Front. Comput. Sci., 2016, 10(4): 717–725
DOI 10.1007/s11704-015-4024-2

RESEARCH ARTICLE

# The M-computations induced by accessibility relations in nonstandard models M of Hoare logic

**Cungen CAO (✉)[1], Yuefei SUI[1], Zaiyue ZHANG[2]**

1 Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
2 Department of Computer Science, Jiangsu University of Sciences and Technology, Zhenjiang 212003, China



**Abstract** Hoare logic is a logic used as a way of specifying semantics of programming languages, which has been extended to be a separation logic to reason about mutable heap structure. In a model $\mathbf{M}$ of Hoare logic, each program $\alpha$ induces an $\mathbf{M}$-computable function $f_\alpha^{\mathbf{M}}$ on the universe of $\mathbf{M}$; and the $\mathbf{M}$-recursive functions are defined on $\mathbf{M}$. It will be proved that the class of all the $\mathbf{M}$-computable functions $f_\alpha^{\mathbf{M}}$ induced by programs is equal to the class of all the $\mathbf{M}$-recursive functions. Moreover, each $\mathbf{M}$-recursive function is $\Sigma_1^{\mathbf{N}^\mathbf{M}}$-definable in $\mathbf{M}$, where the universal quantifier is a number quantifier ranging over the standard part of a nonstandard model $\mathbf{M}$.

**Keywords** Hoare logic, recursive function, computable function, nonstandard model of Peano arithmetic

## 1 Introduction

Hoare [1] introduced an axiomatic method of proving the correctness of programs, Hoare logic, which has been used as a way of specifying semantics of programming languages [2–5]. Separation logic is a spatial logic for reasoning about mutable heap structure [6–8], which is an extension of Hoare logic to describe the applications of programs on the heap structures and the reasoning about memory update. It would be interesting to reconsider Hoare logic and the computability induced by the accessibility relations in a model of Hoare logic, because the models of Hoare logic, taken as a modal logic, are defined on the assignments of models of the first-order languages based on which Hoare logic is defined, which induces the special form of the completeness theorem different from that of the first-order logic.

Hoare logic $H$ is complete if for every interpretation $I$, set of assertions $A$, and specification $\phi$, the following holds: if $A \models_H^I \phi$, then $A \vdash_H \phi$, where $A \models_H^I \phi$ means that $I \models_H A$ implies $I \models_H \phi$ [2].

Cook's completeness theorem of Hoare logic [9] says that for any model $\mathbf{M}$ of Hoare logic, if $\mathbf{M}$ is expressive for programs then $\mathbf{M} \models \phi_1[\alpha]\phi_2$ implies $\mathcal{HL}(\mathbf{M}) \vdash \phi_1[\alpha]\phi_2$, where $\mathcal{HL}(\mathbf{M})$ is the Hoare logic with axioms Th($\mathbf{M}$). Bergstra and Tucker [3] proved that for any theory $T$ of Peano arithmetic, $\mathcal{HL}(T)$, Hoare logic with axioms $T$ is logically complete if and only if $T$ is complete, that is, for any sentence $\phi$ of Peano arithmetic, either $\phi \in T$ or $\neg\phi \in T$.

The theorems on the completeness of Hoare logic show the difference of Hoare logic from the traditional logics: there is no uniform completeness theorem for Hoare logic, even though it was proved recently [5] that the propositional Hoare logic is complete.

The class of all program-computable functions on natural numbers is equal to the class of all recursive functions [10]. In Hoare logic, the programs are abstracted to modalities $[\alpha]$, which are interpreted as the accessibility relations on any models $\mathbf{M}$ of Peano arithmetic [2,4,11]. Given a model $\mathbf{M}$ of Hoare logic, each program $\alpha$ induces a function $f_\alpha^{\mathbf{M}}$ on the universe of $\mathbf{M}$ (such a function is called $\mathbf{M}$-computable in the following).

Received January 19, 2014; accepted August 2, 2015

E-mail: cgcao@ict.ac.cn

In a model $\mathbf{M}$ of Hoare logic, as in the classical recursion theory, we can define another class of functions, the $\mathbf{M}$-recursive functions, which are defined by the elementary functions and via the composition, the recursion and the $\mu$-operator, where the recursion is up to numbers in the standard part $\mathbf{N}^{\mathbf{M}}$ of $\mathbf{M}$, and the $\mu$-operator is restricted on $\mathbf{N}^{\mathbf{M}}$, where the standard part of $\mathbf{M}$ is denoted by $\mathbf{N}^{\mathbf{M}}$, which is isomorphic to the standard model $\mathbf{N}$ of Peano arithmetic. It will be proved that for any model $\mathbf{M}$ of Hoare logic, the class of all $\mathbf{M}$-computable functions is equal to the class of all $\mathbf{M}$-recursive functions.

A basic property of recursive functions is the $\Sigma_1$-definability. For the standard model $\mathbf{N}$ of Peano arithmetic, for any recursive function $f : N \to N, \mathrm{dom}(f)$ (the domain of $f$) is $\Sigma^0_1$ in $\mathbf{N}$, and hence arithmetical in $\mathbf{N}$. For the nonstandard model $\mathbf{M}$ of Peano arithmetic, there is an $\mathbf{M}$-recursive (computable) function $f^{\mathbf{M}}_{\alpha}$ for some program $\alpha$ such that $\mathrm{dom}(f^{\mathbf{M}}_{\alpha})$ is not arithmetical. If for any model $\mathbf{M}$ of Peano arithmetic and any program $\alpha$, $f^{\mathbf{M}}_{\alpha}$ is $\Sigma^0_1$ in $\mathbf{M}$ (at least definable in $\mathbf{M}$) then Hoare's logic would be reduced to a first-order theory, and hence, Hoare's logic is complete, where $\phi[\alpha]\psi$ is reduced to $\phi\wedge\theta \to \psi$, where $\theta$ is the formula defining $\alpha$ in $\mathbf{M}$. We shall use a new quantifier $\forall\mathbf{m}$, where $\mathbf{m}$ ranges in the standard part of a nonstandard model, and prove that with such a quantifier, $N^{\mathbf{M}}$ is arithmetical in $\mathbf{M}$ and each $\mathbf{M}$-recursive function is $\Sigma^{\mathbf{N}^{\mathbf{M}}}_1$-definable in $\mathbf{M}$.

The rest of the paper is organized as follows. Section 2 gives the basic definitions on the standard and nonstandard models of Peano arithmetic, and Hoare logic, its syntax and semantics. Section 3 defines $\mathbf{M}$-computable functions for any model $\mathbf{M}$ of Peano arithmetic. Section 4 defines the $\mathbf{M}$-recursive functions and gives the basic properties of $\mathbf{M}$-recursive functions. Section 5 proves that the class of all $\mathbf{M}$-computable functions is equal to the class of all $\mathbf{M}$-recursive functions. Section 6 defines the $\mathbf{M}^-$-recursive functions and shows that $+^{\mathbf{M}}, \cdot^{\mathbf{M}}$ are not $\mathbf{M}^-$-recursive. Section 7 gives a new logical language of Peano arithmetic with quantifier $\forall\mathbf{m}$ and proves that each $\mathbf{M}$-recursive function is $\Sigma^{\mathbf{N}\mathbf{M}}_1$-definable, and the whole paper is concluded in the final section.

Our notation is standard, a reference is [10]. We shall use $a, b$ to be elements of nonstandard models; $m, n$, natural numbers of the standard parts of nonstandard models of Peano arithmetic; $\mathbf{n}$ closed terms of logical language $L$; $\mathbf{m}$ number variables of logical language $L^+$, and $x, y$, variables in the logical language of Peano arithmetic.

## 2 The basic definitions

Let $L = \{+, \cdot, <, \mathbf{0}, \mathbf{1}\}$ be the logical language of Peano arithmetic.

Let $\mathbf{N} = (N, +^{\mathbf{N}}, \cdot^{\mathbf{N}}, <^{\mathbf{N}}, \mathbf{0}^{\mathbf{N}}, \mathbf{1}^{\mathbf{N}})$ be the standard model of Peano arithmetic; and $\mathbf{M} = (M, +^{\mathbf{M}}, \cdot^{\mathbf{M}}, <^{\mathbf{M}}, 0^{\mathbf{M}}, 1^{\mathbf{M}}, I)$ be a countable nonstandard model of Peano arithmetic, that is, $\mathbf{M} \models \phi$ for each axiom $\phi$ of Peano arithmetic, where $M$ is the universe of $\mathbf{M}$, and $I$ is an interpretation such that

$$I(+, \mathbf{M}) = +^{\mathbf{M}};\ I(\cdot, \mathbf{M}) = \cdot^{\mathbf{M}};$$
$$I(<, \mathbf{M}) = <^{\mathbf{M}};$$
$$I(\mathbf{0}, \mathbf{M}) = \mathbf{0}^{\mathbf{M}};\ \ I(\mathbf{1}, \mathbf{M}) = \mathbf{1}^{\mathbf{M}}.$$

In the following we shall omit $I$ in $\mathbf{M}$.

