DEMONSTRATIO MATHEMATICA
Vol. IX No 4 1976

Ireneusz Nabiałek

DETERMINISTIC COMPUTABILITY

A number of approaches to computation theory has been developed which base on characterizing a computer by the set of its possible computation runs (Pawlak [1], Kwasowiec [2]). It has been remarked also that the same idea applies to computers acting in a continuous way (Konikowska [4], Żakowski [5]). The aim of this paper is to examine those general properties of sets of computation runs of computers (computable sets), which are valid for all types of computers. To this end a general time set is introduced to represent various time axis and computable sets are defined as injective ones and closed under shifts along time axis. The injectivity means that any computation of a computer is uniquely determined by some its initial part. The closedness means that together with every computation all its parts corresponding to right segments of the time axis, when shifted to the origin of the time axis, are computations as well. In this paper it is shown that such general computable sets have basic properties similar to those discussed in [1], [2], [3], [4], [5] and [6].

1. A general time set

Let T be an arbitrary infinite set and $\leqslant$ an ordering relation (linear) in T. The pair $(T, \leqslant)$ is called an ordered set. For any element $a \in T$ by T_a we denote the set $\{t \in T : a \leqslant t\}$. The relation $\leqslant$ restricted to any set $Q \subseteq T$ we denote also by $\leqslant$.

D e f i n i t i o n 1.1. An ordered set $(T, \leqslant)$ is called a homogenously ordered set iff for any element $a \in T$ the ordered set $(T_a, \leqslant)$ is similar to $(T, \leqslant)$.

Every homogenously ordered set $(T, \leqslant)$ has a first element, which we denote by 0.

Let A be the set of all similarity mappings α (an isomorphism) such that $\alpha(T) = T_a$ for some $a \in T$.

Let $\alpha \circ \beta$ be the composition of the functions α and β such that β is the first mapping.

D e f i n i t i o n 1.2. Let $(T, \leqslant)$ be an ordered set. Any function $\Theta : T \to A$ such that

$$\forall_{a \in T} \left[\Theta(a) = \alpha \Longrightarrow \alpha(T) = T_a\right], \tag{1.1}$$

$$\forall_{\alpha, \beta \in \Theta(T)} \left[(\alpha \circ \beta) \in \Theta(T)\right] \tag{1.2}$$

is called a function of movements in $(T, \leqslant)$.

D e f i n i t i o n 1.3. A monotonously ordered set $(T, \leqslant)$ is called a uniformly ordered set iff there exists a function of movements in $(T, \leqslant)$.

D e f i n i t i o n 1.4. A triple $(T, \leqslant, \Theta)$ is called a general time set iff $(T, \leqslant)$ is an uniformly ordered set and Θ is a function of movements in $(T, \leqslant)$.

By ι we denote the identity mapping.

T h e o r e m 1.1. If $(T, \leqslant, \Theta)$ is a general time set, then $\Theta(0) = \iota$.

P r o o f. Let $\Theta(0) = \alpha$. By (1.2) we have $\alpha \circ \alpha \in \Theta(T)$. Since $\alpha(0) = 0$ and $\alpha[\alpha(0)] = 0$, we have $\alpha \circ \alpha = \alpha$ by Definition 1.2. Let $a \in T$ and $\alpha(a) = b$. Since $\alpha \circ \alpha = \alpha$, we have $\alpha[\alpha(a)] = \alpha(b) = \alpha(a)$, hence $a = b$. For every $a \in T$ we have $\alpha(a) = a$, thus $\alpha = \iota$.

2. Partial functions

Let X, Y, Y_* be non-empty sets and $Y \subseteq Y_*$. By Y_*^X we denote the set of all functions $f : X \to Y$.

D e f i n i t i o n 2.1. For any functions $f, g \in Y_*^X$ we define a relation $\underline{\underline{Y}}$ as follows:

$$(2.1)\quad (f \overset{Y}{=} g) \Longleftrightarrow \forall_{p \in X} \left\{ [f(p) \in Y \vee g(p) \in Y] \Leftrightarrow [f(p) = g(p)] \right\}.$$

The relation $\overset{Y}{=}$ is an equivalence relation in Y_*^X.

D e f i n i t i o n 2.2. For any $f \in Y_*^X$ the equivalence class $[f]_{\underline{\underline{Y}}}$ is called a partial function from X to Y.

We denote by $f : X \dot{\rightarrow} Y$ the partial function $[f]_{\underline{\underline{Y}}} \in Y_*^X/_{\underline{\underline{Y}}}$.

D e f i n i t i o n 2.3. A function $f : X \rightarrow Y$ is called the empty function iff $\forall_{p \in X} [f(p) \notin Y]$.

The empty function is denoted by σ.

For any function $f : X \rightarrow Y_*$ and for any set $Q \subseteq X$ we denote by $f|Q$ the restriction of the function f to Q.

