TURING TRANSFORMATION AND STRONG COMPUTABILITY
OF TURING COMPUTERS

By

Seiiti HUZINO
(Received July 1, 1959)

§0. Introduction

A. Nerode found a necessary and sufficient condition for an automaton transformation to be linear when its domain of definition consists of all applications from a set of natural numbers to a finite commutative ring with unit ([10]). To study finite automata, it is sufficient to investigate the properties of the above transformation thoroughly, since the concept of automaton transformation is a generalized variant of the concept of representable event of S.C. Kleene ([9] and [12]). And, in fact, we shall have some interesting problems if we consider finite automata from the point of view of automaton transformation. In this paper, however, we shall further extend the idea of this transformation, define a Turing transformation and consider concretely fundamental operations about Turing computers, based on the theory of sequential automata. And, further, we shall obtain some results relating to strong computability of an integral-valued function.

§1. Definition of Turing transformation

Firstly, we shall define a simple Turing transformation \( T \) and a (general) Turing transformation \( T_s \). Let \( A \) be a given (non empty) finite set, the elements of which are \( a_0, a_1, \ldots, a_n \). Let \( Z \) consist of all rational integers, and denote by \( A^z \) a set of all functions on \( Z \) to \( A \). Let \( \tilde{A}^z \) consist of all elements \( E \) of \( A^z \) such that \( E(i) = a_0 \) for all but a finite number of \( i \in Z \). A map \( T: \tilde{A}^z \to \tilde{A}^z \) is called to be a simple Turing transformation if there exist a finite set \( Q = \{ q_i \}_{i=1,2,\ldots} \), a finite subset \( D = \{ d_i \}_{i=0,1,2,\ldots} \) of \( Z \), a fixed subset \( B \) of \( A \times Q \), a map \( T_q: B \to Q \), a map

\[
T_{AUD}: B \to A \cup D
\]

and an element \( \bar{q} \in Q \) such that corresponding to each \( E^0 \in \tilde{A}^z \) there exists an \( h \in Q^\{d\} \) satisfying

1) A symbol \( N \) means a set of all non-negative rational integers.
(1.1) \( h(0) = q, \ h(n+1) = T_q(E^n(\nu(n)), h(n)) \),

(1.2) \( E^{n+1}(\nu(n+1)) = \begin{cases} E^n(\nu(n)) + T_{A^D}(E^n(\nu(n)), h(n)) & \text{for } T_{A^D}(E^n(\nu(n)), h(n)) \in D, \\ T_{A^D}(E^n(\nu(n)), h(n)) & \text{for } T_{A^D}(E^n(\nu(n)), h(n)) \in D, \end{cases} \)

\( \nu(n) = \begin{cases} \nu(n+1) + T_{A^D}(E^{n-1}(\nu(n-1)), h(n-1)) & \text{for } T_{A^D}(E^{n-1}(\nu(n-1)), h(n-1)) \in D, \\ \nu(n-1) & \text{for } T_{A^D}(E^{n-1}(\nu(n-1)), h(n-1)) \in D, \end{cases} \)

and if

(i) \( T(E^0) = E^r \) when \( T_q(E^r(\nu(r)), h(r)) \) is not defined for some \( r \in N \)

and

(ii) \( T(E^0) = \phi \) when \( (E^r(\nu(r)), h(r)) \in B \) for any \( r \),

where \( \phi \) is an element of \( A^t \) such that \( \phi(i) = a_0 \) for all \( i \in Z \) and

(1.4) \( E^{n+1}(j) = \begin{cases} E^n(j) & \text{for } j \equiv \nu(n+1), \\ E^{n+1}(\nu(n+1)) & \text{for } j \equiv \nu(n+1). \end{cases} \)

We call \( E^0 \) an initial expression or tape expression. It is convenient to take \( E^0 \in \tilde{A}^z \) as an element such that \( E^0(i) = a_0 \) for \( i < 0 \) and \( E^0(0) = a_j \) (\( j \neq 0 \)). Hereafter we shall consider only such an expression as above and let \( q = q_1 \). A subset of \( \tilde{A}^z \) consisting of the initial expressions in the case (i) in the definition of a given simple Turing transformation \( T \) is called a computable domain of \( T \), which we denote by \( \mathcal{C}(T) \). By a resultant of \( E^0 \) with respect to \( T \) we mean the elements \( E^r \) such that \( E^0 \in \mathcal{C}(T) \) and \( T(E^0) = E^r \).