There is a submodel $\mathbf{N}^M = (N^M, +^{\mathbf{N}^M}, \cdot^{\mathbf{N}^M}, <^{\mathbf{N}^M}, 0^{\mathbf{N}^M}, 1^{\mathbf{N}^M})$ of $\mathbf{M}$ such that $\mathbf{N}^M$ is isomorphic to $\mathbf{N}$, where $N^M$ is the universe of $\mathbf{N}^M$,

$$N^M = \{I(\mathbf{n}, \mathbf{M}) : n \in N\},$$

and

$$I(+, \mathbf{N}^{\mathbf{M}}) = +^{\mathbf{N}^{\mathbf{M}}} = +^{\mathbf{M}} \upharpoonright N^{\mathbf{M}};$$
$$I(\cdot, \mathbf{N}^{\mathbf{M}}) = \cdot^{\mathbf{N}^{\mathbf{M}}} = \cdot^{\mathbf{M}} \upharpoonright N^{\mathbf{M}};$$
$$I(\mathbf{0}, \mathbf{N}^{\mathbf{M}}) = \mathbf{0}^{\mathbf{N}^{\mathbf{M}}} = \mathbf{0}^{\mathbf{M}};$$
$$I(\mathbf{1}, \mathbf{N}^{\mathbf{M}}) = \mathbf{1}^{\mathbf{N}^{\mathbf{M}}} = \mathbf{1}^{\mathbf{M}};$$
$$I(<, \mathbf{N}^{\mathbf{M}}) = <^{\mathbf{N}^{\mathbf{M}}} = <^{\mathbf{M}} \upharpoonright N^{\mathbf{M}}.$$

The logical language $L'$ for Hoare logic contains the following symbols:

- constants for numbers: $\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots$;
- variables for numbers: $x_0, x_1, \ldots$;
- function symbols: $+, \cdot$;
- binary predicate symbol: $<$;
- the logical connectives and quantifiers: $\neg, \to, \forall$;
- command symbols: :=, ; , **if then else**, **while do**.

An expression $t$ is defined as follows:

$$t ::= \mathbf{n}|x|t_1 + t_2|t_1 \cdot t_2;$$

and an boolean expression $\theta$ is defined as follows:

$$\theta ::= t_1 < t_2|\neg\theta_1|\theta_1 \to \theta_2.$$

A program $\alpha$ is defined as follows:

$$\alpha ::= x := t|\alpha_1; \alpha_2|\ \textbf{if}\ \theta\ \textbf{then}\ \alpha_1\ \textbf{else}\ \alpha_2|\ \textbf{while}\ \theta\ \textbf{do}\ \alpha_1.$$

An assertion $\phi$ is defined as follows:

$$\phi ::= \theta|\neg\phi|\phi_1 \to \phi_2|\forall x\phi_1(x).$$

A specification $\Phi$ is an Hoare triple of the following forms:

$$\Phi ::= \phi_1[\alpha]\phi_2.$$

The semantics for Hoare logic is a Kripke's possible world semantics, where the possible worlds are stores, and $[\alpha]$ are interpreted as modalities.

Given a model $\mathbf{M} = (M, I)$ of Peano arithmetic, let $W$ be the set of all the assignments (stores), functions from variables to $M$.

A model $\mathbf{M}'$ is a triple $(W, \{R_\alpha : \alpha\}, \mathbf{M})$, such that for each $\alpha, R_\alpha$ is a binary relation on $W$, such that for any $w, w' \in W$,

• $(w, w') \in R_{x:=t}$ iff $w' = w(x/t^{I,w})$, where $t^{I,w}$ is defined in the following;

• $(w, w') \in R_{\alpha_1;\alpha_2}$ iff $(w, w') \in R_{\alpha_1} \circ R_{\alpha_2}$, where $(z, z') \in R_1 \circ R_2$ iff there is a $z''$ such that $(z, z'') \in R_1$ and $(z'', z') \in R_2$;

• $(w, w') \in R_{\textbf{if } \theta \textbf{ then } \alpha_1 \textbf{ else } \alpha_2}$ iff either $\mathbf{M}, w \models \theta$ and $(w, w') \in R_{\alpha_1}$, or $\mathbf{M}, w \not\models \theta$ and $(w, w') \in R_{\alpha_2}$;

• $(w, w') \in R_{\textbf{while } \theta \textbf{ do } \alpha_1}$ iff there are $w_0 = w, w_1, \ldots, w_i = w'$ such that for each $0 \leqslant j \leqslant i-1$, $\mathbf{M}, w_j \models \theta, (w_j, w_{j+1}) \in R_{\alpha_1}$, and $\mathbf{M}, w' \not\models \theta$.

Given a possible world $w$ and an expression $t$, the interpretation $t^{I,w}$ of $t$ in $w$ is illustrated as follows:

$$t^{I,w} = \begin{cases} I(\mathbf{n}), & \text{if } t = \mathbf{n}; \\ w(x), & \text{if } t = x; \\ t_1^{I,w} +^{\mathbf{M}} t_2^{I,w}, & \text{if } t = t_1 + t_2; \\ t_1^{I,w} \cdot^{\mathbf{M}} t_2^{I,w}, & \text{if } t = t_1 \cdot t_2, \end{cases}$$

and given a possible world $w$ and a boolean expression $\theta, \theta$ is satisfied in $w$, denoted by $\mathbf{M}, w \models \theta$, if

$$\begin{cases} t_1^{I,w} <^{\mathbf{M}} t_2^{I,w}, & \text{if } \theta = t_1 < t_2; \\ \mathbf{M}, w \not\models \theta_1, & \text{if } \theta = \neg\theta_1; \\ \mathbf{M}, w \models \theta_1 \Rightarrow \mathbf{M}, w \models \theta_2, & \text{if } \theta = \theta_1 \rightarrow \theta_2, \end{cases}$$

where instead of using $\Rightarrow$ as the logical implication in Hoare logic, we use $\rightarrow$[1)].

Given an assertion $\phi$ and a possible world $w$, we say that $\phi$ is satisfied in $w$, denoted by $\mathbf{M}, w \models \phi$, if

$$\begin{cases} \mathbf{M}, w \models \theta, & \text{if } \phi = \theta; \\ \mathbf{M}, w \not\models \phi_1, & \text{if } \phi = \neg\phi_1; \\ \mathbf{M}, w \models \phi_1 \Rightarrow \mathbf{M}, w \models \phi_2, & \text{if } \phi = \phi_1 \rightarrow \phi_2; \\ \mathbf{A}a \in M(\mathbf{M}, w(x/a) \models \phi_1(\mathbf{m})), & \text{if } \phi = \forall x \phi_1(x). \end{cases}$$

Given a possible world $w$, a specification $\phi_1[\alpha]\phi_2$ is satisfied at $w$, denoted by $\mathbf{M}, w \models \phi_1[\alpha]\phi_2$, if

$$\mathbf{M}, w \models \phi_1 \Rightarrow \mathbf{A}w'((w, w') \in R_\alpha \Rightarrow \mathbf{M}, w' \models \phi_2).$$

## 3 The M-computability induced by the accessibility relations

Given a model $\mathbf{M}' = (W, \{R_\alpha : \alpha\}, \mathbf{M})$ of Hoare logic, for any program $\alpha$, let $R_\alpha$ be the accessibility relation for $\alpha$.

$R_\alpha$ induces a function $f_\alpha^M : M^i \rightarrow M$ such that for any $\vec{\mathbf{a}} \in M^i$ and $b \in M$,

$$f_\alpha^M(\vec{\mathbf{a}}) = b,$$

if and only if there are $w, w' \in W$ and $x_1, \ldots, x_i$ such that $(w, w') \in R_\alpha$ and

$$w(x_1) = a_1, \ldots, w(x_i) = a_i;$$
$$w(x) = b.$$

We say that $f_\alpha^M$ is **M**-computable.

By the definition of $R_\alpha$, we have the following lemma.