D e f i n i t i o n 2.4. For any $[f]_{\underline{\underline{Y}}} \in Y_*^X/_{\underline{\underline{Y}}}$ the equivalence class $[f|Q]_{\underline{\underline{Y}}}$ is called the restriction of the partial function $[f]_{\underline{\underline{Y}}}$ to Q.

3. Informative functions

Let $(T, \leqslant, \Theta)$ be a general time set and X, Y - arbitrary non-empty sets.

D e f i n i t i o n 3.1. A function $\tilde{\alpha} : T \times X \rightarrow T_a \times X$ is called an extension of a similarity mapping $\alpha \in \Theta(T)$ on $T \times X$ iff

$$(3.1)\quad \forall_{t \in T} \forall_{x \in X} \tilde{\alpha}(t, x) = (\alpha(t), x).$$

We denote by $\mathcal{F}$ the set of all partial functions $f : T \times X \dot{\rightarrow} Y$.

D e f i n i t i o n 3.2. Any function $f|Z$, where $f \in \mathcal{F}$ and $Z \subseteq T \times X$, is called an informative function with domain of information Z.

For any $a \in T$ and $f \in \mathcal{F}$ we write $f_a = f \circ \tilde{\alpha}$, where $\alpha \in \Theta(T)$ and $\alpha(0) = a$. For any $a \in T$ and $f \in \mathcal{F}$ the function $f_a|Z$ is an informative function with domain of information Z.

4. Computable sets

Assume that the following sets are arbitrary but fixed: a general time set $(T, \leqslant, \Theta)$, non-empty sets X, Y and a domain of information $Z \subseteq T \times X$. Hence the following system is fixed $(T, \leqslant, \Theta, X, Y, Z)$.

Definition 4.1. A set $F \subseteq \mathcal{F}$ is called a Z-injective set iff

$$(4.1) \qquad \bigvee_{f,g \in F} \left[(f|Z = g|Z) \Rightarrow (f = g) \right].$$

Definition 4.2. For any set $F \subseteq \mathcal{F}$ by the $*$-closure of F we mean the set $F^* = \{\varphi : \varphi = f_a$ for some $f \in F$ and $a \in T\}$.

Definition 4.3. A set $F \subseteq \mathcal{F}$ is called a $*$-closed set iff $F^* = F$.

Definition 4.4. A set $F \subseteq \mathcal{F}$ is called a computable set iff F is Z-injective and $*$-closed.

Definition 4.5. A function $f \in \mathcal{F}$ is called a computable function iff there exists a computable set F such that $f \in F$.

5. Properties of $*$-closure

Theorem 5.1. The operation $H \mapsto H^*$, where $H \subseteq \mathcal{F}$, is a topological closure

Proof. Definition 4.2 implies the following conditions

$$(5.1) \qquad \emptyset^* = \emptyset, \quad (\emptyset \text{ - the empty set}),$$

$$(5.2) \qquad \bigvee_{H \subseteq \mathcal{F}} H \subseteq H^*,$$

(5.3) $$\forall_{H_1,H_2 \subseteq \mathcal{F}} (H_1 \cup H_2)^* = H_1^* \cup H_2^*.$$

Since for any $a,b \in T$ we have $(f_a)_b = (f \circ \tilde{\alpha}) \circ \tilde{\beta} = f \circ (\tilde{\alpha} \circ \tilde{\beta})$, where $\alpha, \beta \in \Theta(T)$ and $\alpha(0) = a$, $\beta(0) = b$ and $\alpha \circ \beta = \gamma \in \Theta(T)$ we infer that $(f_a)_b = f_c$ for some $c \in T$. Thus we have

(5.4) $$\forall_{H \subseteq \mathcal{F}} (H^*)^* = H^*.$$

The conditions (5.1), (5.2), (5.3) and (5.4) imply Theorem 5.1.

From Theorem 5.1 we immediately obtain the following corollaries.

Corollary 5.1. If $G \subseteq H \subseteq \mathcal{F}$, then $G^* \subseteq H^*$.

Corollary 5.2. For any $H_1, H_2 \subseteq \mathcal{F}$, $(H_1 \cap H_2)^* \subseteq H_1^* \cap H_2^*$.

Corollary 5.3. Any finite union of $*$-closed sets is a $*$-closed set.

Corollary 5.4. Any intersection of $*$-closed sets is a $*$-closed set.

Theorem 5.2. If for every $s \in S$ the set F_s is computable, then the set $F = \bigcap_{s \in S} F_s$ is also computable.

Proof. If H is a Z-injective set and $G \subseteq H$, then G is a Z-injective set. Hence if the sets F_s are computable, then the set $F = \bigcap_{s \in S} F_s$ is Z-injective. By Corollary 5.4 the computability of the sets F_s implies $F^* = F$. Hence the set F is computable.