We shall introduce now the following operations among expressions.

Let \( E_i \) and \( E_j \) be two expressions in \( \tilde{A}^z \otimes \{ \phi \} \), and we define the product \( E_i \circ E_j \in \tilde{A}^z \) as

(1.5) \( E_i \circ E_j(k) = \begin{cases} E_i(k) & \text{for } k \leq \Delta(E_i), \\ E_j(k - \Delta(E_i) - 1) & \text{for } k > \Delta(E_i), \end{cases} \)

where the number \( \Delta(E_i) \) is the greatest value \( k \in Z \) such that \( E_i(k) = a_0 \) and \( E_i(k+1) = a_0 \). The operation \( \circ \) is associative, but is not commutative.

We define \( E^{(r)} \) for any \( E \in \tilde{A}^z \) as \( E^{(r)}(k) = E(k-r) \) (\( k \in Z \)). Let \( E_i \circ E_2 \equiv E_i \circ E_2^{(r)} \) for \( E_i, E_2 \in \tilde{A}^z \), and we write simply \( E_i \ast E_2 \) for \( E_i \circ E_2 \). The operation \( \ast \) is also associative, but is not commutative. By \( \bar{k} \in \tilde{A}^z (k \in N) \) we mean an element \( E \) of \( \tilde{A}^z \) such that

(1.6) \( E(i) = \begin{cases} a_0 & (i < 0) \\ a_i & (0 \leq i \leq k) \\ a_0 & (i \geq k + 1), \end{cases} \)
and by \((k_1, k_2, \ldots, k_i) \in \bar{A}^i\) \((k = (k_1, \ldots, k_i) \in N^i)\) an element \(E\) of \(\bar{A}^i\) such that \(E = \bar{k}_1 * \bar{k}_2 * \cdots * \bar{k}_i\), where we write simply \(\bar{k}\) for \((k_1, k_2, \ldots, k_i)\) when confusion does not occur.

For a given element \(E \in \bar{A}^i\), we denote by \(<E>_{j}\) \((j \neq 0)\) the number of the elements \(\{a_i\}\) contained in the image \(\{E(i)\}_{i \in \bar{Z}}\). We write simply \(<E>\) for \(<E>_{1}\).

Given a simple Turing transformation \(T\), the structure \(\{Q, A, A \cup D; T_0, T_{A \cup D}\}\) defines an incomplete sequential machine (or an incomplete sequential automaton) in the sense of [4]~[7]. Here the word "incomplete" means that the state-table and the output-table of the machine (or automaton) are not completely defined in the sense of [4]~[7].

Next we shall define a (general) Turing transformation \(T_\epsilon\) as follows:

A map \(T_\epsilon: \bar{A}^i \rightarrow \bar{A}^i\) is a (general) Turing transformation if there exist a finite set \(Q = \{q_i\}_{i \geq 1}\), a finite subset \(D = \{d_i\}_{i \geq 1}\) of \(\bar{Z}\), a fixed subset \(B\) of \(A \times Q\), a map \(T_0: B \rightarrow Q\), a map \(T_{A \cup Q \cup D}: B \rightarrow A \cup Q \cup D\) and an element \(\bar{q} \in Q\) such that corresponding to each \(E^\epsilon \in \bar{A}^i\) there exists an \(h \in Q^\epsilon\) satisfying