**Lemma 1** $f_\alpha^M$ is well-defined. That is, for any $v, v', w, w' \in W$ with $(v, v'), (w, w') \in R_\alpha$, if

$$v(x_1) = w(x_1), \ldots, v(x_i) = w(x_i),$$

then

$$v'(x) = w'(x).$$

**Proof** By the induction on the structure of $\alpha$. □

**Example 1** Let $\alpha ::= y := y + 1$. Then, for any $w, w' \in W, (w, w') \in R_\alpha$ iff $v' = v(y/(v(y) +^{\mathbf{M}} \mathbf{1}^{\mathbf{M}}))$, where for any $x$,

$$v(y/a)(x) = \begin{cases} v(x), & \text{if } x \neq y; \\ a, & \text{otherwise}. \end{cases}$$

$R_\alpha$ induces a function $f_\alpha^{\mathbf{M}} : M \rightarrow M$ such that for any $a, b \in M$,

$$f_\alpha^{\mathbf{M}}(a) = b \text{ iff } b = a +^{\mathbf{M}} \mathbf{1}^{\mathbf{M}}.$$

That is, $f_\alpha^{\mathbf{M}}(a) = a +^{\mathbf{M}} \mathbf{1}^{\mathbf{M}}$. □

**Example 2** Let

$$\alpha(a, m) ::= x := a; \textbf{while } y \neq m \textbf{ do } (x := x + 1; y := y + 1).$$

Then, for any $w, w' \in W, (w, w') \in R_\alpha$ iff there are $w_1, \ldots, w_m$ such that $(w, w_1) \in R_{x:=x+1}, (w_m, w') \in R_{x:=x+1}$ and for each $1 \leqslant i < m, (w_i, w_{i+1}) \in R_{x:=x+1}$.

$R_\alpha$ induces a function $f_\alpha^{\mathbf{M}}: M \rightarrow M$ such that for any $a \in M$ and $m \in N^M$,

$$f_\alpha^{\mathbf{M}}(a, m) = b \text{ iff } b = a +^{\mathbf{M}} m.$$

That is, $f_\alpha^{\mathbf{M}}(a, m) = a +^{\mathbf{M}} m$. □

**Example 3** Let

$$\alpha(x) ::= y := 0; \textbf{while } x \neq y \textbf{ do } y := y + 1.$$

1) In syntax, we use $\neg, \wedge, \rightarrow, \forall, \exists$ to denote the logical connectives and quantifiers; and in semantics we use $\sim, \&, \Rightarrow, \mathbf{A}, \mathbf{E}$ to denote the corresponding connectives and quantifiers

For any $w, w' \in W, (w, w') \in R_\alpha$ if and only if

1) $w(y) = 0$;
2) $w'(y) = w'(x)$;
3) $w(x)$ is standard.

The corresponding function $f_\alpha^{\mathbf{M}}$ is such that for any $a \in M$,

$$f_\alpha^{\mathbf{M}}(a) = \begin{cases} a, \text{ if } a \in N^M; \\ \uparrow, \text{ otherwise.} \end{cases} \qquad \square$$

Notice that the domain $\text{dom}(f_\alpha^{\mathbf{M}}) = N^M$ of $f_\alpha^{\mathbf{M}}$ is not arithmetical in **M** and the first-order logic, that is, there is no formula $\phi(x)$ such that for any $a \in M$,

$$a \in \text{dom}(f_\alpha^{\mathbf{M}}) \text{ iff } \mathbf{M} \models \phi(x/a).$$

$\text{dom}(f_\alpha^{\mathbf{M}}) = N^M$ is arithmetical in **M** and Hoare logic, that is, there is a formula $\phi_1[\alpha]\phi_2$ such that for any $a \in M$,

$$a \in \text{dom}(f_\alpha^{\mathbf{M}}) = N^M \text{ iff } \mathbf{M} \models \phi(x/a)[\alpha(x/a)]\psi(x/a).$$

In fact, let $\phi_1[\alpha]\phi_2 = \mathbf{true}[\alpha]\mathbf{true}$. Then,

$$\begin{aligned} \mathbf{M} \models \phi_1(x/a)[\alpha(x/a)]\psi(x/a) \text{ iff } & \alpha(x/a) \text{ terminates} \\ \text{iff } & a \in \text{dom}(f_\alpha^{\mathbf{M}}). \end{aligned}$$

**Example 4** Let

$$\alpha(x, z) ::= y := z; \mathbf{while}\ x \neq y\ \mathbf{do}\ y := y + 1.$$

For any $w, w' \in W, (w, w') \in R_\alpha$ if and only if

1) $w(y) = w(z)$;
2) $w'(y) = w'(x)$;
3) $w(x) = w(z) +^{\mathbf{M}} i$ for some standard number $i$.

The corresponding function $f_\alpha(x, z)$ is such that for any $a, b \in M$,

$$f_\alpha(a, b) = \begin{cases} a, \text{ if } a = b +^{\mathbf{M}} i \text{ for some } i \in N^{\mathbf{M}}; \\ \uparrow, \text{ otherwise.} \end{cases} \qquad \square$$

Let $\mathbf{C^M}$ be the set of all **M**-computable functions.

## 4 The M-recursive functions

The **M**-recursive functions are a generalization of the recursive functions from the standard model to the nonstandard models of Peano arithmetic.

**Definition 1** A function $f : M^i \to M$ is **M**-*recursive* if either

1) $f$ is elementary, that is,

$$f = \lambda a.n | \lambda a, b.(a +^{\mathbf{M}} b) | \lambda a, b.(a \cdot^{\mathbf{M}} b),$$

where $n \in N^M$;

2) (Composition) there are **M**-recursive functions $f_1(\vec{\mathbf{a}})$ and $f_2(b, \vec{\mathbf{a}})$ such that for any $\vec{\mathbf{a}} \in M$,

$$f(\vec{\mathbf{a}}) = f_2(f_1(\vec{\mathbf{a}}), \vec{\mathbf{a}});$$

there are **M**-recursive functions $f_1(\vec{\mathbf{a}})$ and $f_2(m, \vec{\mathbf{a}})$ such that $\text{range}(f_1) \subseteq N^M$, and for any $\vec{\mathbf{a}} \in M$,

$$f(\vec{\mathbf{a}}) = f_2(f_1(\vec{\mathbf{a}}), \vec{\mathbf{a}});$$

3) (Recursion) there are **M**-recursive functions $f_1(\vec{\mathbf{a}}), f_2(m, b, \vec{\mathbf{a}})$ such that for any $m \in N^M$ and $\vec{\mathbf{a}} \in M^i$,

$$\begin{cases} f(\mathbf{0^M}, \vec{\mathbf{a}}) = f_1(\vec{\mathbf{a}}); \\ f(m, \vec{\mathbf{a}}) = f_2(f(m -^{\mathbf{M}} \mathbf{1^M}, \vec{\mathbf{a}}), m, \vec{\mathbf{a}}), \end{cases}$$

or

4) ($\mu$-operator) there is a total **M**-recursive function $f_1(m, \vec{\mathbf{a}})$ such that for any $\vec{\mathbf{a}} \in M$,

$$f(\vec{\mathbf{a}}) = \mu m \in N^M(f_1(m, \vec{\mathbf{a}}) = \mathbf{0^M}).$$

Let $\mathbf{R^M}$ be the set of all **M**-recursive functions.

For example, function

$$f(m, a) = a - m = \begin{cases} a, & \text{if } m = \mathbf{0^M}; \\ f(m -^{\mathbf{M}} \mathbf{1^M}, a) -^{\mathbf{M}} \mathbf{1^M}, & \text{if } m = m' +^{\mathbf{M}} \mathbf{1^M}, \end{cases}$$

is **M**-recursive

**Proposition 1** The following function $f$ is **M**-recursive, where for any $a \in M$,

$$f(a) = \begin{cases} a, \text{ if } a \in N^M; \\ \uparrow, \text{ otherwise.} \end{cases}$$

**Proof** Let $g(a, m) = a -^{\mathbf{M}} m$. Define $f(a) = \mu m(g(a, m) = \mathbf{0^M}) = \mu m(a -^{\mathbf{M}} m = \mathbf{0^M})$. Then, for any $a \in M$,

$$f(a) \downarrow = a \text{ iff } a \in N^M. \qquad \square$$

**Proposition 2** $\lambda a : M, m : N^M.a^m$ is **M**-recursive.

**Proof** For any $a \in M$ and $m \in N^M$, if $m = \mathbf{0^M}$ then $a^m = \mathbf{1^M}$; if $m \neq \mathbf{0^M}$ then $a^m = a^{m -^{\mathbf{M}} \mathbf{1^M}} \cdot^{\mathbf{M}} a$. Hence, $\lambda a : M, m : N^M.a^m$ is **M**-recursive. $\square$

It can be proved that $\lambda a : M.n^a$ and $\lambda a, b : M.\ a^b$ are not **M**-recursive.