Theorem 5.3. For any $F \subseteq \mathcal{F}$ the following condition holds

(5.5) $$F^* = \bigcap_{f \in F} \{f\}^*.$$

Proof. Since for any $f \in F$ we have $\{f\} \subseteq F$, by Corollary 5.1 we infer that for any $f \in F$, $\{f\}^* \subseteq F^*$, hence

$\bigcup_{f\in F}\{f\}^*\subseteq F$. If $g\in F^*$, then there exist $a\in T$ and $f\in F$ such that $g=f_a$, thus $g\in\bigcup_{f\in F}\{f\}^*$. Hence $F^*\subseteq\bigcup_{f\in F}\{f\}^*$.

Theorem 5.3 implies the following corollary

C o r o l l a r y 5.5. Any union of $*$-closed sets is a $*$-closed set.

T h e o r e m 5.4. If the set $F=\bigcup_{s\in S}F_s$ is Z-injective and for every $s\in S$ the set F_s is $*$-closed, then the set F is computable.

P r o o f. The set F is $*$-closed by Corollary 5.5. Since the set F is Z-injective, it is computable.

6. Fundamental properties of computable functions

T h e o r e m 6.1. A function $f\in\mathcal{F}$ is computable iff

$$\forall_{a,b\in T}\ (f_a|Z=f_b|Z)\Longrightarrow(f_a=f_b). \tag{6.1}$$

P r o o f. If the function $f\in\mathcal{F}$ is computable, then there exists a computable set F such that $f\in F$. Any computable set is $*$-closed, hence for any $a\in T$, $f_a\in F$. Any computable set is Z-injective, thus if $\varphi,\psi\in F$, then $(\varphi|Z=\psi|Z)\Longrightarrow(\varphi=\psi)$. Let $\varphi=f_a$ and $\psi=f_b$. We see that condition (6.1) holds. If f satisfies condition (6.1), then the set $F=\{f\}^*$ is Z-injective and $*$-closed. Thus the set F is computable. Since $f\in F$, f is a computable function.

D e f i n i t i o n 6.1. A function $f\in\mathcal{F}$ is called Z-injective iff

$$\forall_{a,b\in T}\ \left[(a\neq b)\Longrightarrow(f_a|Z\neq f_b|Z)\right]. \tag{6.2}$$

C o r o l l a r y 6.1. If a function $f\in\mathcal{F}$ is Z-injective, then for any $a\in T$ the function f_a is also Z-injective.

C o r o l l a r y 6.2. Any Z-injective function is computable.

Let $c \in T$, $\gamma \in \Theta(T)$ and $\gamma(0) = c$. We denote: $\gamma^1 = \gamma$, $\gamma^n = \gamma \circ \gamma^{n-1}$ for any natural number $n > 1$. If $d \in T$ and $\gamma^n(0) = d$, then we write $d = n \cdot c$, and if $e \in T$, $\alpha[\gamma^n(0)] = e$ and $\alpha(0) = e$, then we write $e = a \oplus n \cdot c$.

D e f i n i t i o n 6.2. A function $f \in \mathcal{F}$ is called Z-periodic iff

$$\exists_{a \in T} \exists_{0 \neq c \in T} \forall_{n \in N} (f_a|Z = f_{a \oplus n \cdot c}|Z). \tag{6.3}$$

C o r o l l a r y 6.3. If there exists $a \in T$ such that the function f_a is Z-periodic, then the function f is also Z-periodic.

T h e o r e m 6.2. If a function $f \in \mathcal{F}$ is computable and not Z-injective, then it is Z-periodic.

P r o o f. If a function $f \in \mathcal{F}$ is not Z-injective, then

$$\exists_{a,b \in T} \left[(a \neq b) \wedge (f_a|Z = f_b|Z)\right]. \tag{6.4}$$

Let $a \leqslant b$ and $\alpha, \beta \in \Theta(T)$, $\alpha(0) = a$, $\beta(0) = b$. Let $c \in T$ be such that $\alpha(c) = b$ and $\gamma \in \Theta(T)$, $\gamma(0) = c$. We have $\alpha[\gamma(0)] = \alpha(c) = b$. Since $a \neq b$, we have $\gamma \neq \iota$. Hence $b = a \oplus c$ and from (6.4) it follows that $f_a|Z = f_{a \oplus c}|Z$. The function $f \in \mathcal{F}$ is computable, thus from (6.1) if $f_a|Z = f_{a \oplus c}|Z$ then $f_a = f_{a \oplus c}$ and $f_a \circ \tilde{\gamma} = f_{a \oplus c}$. Consequently $f_{a \oplus c} = f_{a \oplus 2c}$, hence $f_{a \oplus c}|Z = f_{a \oplus 2c}|Z$. Analogously, if $f_a|Z = f_{a \oplus c}|Z = \ldots = f_{a \oplus k \cdot c}|Z$, then $f_{a \oplus c}|Z = f_{a \oplus 2c}|Z = \ldots f_{a \oplus (k+1) \cdot c}|Z$. Thus if $f_a|Z = f_{a \oplus k \cdot c}|Z$, then $f_a\ Z = f_{a \oplus (k+1) \cdot c}|Z$. Hence condition (6.3) holds.