(1.7) \(h(0) = \bar{q}\),

(1.8) \(h(n + 1) = \begin{cases} 
T_0(E^n(\nu(n)), h(n)) & \text{for } T_{A \cup Q \cup D}(E^n(\nu(n)), h(n)) \in Q, \\
T_0(E^n(\nu(n)), h(n)) & \text{for } T_{A \cup Q \cup D}(E^n(\nu(n)), h(n)) \in Q \text{ and } <E^n> \in C, \\
T_{A \cup Q \cup D}(E^n(\nu(n)), h(n)) & \text{for } T_{A \cup Q \cup D}(E^n(\nu(n)), h(n)) \in Q \\
& \text{and } <E^n> \in C,
\end{cases}
\)

(1.9) \(\nu(n) = \begin{cases} 
\nu(n - 1) + T_{A \cup Q \cup D}(E^{n-1}(\nu(n - 1)), h(n)) & \text{for } T_{A \cup Q \cup D}(E^{n-1}(\nu(n - 1)), h(n - 1)) \in D, \\
\nu(n - 1) & \text{for } T_{A \cup Q \cup D}(E^{n-1}(\nu(n - 1)), h(n - 1)) \in D,
\end{cases}
\)

(1.10) \(E^{n+1}(\nu(n + 1)) = \begin{cases} 
E^{n}(\nu(n)) & \text{for } T_{A \cup Q \cup D}(E^{n}(\nu(n)), h(n)) \in D, \\
T_{A \cup Q \cup D}(E^{n}(\nu(n)), h(n)) & \text{for } T_{A \cup Q \cup D}(E^{n}(\nu(n)), h(n)) \in A, \\
E^{n}(\nu(n)) & \text{for } T_{A \cup Q \cup D}(E^{n}(\nu(n)), h(n)) \in Q,
\end{cases}
\)

where \(C\) is a fixed subset of \(N\), and if

(i) \(T_\epsilon(E^0) = E^r\) when \(T_0(E^r(\nu(r)), h(r))\) is not defined for some \(r\), and

(ii) \(T_\epsilon(E^0) = \phi\) when \((E^r(\nu(r)), h(r)) \in B\) for any \(r\),

where
(11) \[ E_{n+1}(j) = \begin{cases} E_{n+1}(\nu(n+1)) & \text{for } j = \nu(n+1), \\ E_n(j) & \text{for } j \neq \nu(n+1). \end{cases} \]

When \( D = D_1 = \{-1, 0, 1\} \), the above transformation induces a Turing machine in the sense of Turing-Davis ([3]). The domain of computability is defined analogously as in the case of simple Turing transformation, and the resultant of an element of the domain is called a resultant of \( C \)-computation (or a \( C \)-resultant).

Given a (general) Turing transformation \( T_e \), we have a chain of tape expressions:

(12) \[ E^0 \rightarrow \delta(E^0; T_e) \rightarrow \delta(E^0; T_e) \rightarrow \cdots \rightarrow \delta^r(E^0; T_e) \rightarrow \delta^r(E^0; T_e) = E^r, \]

if \( E^0 \) belongs to \( \mathcal{E}(T_e) \) and \( T_e(E^0) = E^r \), where the maps \( \delta^k : \tilde{A}^k \times \tilde{A}^k \rightarrow \tilde{A}^k \) (\( k = 1, 2, \cdots, \)) are such that \( \delta^k(E; T_e) \) is defined as an instantaneous \( k \)th tape expression obtained when we use a Turing transformation \( T_e \in \mathcal{E} \) under an initial tape expression \( E^0 \in \mathcal{E}(T_e) \), and consequently is not defined when \( E^0 \notin \mathcal{E}(T_e) \) for a given \( T_e \in \mathcal{E} \). Here \( \mathcal{E} \) means a set of all Turing transformations: \( \tilde{A}^k \rightarrow \tilde{A}^k \). In the case of a simple Turing transformation we get an analogous chain of tape expressions.