**Proposition 3** For any **M**-recursive function $f$, there is a polynomial $p$ with coefficients of form $f_\alpha : M^n \to N^{\mathbf{M}}$ such that for any $\vec{\mathbf{a}}, b \in M$,

$$f(\vec{\mathbf{a}}) = b \text{ iff } p(\vec{\mathbf{a}}) = b.$$

**Proof** We prove by induction on the recursiveness definition of $f$.

If $f$ is elementary, then the proposition holds.

Assume that $f(\vec{\mathbf{a}}) = f_1(f_2(\vec{\mathbf{a}}), \vec{\mathbf{a}})$ and the proposition holds for $f_1$ and $f_2$. Let $p_1, p_2$ be two polynomials such that for any $\vec{\mathbf{a}}, a \in M$,

$$f_1(a, \vec{\mathbf{a}}) = p_1(a, \vec{\mathbf{a}}); \quad f_2(\vec{\mathbf{a}}) = p_2(\vec{\mathbf{a}}).$$

Then, $f(\vec{\mathbf{a}}) = p_1(p_2(\vec{\mathbf{a}}), \vec{\mathbf{a}})$ is a polynomial.

Assume that $f$ is defined by the recursion and $f_1, f_2$, and the proposition holds for $f_1$ and $f_2$. Let $p_1, p_2$ be two polynomials such that for any $\vec{\mathbf{a}} \in M^i$ and $a \in M$,

$$f_1(\vec{\mathbf{a}}) = p_1(\vec{\mathbf{a}}); \quad f_2(a, \vec{\mathbf{a}}) = p_2(a, \vec{\mathbf{a}}).$$

Then,

$$f(m, \vec{\mathbf{a}}) = \begin{cases} p_1(\vec{\mathbf{a}}), & \text{if } m = \mathbf{0}^{\mathbf{M}}; \\ p_2(f(m -^{\mathbf{M}} \mathbf{1}^{\mathbf{M}}, \vec{\mathbf{a}}), \vec{\mathbf{a}}), & \text{if } m \neq \mathbf{0}^{\mathbf{M}}, \end{cases}$$

is a polynomial.

Assume that $f(\vec{\mathbf{a}}) = \mu m(f_1(m, \vec{\mathbf{a}}) = \mathbf{0}^{\mathbf{M}})$. Then, $f(\vec{\mathbf{a}}) \cdot a_1^{\mathbf{0}^{\mathbf{M}}}$ is the polynomial for $f(\vec{\mathbf{a}})$. □

**Corollary 1** $\lambda a, b : M.a -^{\mathbf{M}} b, \lambda a : M.2^a$ and $\lambda a, b : M.a^b$ are not **M**-recursive, where $a -^{\mathbf{M}} b = c$ iff $a +^{\mathbf{M}} c = b$ or $a <^{\mathbf{M}} b$. □

Each primitively recursive function is provably recursive, and so is $\lambda n.2^n$. There is a $\Sigma_1$-formula $\exists z\theta(x, y, z)$ such that $PA \vdash \forall x \exists! y \exists z \theta(x, y, z)$. For any nonstandard model **M** of Peano arithmetic, for any $a, b \in M, 2^a = b$ iff $\mathbf{M} \models \exists z\theta(x/a, y/b, z)$, iff there is an element $c \in M$ such that $\mathbf{M} \models \theta(x/a, y/b, z/c)$. If $a, b \in M - N^{\mathbf{M}}$ then $c \notin N^{\mathbf{M}}$. $\exists z\theta(x, y, z)$ defines $\lambda m : N^{\mathbf{M}}.2^m$ in **M**.

## 5 The equivalence of $\mathbf{C}^{\mathbf{M}}$ and $\mathbf{R}^{\mathbf{M}}$

In this section, we prove that each **M**-computable function is **M**-recursive, and conversely, each **M**-recursive function is **M**-computable.

**Theorem 1** $\mathbf{C}^{\mathbf{M}} = \mathbf{R}^{\mathbf{M}}$.

**Proof** Firstly we prove that $\mathbf{C}^{\mathbf{M}} \subseteq \mathbf{R}^{\mathbf{M}}$, that is, for any **M**-computable function $f_\alpha^{\mathbf{M}}$ for some program $\alpha$, $f_\alpha^{\mathbf{M}}$ is **M**-recursive. We prove the claim by induction on $\alpha$.

• Case 1 $\alpha = x := t$. If $t = \mathbf{n}$ then $f_\alpha^{\mathbf{M}}(a) = n$ for any $a \in M$, is **M**-recursive; if $t = x$ then $f_\alpha^{\mathbf{M}}(a) = a$ for any $a \in M$, is **M**-recursive; if $t = t_1 + t_2$ and $f_{\alpha_1}^{\mathbf{M}}$ and $f_{\alpha_2}^{\mathbf{M}}$ are **M**-recursive for $t_1$ and $t_2$, respectively, then $f_\alpha^{\mathbf{M}}(a) = f_{\alpha_1}^{\mathbf{M}}(a) +^{\mathbf{M}} f_{\alpha_2}^{\mathbf{M}}(a)$ is **M**-recursive; similar for $t = t_1 \cdot t_2$.

• Case 2 $\alpha = \alpha_1; \alpha_2$. Assume that $f_{\alpha_1}^{\mathbf{M}}(\vec{\mathbf{a}}), f_{\alpha_2}^{\mathbf{M}}(b, \vec{\mathbf{a}})$ are **M**-recursive. Then,

$$f_{\alpha_1;\alpha_2}^{\mathbf{M}}(\vec{\mathbf{a}}) = f_{\alpha_2}^{\mathbf{M}}(f_{\alpha_1}^{\mathbf{M}}(\vec{\mathbf{a}}), \vec{\mathbf{a}})$$

is **M**-recursive;

• Case 3 $\alpha$ = **if** $\phi$ **then** $\alpha_1$ **else** $\alpha_2$. Assume that $\phi(x_1, \ldots, x_i)$ contains variables $x_1, \ldots, x_i$ without any quantifier. Then, for any $\vec{\mathbf{a}} \in M$,

$$\chi_\phi(\vec{\mathbf{a}}) = \begin{cases} \mathbf{1}^{\mathbf{M}}, & \text{if } \mathbf{M} \models \phi(x_1/a_1, \ldots, x_i/a_i); \\ \mathbf{0}^{\mathbf{M}}, & \text{otherwise,} \end{cases}$$

is **M**-recursive. Assume that $f_{\alpha_1}^{\mathbf{M}}(\vec{\mathbf{a}}), f_{\alpha_2}^{\mathbf{M}}(\vec{\mathbf{a}})$ are **M**-recursive for $\alpha_1$ and $\alpha_2$, respectively. Then,

$$f_\alpha^{\mathbf{M}}(\vec{\mathbf{a}}) = \begin{cases} f_{\alpha_1}^{\mathbf{M}}(\vec{\mathbf{a}}), & \text{if } \chi_\phi(\vec{\mathbf{a}}) = \mathbf{1}^{\mathbf{M}}; \\ f_{\alpha_2}^{\mathbf{M}}(\vec{\mathbf{a}}), & \text{if } \chi_\phi(\vec{\mathbf{a}}) = \mathbf{0}^{\mathbf{M}}, \end{cases}$$
$$= \chi_\phi(\vec{\mathbf{a}}) f_{\alpha_1}^{\mathbf{M}}(\vec{\mathbf{a}}) +^{\mathbf{M}} (\mathbf{1}^{\mathbf{M}} -^{\mathbf{M}} \chi_\phi(\vec{\mathbf{a}})) f_{\alpha_2}^{\mathbf{M}}(\vec{\mathbf{a}})$$

is **M**-recursive;

• Case 4 $\alpha$ = **while** $\phi$ **then** $\alpha_1$. Assume that $f_{\alpha_1}^{\mathbf{M}}(\vec{\mathbf{a}})$ is **M**-recursive for $\alpha_1$. Then, for any $m \in N^M, \vec{\mathbf{a}} \in M^{i-1}$,

$$f_\alpha^{\mathbf{M}}(\vec{\mathbf{a}}) = \begin{cases} f_1^{\mathbf{M}}(m, \vec{\mathbf{a}}), & \text{if } \mathbf{E}m(\mathbf{M} \not\models \phi(x/f_1^{\mathbf{M}}(m, \vec{\mathbf{a}}))); \\ \uparrow, & \text{otherwise,} \end{cases}$$

where

$$f_1^{\mathbf{M}}(m, \vec{a}) = \begin{cases} f_{\alpha_1}^{\mathbf{M}}(\vec{a}), & \text{if } m = \mathbf{0}^{\mathbf{M}}; \\ f_{\alpha_1}^{\mathbf{M}}(f_1^{\mathbf{M}}(m -^{\mathbf{M}} \mathbf{1}^{\mathbf{M}}, \vec{\mathbf{a}}), \vec{\mathbf{a}}), & \text{otherwise.} \end{cases}$$

Because $\mathbf{E}m(\mathbf{M} \not\models \phi(x/f_1^{\mathbf{M}}(m, \vec{\mathbf{a}})))$ and $f_1^{\mathbf{M}}(m, \vec{a})$ are **M**-recursive, $f_\alpha^{\mathbf{M}}(\vec{\mathbf{a}})$ is **M**-recursive.