T h e o r e m 6.3. For any function $f \in \mathcal{F}$ the set $\{f\}^*$ is computable iff the function f is computable.

P r o o f. Since $f \in \{f\}^*$, we infer that if the set $\{f\}^*$ is computable, then the function f is computable. If the function f is computable, then from (6.1) it follows that the set $\{f\}^*$ is Z-injective and the set $\{f\}^*$ is $*$-closed, thus it is computable.

T h e o r e m 6.4. If $g,h \in \{f\}^*$, then $g \in \{h\}^*$ or $h \in \{g\}^*$.

P r o o f. If $g,h \in \{f\}^*$, then there exist $a,b \in T$ such that $g = f_a$ and $h = f_b$. Let $a \leqslant b$ and $\alpha,\beta \in \Theta(T)$, $\alpha(0) = a$, $\beta(0) = b$. Let $c \in T$ be such that $\alpha(c) = b$ and $\gamma \in \Theta(T)$, $\gamma(0) = c$. We have $(f_a)_c = (f \circ \tilde{\alpha}) \circ \tilde{\gamma} = f \circ (\tilde{\alpha} \circ \tilde{\gamma}) = f \circ \tilde{\beta}$, because $\alpha[\gamma(0)] = b = \beta(0)$. Thus $(f_a)_c = f_b$, hence $h = g_c$ and $h \in \{g\}^*$. Analogously, if $b \leqslant a$, then $g \in \{h\}^*$.

7. Conclusions

The concept of deterministic computability seems to provide a base for uniform approach both to discrete and continuous computability. To get the notions of discrete computability as those in [1] and [3] it suffices to consider the system $(T, \leqslant, \Theta, X, Y, Z)$ with: T - the set of all non-negative integers, $\leqslant$ - the natural ordering of T, $\Theta(T)$ - the set of all isometrical mappings from T to T, X - an arbitrary one-element set, Y - an arbitrary non-empty set, $Z = \{0\}$ (Pawlak's computability in [1]) or $Z = \{0,1,\dots,j-1\}$ (Grodzki's computability in [3]). To get the notions of continuous computability like those in [4] and [5] it suffices to consider the system $(T, \leqslant, \Theta, X, Y, Z)$ with: T - the set of all non-negative real numbers, $\leqslant$ - the natural ordering of T, $\Theta(T)$ - the set of all isometrical mappings from T to T, X - an arbitrary one-element set and $Z = \langle 0;\tau)$ (the τ-computability of [4]) or X - $(n-1)$-dimensional Euclidean space and Z - an arbitrary non-empty set such that $Z \subseteq T \times X$ (the Z-computability of [5]), Y - an arbitrary non-empty set. Moreover, to get the notions of computability of [1], [3], [4] and [5] it suffices to assume that the computation is a function $f : T \times X \to Y$.

Having introduced partial functions we may as well examine hybrid computations. A hybrid computation can be defined as follows. Let, for any $\varphi \in \mathcal{T}$, $Z_\varphi = \{(t,x) \in Z : \varphi(t,x) \in Y\}$. A computable function f is called a hybrid computation iff there exists $a \in T$ such that $Z_{f_a} \neq Z_f$. Because the empty function σ is computable, it is possible to examine also computations which are finite in time (a computation f is said to be finite in time iff there exists $a \in T$ such that $f_a = \sigma$).

BIBLIOGRAPHY

[1] Z. Pawlak: Stored program computers, Algorytmy 10 (1969) 5-19.

[2] W. Kwasowiec: Generable sets, Information and Control, 17 (1970) 220-226.

[3] Z. Grodzki: The k-machines, Bull. Acad. Polon. Sci., Ser. Sci. Math. Astronom. Phys., 18 (1970) 399-402.

[4] B. Konikowska: Continuous machines, τ-computations and τ-computable sets, Bull. Acad. Polon. Sci., Ser. Sci. Math. Astronom. Phys., 19 (1971) 525-529.

[5] W. Żakowski: Continuous simple Z-machines, Z-computations and Z-computable sets, Bull. Acad. Polon. Sci., Ser. Sci. Math. Astronom. Phys., 22 (1974) 1063--1067.

[6] I. Nabiałek, W. Żakowski: On $*$-closure of sets of functions, Demonstratio Math. 8 (1975) 249-252.

INSTITUTE OF MATHEMATICS, TECHNICAL UNIVERSITY OF WARSAW

Received February 14, 1976.