Let \( T_{e_1}^{(3)} \) and \( T_{e_2}^{(3)} \) be two Turing transformations, and let \( Q_i = \{q_i^{(3)}\}_j \) be a state-set associated with \( T_{e_i}^{(3)} \) (\( i = 1, 2 \)). A composition \( T_{e_1}^{(3)} \wedge T_{e_2}^{(3)} \) is defined as \( T_{e_1}^{(3)} \wedge T_{e_2}^{(3)}(E) = T_{e_1}^{(3)}(T_{e_2}^{(3)}(E)) \) for \( E \in \mathcal{E}(T_{e_1}^{(3)} \wedge T_{e_2}^{(3)}) \), where \( \mathcal{E} \) (\( T_{e_1}^{(3)} \wedge T_{e_2}^{(3)}) \) is a set of all \( E \) such that \( E \in \mathcal{E}(T_{e_1}^{(3)}) \) and \( T_{e_1}^{(3)}(E) \in \mathcal{E}(T_{e_1}^{(3)}) \). Here the transformation \( T_{e_1}^{(3)} \) is a transformation such that the transition diagram of the sequential automaton induced from it is obtained by adding transitions \( q_i^{(3)} \xrightarrow{a_k} q_j^{(3)} \) (\( j = 1, 2, \cdots, \#(Q_2) \)) to the transition diagram of the sequential automaton induced from \( T_{e_1}^{(3)} \), where \( a_k \) represents any element which is not contained in the input-set of \( q_i^{(3)} \). And \( T_{e_1}^{(3)} \) is called to be a transformation obtained from \( T_{e_1}^{(3)} \) by operating a canonical transition to the state \( q_i^{(3)} \) as \( q_i^{(3)} \xrightarrow{\#(Q_2)} q_i^{(3)} \). And \( T_{e_1}^{(3)} \) is the transformation such that the state-set of the sequential automaton induced from it is obtained by renumbering the elements \( q_i^{(3)} \) as \( q_i^{(3)} (i = 1, 2, \cdots, \#(Q_1)) \), therefore, two transformations \( T_{e_1}^{(3)} \) and \( T_{e_1}^{(3)} \) are essentially the same. Obviously the product \( \wedge \) is not commutative.

By Turing computer \([T_e]\) we mean any one of incomplete sequential
automata which are constructed from the structure \(\{Q, A, A \cup Q \cup D; T, T_A \cup Q \cup D\}\) associated with a given Turing transformation \(T\). When \(D = D = \{-1, 0, +1\}\), the concept of the computer defined above coincides with that of ordinary Turing machine ([2], [3]). Here we assume the existence of infinite tape.

\section*{2. Computability}

An integral-valued function \(f : D(\subset N^i) \to N\) is partially computable if there exists a simple Turing transformation \(T\) such that \(\overline{x} \in A^{\theta}(x \in D = D(f))\) belongs to \(C(T)\) and \(f(x) = \langle T(\overline{x})\rangle\). When \(D(f) = N^i\), \(f\) is said to be computable and \(T\) is said to compute \(f\). Similarly, an integral-valued function \(f : D(\subset N^i) \to N\) is partially \(C\)-computable if there exists a Turing transformation \(T_c\) such that \(\overline{x} \in A^{\theta}(x \in D)\) belongs to \(C(T_c)\) and \(f(x) = \langle T_c(\overline{x})\rangle\). A function \(f : D(\subset N^i) \to N\) is strongly partially \(C\)-computable if there exists a Turing transformation \(T_e\) such that \(\overline{x} \in A^{\theta}(x \in D)\) belongs to \(C(T_e)\) and \(T_e(\overline{x}) = f(x) - 1^{(r)}\) for some \(r \geq 0\). If \(f\) is strongly partially \(C\)-computable, \(f\) is also partially \(C\)-computable, since \(\langle T_e(\overline{x})\rangle = f(x)\). Next we shall define another computability. Define a map \(v : A \to N\) by \(v(a_i) = i\) \((i = 0, 1, \ldots, a)\) and a map \(\rho_\alpha : A^{\theta} \to R\) \((\theta \leq \#(A))\) by