Then, we prove that $\mathbf{R}^{\mathbf{M}} \subseteq \mathbf{C}^{\mathbf{M}}$, that is, for any **M**-recursive function $f$, there is a program $\alpha$ such that $f_\alpha^{\mathbf{M}} = f$. We prove the claim by induction on the structure of $f$.

• Case 5 $f(\vec{\mathbf{a}}) = n | a_1 +^{\mathbf{M}} a_2 | a_1 \cdot^{\mathbf{M}} a_2$.

If $f(\vec{\mathbf{a}}) = n$ then let $\alpha ::= x := \mathbf{n}$, and $f_\alpha^{\mathbf{M}}(\vec{\mathbf{a}}) = n$;

If $f(\vec{\mathbf{a}}) = a_1 +^{\mathbf{M}} a_2$ then let $\alpha ::= x := x_1 + x_2$, and $f_\alpha^{\mathbf{M}}(\vec{\mathbf{a}}) = a_1 +^{\mathbf{M}} a_2$; and similar for $f(\vec{\mathbf{a}}) = a_1 \cdot^{\mathbf{M}} a_2$.

• Case 6 $f(\vec{\mathbf{a}}) = f_2(f_1(\vec{\mathbf{a}}), \vec{\mathbf{a}})$. Assume that $\alpha_1, \alpha_2$ are such that $f_{\alpha_1}^{\mathbf{M}}(\vec{\mathbf{a}}) = f_1(\vec{\mathbf{a}})$ and $f_{\alpha_2}^{\mathbf{M}}(b, \vec{\mathbf{a}}) = f_2(b, \vec{\mathbf{a}})$. Then, let $\alpha = \alpha_1; \alpha_2$, and

$$f_\alpha^{\mathbf{M}}(\vec{\mathbf{a}}) = f(\vec{\mathbf{a}}).$$

• Case 7 $f(m, \vec{\mathbf{a}}) = \begin{cases} f_1(\vec{\mathbf{a}}), & \text{if } m = \mathbf{0}^{\mathbf{M}}; \\ f_2(f(m -^{\mathbf{M}} \mathbf{1}^{\mathbf{M}}, \vec{\mathbf{a}}), m, \vec{\mathbf{a}}), & \text{otherwise.} \end{cases}$

Assume that $\alpha_1, \alpha_2$ are such that $f_{\alpha_1}^{\mathbf{M}}(\vec{\mathbf{a}}) = f_1(\vec{\mathbf{a}}), f_{\alpha_2}^{\mathbf{M}}(b, m, \vec{\mathbf{a}}) =$

$f_2(b, m, \vec{\mathbf{a}})$. Define

$$\alpha ::= \textbf{if } m = 0 \textbf{ then } \alpha_1, \quad \textbf{else } (y := 0; (\textbf{while } y \neq m \textbf{ do } \alpha_2; y := y + 1)).$$

Then, $f^{\mathbf{M}}_{\alpha}(m, \vec{\mathbf{a}}) = f(m, \vec{\mathbf{a}})$.

• Case 8 $f(\vec{\mathbf{a}}) = \mu m \in N^M(f_1(m, \vec{\mathbf{a}}) = \mathbf{0}^{\mathbf{M}})$. Assume that $\alpha_1$ is such that $f^{\mathbf{M}}_{\alpha_1}(m, \vec{\mathbf{a}}) = f_1(m, \vec{\mathbf{a}})$. Define

$$\alpha ::= y := 0; \alpha_1; (\textbf{while } x \neq 0 \textbf{ do } y := y + 1; \alpha_1).$$

Then,

$$f^{\mathbf{M}}_{\alpha}(\vec{\mathbf{a}}) = f(\vec{\mathbf{a}}).$$

Assume that $m = \mu m \in N^M(f_1(m, \vec{\mathbf{a}}) = \mathbf{0}^{\mathbf{M}})$. Then, when $y = m, x = f_1(m, \vec{\mathbf{a}}) = \mathbf{0}^{\mathbf{M}}$ and $f^{\mathbf{M}}_{\alpha}(\vec{\mathbf{a}}) = m$ (the value of $y$); and if $f^{\mathbf{M}}_{\alpha}(\vec{\mathbf{a}}) = m$, then $f_1(m, \vec{\mathbf{a}}) = \mathbf{0}^{\mathbf{M}}$ and because $m$ is the least one, $f(\vec{\mathbf{a}}) = m$. □

## 6 The $\mathbf{M}^-$-recursive functions

In this section, we show that if the logical language $L$ is restrained to $L^- = \{\mathbf{s}, \mathbf{0}, \mathbf{1}, =\}$, where $\mathbf{s}$ is the successor operator, then for any nonstandard model $\mathbf{M}$ of $L$, $+^{\mathbf{M}}$ and $\cdot^{\mathbf{M}}$ are not $\mathbf{M}^-$-recursive.

**Definition 2** A function $f : M \to M$ is $\mathbf{M}^-$-recursive if either

1) $f$ is elementary, that is,

$$f(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = n|m_1|(m +^{\mathbf{M}} \mathbf{1}^{\mathbf{M}})|a_1|a_1 +^{\mathbf{M}} \mathbf{1}^{\mathbf{M}},$$

where $n \in N^M$;

2) (Composition) there are $\mathbf{M}^-$-recursive functions $f_1(\vec{\mathbf{m}}; \vec{\mathbf{a}}) : (N^{\mathbf{M}})^i \times M^j \to M$ and $f_2(\vec{\mathbf{m}}; \vec{\mathbf{a}}, a) : (N^{\mathbf{M}})^i \times M^{j+1} \to M$ such that for any $\vec{\mathbf{m}} \in (N^{\mathbf{M}})^i$, and $\vec{\mathbf{a}} \in M^j$,

$$f(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = f_2(\vec{\mathbf{m}}; \vec{\mathbf{a}}, f_1(\vec{\mathbf{m}}, \vec{\mathbf{a}}));$$

there are $\mathbf{M}^-$-recursive functions $f_1(\vec{\mathbf{m}}; \vec{\mathbf{a}}) : (N^{\mathbf{M}})^i \times M^j \to N^{\mathbf{M}}$ and $f_2(\vec{\mathbf{m}}, m; \vec{\mathbf{a}})$ such that for any $\vec{\mathbf{m}} \in (N^{\mathbf{M}})^i, m \in N^{\mathbf{M}}$ and $\vec{\mathbf{a}} \in M^j$,

$$f(\vec{\mathbf{m}}, m; \vec{\mathbf{a}}) = f_2(\vec{\mathbf{m}}, f_1(\vec{\mathbf{m}}; \vec{\mathbf{a}}); \vec{\mathbf{a}});$$

3) (Recursion) there are $\mathbf{M}^-$-recursive functions $f_1(\vec{\mathbf{m}}; \vec{\mathbf{a}}) : (N^{\mathbf{M}})^i \times M^j \to M$, $f_2(\vec{\mathbf{m}}; a, \vec{\mathbf{a}}) : (N^{\mathbf{M}})^i \times M^{j+1} \to M$ such that for any $\vec{\mathbf{m}} \in (N^{\mathbf{M}})^i, m \in N^M$ and $\vec{\mathbf{a}} \in M^j$,

$$\begin{cases} f(\mathbf{0}^{\mathbf{M}}, \vec{\mathbf{m}}; \vec{\mathbf{a}}) = f_1(\vec{\mathbf{m}}; \vec{\mathbf{a}}); \\ f(m +^{\mathbf{M}} \mathbf{1}^{\mathbf{M}}, \vec{\mathbf{m}}; \vec{\mathbf{a}}) = f_2(\vec{\mathbf{m}}; f(m, \vec{\mathbf{m}}; \vec{\mathbf{a}}), \vec{\mathbf{a}}); \end{cases}$$

or

4) ($\mu$-operator) there is a total $\mathbf{M}^-$-recursive function $f_1(m, \vec{\mathbf{m}}; \vec{\mathbf{a}}) : (N^{\mathbf{M}})^{i+1} \times M^j \to M$ such that for any $\vec{\mathbf{m}} \in (N^{\mathbf{M}})^i$ and $\vec{\mathbf{a}} \in M^j$,

$$f(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = \mu m \in N^M(f_1(m, \vec{\mathbf{m}}; \vec{\mathbf{a}}) = \mathbf{0}^{\mathbf{M}}).$$

Let $\mathbf{R}^{\mathbf{M}^-}$ be the set of all the $\mathbf{M}^-$-recursive functions.

