it has proven advantageous to use hidding if specifications are transformed, e.g., into versions which provide algorithmic solutions. As two examples for this latter application we mention the papers [5] and [4]. In all these cases the decisive question is how to find suitable hidden functions and their defining formulae. This aspect of hidding was not addressed here, but some work can be found in the literature. However, it seems that a general methodology for the *practical* use of hidden machinery remains to be developed.

Acknowledgement. This paper benefited from valuable discussions with Ulf Schmerl. I am also grateful to the referees for their helpful comments.

References

[1] Bauer F.L., Wössner H.: Algorithmic Language and Program Development. Springer Verlag, Berlin-Heidelberg-New York (1982)

[2] Bergstra J.A., Tucker J.V.: Algebraic Specifications of Computable and Semicomputable Data Structures. Research Report IW 115, Department of Computer Science, Mathematical Centre, Amsterdam (1979)

[3] Bergstra J.A., Tucker J.V.: Algebraic Specifications of Computable and Semicomputable Data Types. *Theoretical Computer Science* 50, 137-181 (1987)

[4] Broy M.: Deductive Program Development: Evaluation in Reverse Polish Notation as an Example. In: Broy M., Wirsing M. (eds.): Methods of Programming. LNCS 544, Springer Verlag, Berlin-Heidelberg-New York, 79-99 (1991)

[5] Dosch W., Wirsing M., Ausiello G., Mascari G.F.: Polynomials – The Specification, Analysis and Development of an Abstract Data Type. In: Wilhelm R. (ed.): GI - 10. Jahrestagung, Informatik Fachberichte 33, Springer Verlag, Berlin-Heidelberg-New York, 306-320 (1991)

[6] Ehrig H., Mahr B.: Fundamentals of Algebraic Specifications 1. Equations and Initial Semantics. EATCS Monographs in Theoretical Computer Science, Vol. 6, Springer Verlag, Berlin-Heidelberg-New York (1985)

[7] Enderton H.B.: A Mathematical Introduction to Logic. Academic Press, London (1972)

[8] Kreisel G., Krivine J.-L.: Modelltheorie – Eine Einführung in die mathematische Logik. Hochschultexte, Springer Verlag, Berlin-Heidelberg-New York (1972)

[9] Majster M.: Data Types, Abstract Data Types and their Specification Problem. Report TUM-I7740, Institut für Informatik, TU München (1977) Also in: *Theoretical Computer Science 8,* 89-127 (1979)

[10] Monk J.D.: Mathematical Logic. Springer Verlag, New York-Heidelberg-Berlin (1976)

[11] Thatcher J.W., Wagner E. G., Wright J.B.: Data Type Specification: Parameterization and the Power of Specification Techniques. Proc. 10^{th} SIGACT Symp. Theory of Computing, 119-132 (1978)

[12] Wirsing M., Pepper P., Partsch H., Dosch D., Broy M.: On Hierarchies of Abstract Data Types. *Acta Informatica* 20, 1-33 (1983)

[13] Wirsing M.: Algebraic Specifications. In: van Leeuwen J. (ed.): Handbook of Theoretical Computer Science B, North-Holland, 675-788 (1990)

Received October 22, 1992