**Theorem 2** For any $\mathbf{M}^-$-recursive function $f : (N^{\mathbf{M}})^i \times M^j \to M$, there is an $\mathbf{M}^-$-recursive function $f' : (N^{\mathbf{M}})^i \times M \to M$ such that

$$f = f'.$$

**Proof** We prove the theorem by induction on the definition of $f$.

For the elementary $\mathbf{M}^-$-recursive functions, the claim is clear.

Assume that $f(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = f_2(\vec{\mathbf{m}}, \vec{\mathbf{a}}, f_1(\vec{\mathbf{m}}, \vec{\mathbf{a}}))$, $f_2(\vec{\mathbf{m}}, \vec{\mathbf{a}}, a) = f'_2(\vec{\mathbf{m}}, a)$ and $f_1(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = f'_1(\vec{\mathbf{m}}; a_1)$. Then, let $f'(\vec{\mathbf{m}}; a) = f_2(\vec{\mathbf{m}}; f_1(\vec{\mathbf{m}}; a))$, and

$$f(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = f'(\vec{\mathbf{m}}; a).$$

Assume that $f(m, \vec{\mathbf{m}}; \vec{\mathbf{a}})$ is defined by the recursion and $f_1(\vec{\mathbf{m}}; \vec{\mathbf{a}}), f_2(\vec{\mathbf{m}}; f(m', \vec{\mathbf{m}}; \vec{\mathbf{a}}), \vec{\mathbf{a}})$. By the induction assumption, let $f'_1(\vec{\mathbf{m}}; a)$ and $f'_2(\vec{\mathbf{m}}; a)$ be the functions such that

$$f_1(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = f'_1(\vec{\mathbf{m}}; a_1);$$
$$f_2(\vec{\mathbf{m}}; a, \vec{\mathbf{a}}) = f'_2(\vec{\mathbf{m}}; a).$$

Let

$$f'(m, \vec{\mathbf{m}}; a) = \begin{cases} f'_1(\vec{\mathbf{m}}; a), & \text{if } m = \mathbf{0}^{\mathbf{M}}; \\ f'_2(\vec{\mathbf{m}}; f'(m', \vec{\mathbf{m}}; a)), & \text{if } m = m' +^{\mathbf{M}} \mathbf{1}^{\mathbf{M}}. \end{cases}$$

Then,

$$f(m, \vec{\mathbf{m}}; \vec{\mathbf{a}}) = f'(m, \vec{\mathbf{m}}; a_1).$$

Assume that $f(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = \mu m(g(m, \vec{\mathbf{m}}; \vec{\mathbf{a}}) = \mathbf{0}^{\mathbf{M}})$, and $g(m, \vec{\mathbf{m}}; \vec{\mathbf{a}}) = g'(m, \vec{\mathbf{m}}; a_1)$. Let $f'(\vec{\mathbf{m}}; a) = \mu m(g'(m, \vec{\mathbf{m}}; a) = \mathbf{0}^{\mathbf{M}})$. Then,

$$f(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = f'(\vec{\mathbf{m}}; a_1).$$ □

The theorem shows that each $\mathbf{M}^-$-recursive function has at most one parameter in $\mathbf{M}$; and may have several parameters in $\mathbf{N}^{\mathbf{M}}$.

**Corollary 2** $+^{\mathbf{M}}$ and $\cdot^{\mathbf{M}}$ are not $\mathbf{M}^-$-recursive.

**Proof** Because $\lambda a_1, a_2.a_1 +^{\mathbf{M}} a_2$ is a function with two parameters in $\mathbf{M}$, there is no $\mathbf{M}^-$-recursive function $f'(\vec{\mathbf{m}}; a)$ such that

$$f'(\vec{\mathbf{m}}; a) = a_1 +^{\mathbf{M}} a_2.$$

Similarly, there is no $\mathbf{M}^{-}$-recursive function $f'(\vec{\mathbf{m}}; a)$ such that

$$f'(\vec{\mathbf{m}}; a) = a_1 \cdot^{\mathbf{M}} a_2. \qquad \square$$

Tennenbaum's theorem says that for any nonstandard model $\mathbf{M}$ of Peano arithmetic, the addition $+^{\mathbf{M}}$ and multiplication $\cdot^{\mathbf{M}}$ are not recursive. Precisely, given a nonstandard model $\mathbf{M}$ of Peano arithmetic, let $\mathbf{N}$ be the standard model of Peano arithmetic such that $N = M$. Then, $+^{\mathbf{M}}$ and $\cdot^{\mathbf{M}}$, as functions on $\mathbf{N}$, are not recursive in $\mathbf{N}$.

Theorem 2 shows that in the logical language $L^{-}$, $+^{\mathbf{M}}$ and $\cdot^{\mathbf{M}}$ are not $\mathbf{M}^{-}$-recursive. For the standard model $\mathbf{N}$ of Peano arithmetic, $+^{\mathbf{N}}$ and $\cdot^{\mathbf{N}}$ are $\mathbf{N}^{-}$-recursive, where the $\mathbf{N}^{-}$-recursiveness is equivalent to the $\mathbf{N}$-recursiveness.

## 7 The $\Sigma_1^{\mathbf{N}^{\mathbf{M}}}$-definability of the M-recursive functions

In the first-order theory of Peano arithmetic, for any nonstandard model $\mathbf{M}$ of Peano arithmetic, there is an $\mathbf{M}$-recursive function $f_\alpha^{\mathbf{M}}$ such that $f_\alpha^{\mathbf{M}}$ is not arithmetical, i.e., not definable in $\mathbf{M}$. Let

$$\alpha(x) ::= y := 0; \textbf{while } y \neq x \textbf{ do } y =: y + 1.$$

Then, $f_\alpha^{\mathbf{M}}$ is defined as follows: for any $a \in M$,

$$f_\alpha^{\mathbf{M}}(a) = a \text{ iff } a \in N^M.$$

Because $N^M$ is not arithmetical in $\mathbf{M}$, $f_\alpha^{\mathbf{M}}$ is not arithmetical in $\mathbf{M}$.

We change the first-order logic for Peano arithmetic so that $f_\alpha^{\mathbf{M}}$ is definable in $\mathbf{M}$.

The logical language $L^{+}$ contains the following symbols:

- constant symbols: $\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}, \ldots$;
- number variables: $\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2, \ldots$;
- variables: $x_0, x_1, x_2, \ldots$;
- binary function symbols: $+, \cdot$;
- binary predicate symbol: $<$;
- logical connectives and quantifiers: $\neg, \to, \forall$.

Terms are defined as follows

$$t ::= \mathbf{n}|\mathbf{m}|x|t_1 + t_2|t_1 \cdot t_2.$$

Formulas are defined as follows:

$$\phi ::= t_1 < t_2|\neg\phi_1|\phi_1 \to \phi_2|\forall\mathbf{m}\phi_1(\mathbf{m}).$$

A model $\mathbf{M}$ is a pair $(M, +^{\mathbf{M}}, \cdot^{\mathbf{M}}, <^{\mathbf{M}}, \mathbf{0}^{\mathbf{M}}, \mathbf{1}^{\mathbf{M}}, I)$, where

- $M$ is the universe;
- $I(+) = +^{\mathbf{M}}, I(\cdot) = \cdot^{\mathbf{M}}, I(<) = <^{\mathbf{M}}$;
- $I(\mathbf{0}) = \mathbf{0}^{\mathbf{M}}, I(\mathbf{1}) = \mathbf{1}^{\mathbf{M}} \in M$.

Let

$$N^M = \{I(\mathbf{n}) : n \in N\}.$$

An assignment $v$ is a function from variables to $M$, and a number assignment $w$ is a function from number variables to $N^M$.

The interpretation of $t$ at $(\mathbf{M}, v, w)$, denoted by $t^{\mathbf{M},v,w}$, is defined as follows:

$$t^{\mathbf{M},v,w} = \begin{cases} I(\mathbf{n}), & \text{if } t = \mathbf{n}; \\ w(\mathbf{m}), & \text{if } t = \mathbf{m}; \\ v(x), & \text{if } t = x; \\ t_1^{\mathbf{M},v,w} +^{\mathbf{M}} t_2^{\mathbf{M},v,w}, & \text{if } t = t_1 + t_2; \\ t_1^{\mathbf{M},v,w} \cdot^{\mathbf{M}} t_2^{\mathbf{M},v,w}, & \text{if } t = t_1 \cdot t_2. \end{cases}$$

A formula $\phi$ is satisfied at $(\mathbf{M}, v, w)$, denoted by $(\mathbf{M}, v, w) \models \phi$, if

$$\begin{cases} t_1^{\mathbf{M},v,w} <^{\mathbf{M}} t_2^{\mathbf{M},v,w}, & \text{if } \phi = t_1 < t_2; \\ (\mathbf{M}, v, w) \not\models \phi_1, & \text{if } \phi = \neg\phi_1; \\ (\mathbf{M}, v, w) \models \phi_1 \Rightarrow (\mathbf{M}, v, w) \models \phi_2, & \text{if } \phi = \phi_1 \to \phi_2; \\ \mathbf{A}n \in N^M((\mathbf{M}, v, w(\mathbf{m}/n)) \models \phi_1(\mathbf{m})), & \text{if } \phi = \forall\mathbf{m}\phi(\mathbf{m}). \end{cases}$$

**Proposition 4** $N^{\mathbf{M}}$ is definable in $\mathbf{M}$.

**Proof** Let $\phi(x) = \exists\mathbf{m}(x = \mathbf{m})$. Then,

$$N^{\mathbf{M}} = \{a \in M : \mathbf{M} \models \exists\mathbf{m}(x/a = \mathbf{m})\}. \qquad \square$$

**Remark** $L^{+}$ is different from $L$. Robinson's overspill lemma holds in $L$ and does not in $L^{+}$. $\square$

**Theorem 3** For any $\mathbf{M}$-recursive function $f : (N^{\mathbf{M}})^i \times M^j \to M$, there are recursive functions $g_1, \ldots, g_k : (N^{\mathbf{M}})^i \to N^{\mathbf{M}}$ such that for any $\vec{\mathbf{m}} \in (N^{\mathbf{M}})^i$ and $\vec{\mathbf{a}} \in M^j$,

$$f(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = \sum_{h=1}^{k} g_h(\vec{\mathbf{m}}) a_1^{i_{h1}} \cdots a_j^{i_{hj}}.$$

**Proof** We prove by induction on the recursiveness definition of $f$.

If $f$ is elementary then the theorem holds for $f$.

Assume that $f(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = f_1(\vec{\mathbf{m}}; f_2(\vec{\mathbf{m}}; \vec{\mathbf{a}}), \vec{\mathbf{a}}))$ and the theorem holds for $f_1$ and $f_2$. Let

$$f_1(\vec{\mathbf{m}}; a, \vec{\mathbf{a}}) = \textstyle\sum_{1 \leqslant h \leqslant k} g_h(\vec{\mathbf{m}}) a_1^{i_{h1}} \cdots a_j^{i_{hj}} \cdot a^{i_h};$$
$$f_2(\vec{\mathbf{m}}; \vec{\mathbf{a}}) = \textstyle\sum_{1 \leqslant h' \leqslant k'} h_{h'}(\vec{\mathbf{m}}) a_1^{i_{h'1}} \cdots a_j^{i_{h'j}}.$$

Then,

$$
\begin{aligned}
f(\vec{\mathbf{m}};\vec{\mathbf{a}}) &= \sum_{1\leqslant h\leqslant k} g_h(\vec{\mathbf{m}})a_1^{i_{h1}}\cdots a_j^{i_{hj}}\left(\sum_{1\leqslant h'\leqslant k'} h_{h'}(\vec{\mathbf{m}})a_1^{j_{h'1}}\cdots a_j^{j_{h'j}}\right)^{i_h}\\
&= \sum_{1\leqslant h\leqslant k}\sum_{1\leqslant h''\leqslant k''} g_h(\vec{\mathbf{m}})h_{h''}(\vec{\mathbf{m}})a_1^{i_{h1}+i_{h''1}}\cdots a_j^{i_{hj}+j_{h''j}}\\
&= \sum_{1\leqslant h'''\leqslant k'''} g'_{h'''}(\vec{\mathbf{m}})a_1^{j_{1h'''}}\cdots a_j^{j_{jh'''}},
\end{aligned}
$$

where

$$
\left(\sum_{1\leqslant h'\leqslant k'} h_{h'}(\vec{\mathbf{m}})a_1^{j_{h'1}}\cdots a_j^{j_{h'j}}\right)^{i_h} = \sum_{1\leqslant h''\leqslant k''} h_{h''}(\vec{\mathbf{m}})a_1^{j_{1h''}}\cdots a_j^{j_{jh''}},
$$

and for each $h''' \leqslant k'''$, $g'_{h'''}(\vec{\mathbf{m}})$ is recursive,.

Assume that $f$ is defined by the natural recursion and $f_1, f_2$; and the proposition holds for $f_1$ and $f_2$. Let

$$
\begin{aligned}
f_1(\vec{\mathbf{m}};\vec{\mathbf{a}}) &= \textstyle\sum_{1\leqslant h\leqslant k} g_h(\vec{\mathbf{m}})a_1^{i_{1h}}\cdots a_j^{i_{jh}};\\
f_2(\vec{\mathbf{m}};a,\vec{\mathbf{a}}) &= \textstyle\sum_{1\leqslant h'\leqslant k'} h_h(\vec{\mathbf{m}})a_1^{j_{1h'}}\cdots a_j^{j_{jh}}\cdot a^{j_{h'}}.
\end{aligned}
$$

Then, by the induction on $m$,

$$
f(\mathbf{0^M},\vec{\mathbf{m}};\vec{\mathbf{a}}) = \sum_{1\leqslant h\leqslant k} g_h(\vec{\mathbf{m}})a_1^{i_{1h}}\cdots a_j^{i_{jh}};
$$

$$
\begin{aligned}
&f(m +^{\mathbf{M}} \mathbf{1^M},\vec{\mathbf{m}};\vec{\mathbf{a}})\\
&=\textstyle\sum_{1\leqslant h'\leqslant k'} h_{h'}(\vec{\mathbf{m}})a_1^{j_{1h'}}\cdots a_j^{j_{jh'}}\cdot(f(m,\vec{\mathbf{m}};\vec{\mathbf{a}}))^{j_{h'}}\\
&=\textstyle\sum_{1\leqslant h'\leqslant k'} h_{h'}(\vec{\mathbf{m}})a_1^{j_{1h'}}\cdots a_j^{j_{jh'}}\cdot\left(\sum_{1\leqslant h''\leqslant k''} h_{h''}(\vec{\mathbf{m}})a_1^{j_{1h''}}\cdots a_j^{j_{jh''}}\right)^{j_{h'}}\\
&=\textstyle\sum_{1\leqslant h'\leqslant k'}\sum_{1\leqslant h'''\leqslant k'''} h_{h'}(\vec{\mathbf{m}})h_{h'''}(\vec{\mathbf{m}})a_1^{j_{1h'}+j_{1h'''}}\cdots a_j^{j_{jh'}+j_{jh'''}}\\
&=\textstyle\sum_{1\leqslant h^{(4)}\leqslant k^{(4)}} g'_{h^{(4)}}(\vec{\mathbf{m}})a_1^{j_{1h^{(4)}}}\cdots a_j^{j_{jh^{(4)}}}
\end{aligned}
$$

is a polynomial, where by the induction assumption,

$$
(f(m,\vec{\mathbf{m}};\vec{\mathbf{a}}))^{j_{h'}} = \sum_{1\leqslant h''\leqslant k''} h_{h''}(\vec{\mathbf{m}})a_1^{j_{1h''}}\cdots a_j^{j_{jh''}},
$$

and $h_{h''}(\vec{\mathbf{m}})$ and $g'_{h^{(4)}}(\vec{\mathbf{m}})$ are recursive.

Assume that $f$ is defined by the minimalization operator and $f_1$, and the proposition holds for $f_1$. Let

$$
f_1(m,\vec{\mathbf{m}};\vec{\mathbf{a}}) = \sum_{1\leqslant h'\leqslant k'} h_h(\vec{\mathbf{m}})a_1^{j_{1h'}}\cdots a_j^{j_{jh}}.
$$

Then,

$$
\begin{aligned}
f(\vec{\mathbf{m}};\vec{\mathbf{a}}) &= \mu m(f_1(m,\vec{\mathbf{m}};\vec{\mathbf{a}}) = \mathbf{0^M})\\
&= \begin{cases} \mu m(f_1(m,\vec{\mathbf{m}};\vec{\mathbf{n}}) = \mathbf{0^M}), & \text{if } \vec{\mathbf{a}} = \vec{\mathbf{n}} \text{ for some } \vec{\mathbf{n}} \in (N^{\mathbf{M}})^j;\\ \text{undefined}, & \text{otherwise.}\end{cases}
\end{aligned}
$$

Because for any $\vec{\mathbf{m}} \in (N^{\mathbf{M}})^i$ and $\vec{\mathbf{a}} \in M^j, f(\vec{\mathbf{m}};\vec{\mathbf{a}}) = \sum_{h=1}^k g_h(\vec{\mathbf{m}})a_1^{i_{h1}}\cdots a_j^{i_{hj}}$, we have

$$
f(\vec{\mathbf{m}};\vec{\mathbf{a}}) \in N^{\mathbf{M}} \text{ iff } \vec{\mathbf{a}} \in (N^{\mathbf{M}})^i. \qquad \square
$$

**Corollary 3** $\lambda a, b : M.\ a \ominus^{\mathbf{M}} b = \begin{cases} a -^{\mathbf{M}} b & \text{if } b \leqslant^{\mathbf{M}} a\\ \mathbf{0^M} & \text{otherwise}\end{cases}$ is not **M**-recursive, where

$$
a -^{\mathbf{M}} b = c \text{ iff } b +^{\mathbf{M}} c = a. \qquad \square
$$

**Corollary 4** For any **M**-recursive function $f(\vec{\mathbf{m}};\vec{\mathbf{a}})$, $\text{range}(f(\vec{\mathbf{m}};\vec{\mathbf{a}})) \cap N^{\mathbf{M}} = \text{range}(f(\vec{\mathbf{m}};\vec{\mathbf{m}}'))$. □

**Definition 3** A set $A \subseteq M$ is **M**-recursively enumerable if there is an **M**-recursive function $f(\vec{\mathbf{m}};\vec{\mathbf{a}})$ such that

$$
\text{range}(f(\vec{\mathbf{m}};\vec{\mathbf{a}})) = A.
$$

**Corollary 5** For any set $A \subseteq N^{\mathbf{M}}, A$ is **M**-recursively enumerable if and only if $A$ is recursively enumerable. □

**Corollary 6** For any nonstandard number $a \in M$ and any subset $A \subseteq [a]^{\mathbf{M}} = \{a +^{\mathbf{M}} n : n \in N^{\mathbf{M}}\}, A$ is **M**-recursively enumerable if and only if there is a recursively enumerable set $B \subseteq N^{\mathbf{M}}$ such that $B + a = \{n +^{\mathbf{M}} a : n \in B\} = A$. □

**Theorem 4** For any **M**-recursive function $f(\vec{\mathbf{m}};\vec{\mathbf{a}}), \{(\vec{\mathbf{m}};\vec{\mathbf{a}};b) : f(\vec{\mathbf{m}};\vec{\mathbf{a}}) = b\}$ is $\Sigma_1^{\mathbf{N^M}}$-definable in **M**.

**Proof** For any $\vec{\mathbf{m}} \in (N^{\mathbf{M}})^i, b \in M$ and $\vec{\mathbf{a}} \in M^j$, we have

$$
\begin{aligned}
&f(\vec{\mathbf{m}};\vec{\mathbf{a}}) = b,\\
\text{iff } & b = \sum_{h=1}^{k} g_h(\vec{\mathbf{m}})a_1^{i_{h1}}\cdots a_j^{i_{hj}},\\
\text{iff } & \mathbf{A}n_1,\ldots,n_k\left(\bigwedge_{h=1}^{k}\phi^h(\vec{\mathbf{m}},n_h) \Rightarrow b = \sum_{h=1}^{k} n_h a_1^{i_{h1}}\cdots a_j^{i_{hj}}\right),\\
\text{iff } & \mathbf{M} \models \forall \mathbf{n}_1,\ldots,\mathbf{n}_k\Bigg(\bigwedge_{h=1}^{k}\phi^h(\vec{\mathbf{m}},\mathbf{n}_h) \rightarrow y/b\\
&\quad = \sum_{h=1}^{k} n_h(x_1/a_1)^{i_{h1}}\cdots(x_j/a_j)^{i_{hj}}\Bigg),\\
\text{iff } & \mathbf{E}n_1,\ldots,n_k\left(\bigwedge_{h=1}^{k}\phi^h(\vec{\mathbf{m}},n_h) \wedge b = \sum_{h=1}^{k} n_h a_1^{i_{h1}}\cdots a_j^{i_{hj}}\right),\\
\text{iff } & \mathbf{M} \models \exists \mathbf{n}_1,\ldots,\mathbf{n}_k\Bigg(\bigwedge_{h=1}^{k}\phi^h(\vec{\mathbf{m}},\mathbf{n}_h) \wedge y/b\\
&\quad = \sum_{h=1}^{k} n_h(x_1/a_1)^{i_{h1}}\cdots(x_j/a_j)^{i_{hj}}\Bigg). \qquad \square
\end{aligned}
$$

**Corollary 7** For any **M**-recursive function $f$, there is a quantifier-free formula $\phi(\mathbf{m}, x_1,\ldots,x_i,y)$ such that for any $a_1,\ldots,a_i,b \in M$,

$$
f(a_1,\ldots,a_i) = b \text{ iff } \mathbf{M} \models \exists \mathbf{m}\phi(\mathbf{m}, x_1/a_1,\ldots,x_i/a_i,y/b). \qquad \square
$$

## 8 Conclusions

The classical recursion theory was generalized to the $\alpha$-recursion theory for admissible ordinals $\alpha$, the $\beta$-recursion theory, etc. However, the recursion theory has not been generalized to be on any nonstandard model of Peano arithmetic, because a nonstandard model of Peano arithmetic is not well-ordered.

The **M**-recursive functions are the recursive functions generalized to a structure (nonstandard model) **M** with substructure $\mathbf{N^M}$. Hence, strictly speaking, the **M**-recursive functions are the $(\mathbf{M}, \mathbf{N^M})$-recursive functions, where the recursion and the $\mu$-operator are applied with respect to the elements in $\mathbf{N^M}$, not with respect to any element in **M**.

We extend the logical language $L$ of Peano arithmetic to $L^+$, where $L^+$ has the variables for natural numbers and the quantifier restrained to be on the standard part of **M**. With the uniformity of the definability of **M**-recursive functions with respect to nonstandard models **M** of Peano arithmetic, we could prove that each specification $\phi[\alpha]\psi$ is reducible to some formula $\theta$ in $L^+$ such that $HL(PA) \vdash \phi[\alpha]\psi$ if and only if $\theta$ is provable in $L^+$, and hence, Hoare logic is complete in this semantics.

**Acknowledgements** This work was supported by the National Natural Science Foundation of China (Grant Nos. 91224006, 61035004, 61173063 and 61203284).

## References

1. Hoare C A R. An axiomatic basis for computer programming. Communications of the ACM, 1969, 12(10), 576–583
2. Apt K R. Ten years of Hoare's logic: a survey-part 1. ACM Transactions on Programming Languages and Systems, 1981, 3(4), 431–483
3. Bergstra J A, Tucker J V. Expressiveness and the completeness of Hoare's logic. Journal of Computer and System Science, 1982, 25(3), 267–284
4. Kozen D, Tiuryn J. On the completeness of propositional Hoare logic. Information Science, 2001, 139(3), 187–195
5. Lamport L, Schneider F B. The "Hoare logic" of CSP, and all that. ACM Transactions on Programming Languages and Systems, 1984, 6(2), 281–296
6. O'Hearn P W, Yang H, Reynolds J C. Separation and information hiding. In: Proceedings of the 31st ACM SIGPLAN-SIGACT symposium on Principles of programming languages. 2004, 268–280
7. Parkinson M, Bierman G. Separation logic and abstraction. ACM SIGPLAN Notices, 2005, 40(1): 247–258
8. Reynolds J C. Separation logic: a logic for shared mutable data structures. In: Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science. 2002, 55–74
9. Cook S A. Soundness and completeness of an axiom system for program verification. SIAM Journal on Computing, 1978, 7(1): 70–90
10. Rogers H. Theory of Recursive Functions and Effective Computability. New York: McGraw-Hill, 1967
11. Kaye R. Models of Peano Arithmetic. Oxford: Oxford University Press, 1991

Cungen Cao is a professor in the Institute of Computing Technology, Chinese Academy of Sciences, China. His main research interest is knowledge representation.

Yuefei Sui is a professor in the Institute of Computing Technology, Chinese Academy of Sciences, China. His main interests include knowledge representation, applied logic and the theory of computability.

Zaiyue Zhang is a professor in Jiangsu University of Sciences and Technology, China. His main interests include applied logic and computer theory.