\[\rho_\alpha(E) = \sum_{i \in \mathbb{Z}} v(E(i))\theta^i,\]

where \(R\) consists of all \(\theta\)-adic numbers. \(\rho_\alpha\) is defined if and only if \(v(E(i)) < \theta^i\) for all \(i\), hence \(\rho_{\#(A)}\) is always defined. A function \(f : D(\subset N^i) \to N\) is indirectly partially \((\theta, C)\)-computable if there exists a Turing transformation \(T_c\) such that \(\overline{x} \in A^{\theta}(x \in D)\) belongs to \(C(T_c)\) and

\[\langle f(x) \rangle_{\theta} = \langle \rho_\alpha(T_c(\overline{x})) \rangle \theta^r\] for some \(r \in \mathbb{Z}\),

where \(\langle f(x) \rangle_{\theta}\) is a number \(f(x) \in N\) represented as \(\theta\)-adic number. When \(r = 0\) for all \(\overline{x} \in C(T_c)\) in the above definition, \(f\) is said to be directly partially \((\theta, C)\)-computable.

\textbf{Lemma 2.1. If there exists a Turing transformation \(T_c\) such that \(T_c(\overline{x}) = \overline{y}^{(r)}(x \in N^i, y \in N \ominus \{0\}, r \in \mathbb{Z})\), then there exists a Turing trans-}

2) \(D(f)\) means a domain of definition of a function \(f\).

3) Let \(-1 \phi = \phi\) for any \(r\). See [1].
formation \( \bar{T}_e \) such that \( \bar{T}_e(\bar{x}) = \bar{y} \) with respect to relative coordinates.\(^4\)

**Proof.** We consider a computer \([T^{(1)}] \) shown in Figure 1, where \( a_i \) is an element different from \( a_0 \) and \( a_1 \). Let \( d \equiv \max \{|d_i|\} \), where \( d_i \) belongs to \( D_{x_e} \), which is the set \( D \) in the definition of \( T_e \).

If \( r = 0 \), then we have

\[ S(T_e(\bar{x}); T^{(1)}) = \bar{y}^{(r+1)}, \quad S(T_e(\bar{x}); T^{(1)}) = \bar{y}. \]

Set \( S^{(1)} \equiv S(T_e(\bar{x}); T^{(1)}) \), then if \( pd \geq r > (p-1)d \) \((p = 1, 2, \ldots)\), we have

\[ S^{(1)} = \left\{ \tilde{a}_i \quad \text{for} \quad (r-1) \quad \text{y} \right\}_0. \]

Here the lower index outside parentheses shows the position of the computer and \( \tilde{a}_i \in \tilde{A}e \) has the property:

\[ \tilde{a}_i(k) = \begin{cases} a_0 & (k \neq 0) \\ a_i & (k = 0). \end{cases} \]

\(^4\) The relative coordinates mean the one obtained when we take the position which the computer catches at this time as origin of coordinates.
Proceeding in succession, the following expressions are obtained:

\[ S^{(d)} = \left\{ \bar{a}_1 \ast \bar{a}_1 \ast \bar{y} \right\}^{(d)} \]

\[ S^{(d+1)} = \left\{ \bar{a}_1 \ast \bar{y} \right\}^{(d+1)} \]

\[ S^{(d+2)} = \left\{ \bar{a}_2 \ast \bar{a}_1 \ast \bar{y} \right\}^{(d+2)} \]

\[ S^{(d+3)} = \left\{ \bar{a}_1 \ast \bar{y} \right\}^{(d+3)} \]

\[ S^{(d+4)} = \left\{ \bar{a}_2 \ast \bar{a}_1 \ast \bar{y} \right\}^{(d+4)} \]

\[ S^{(d+5)} = \left\{ \bar{a}_1 \ast \bar{y} \right\}^{(d+5)} \]

\[ S^{(d+6)} = \left\{ \bar{a}_2 \ast \bar{a}_1 \ast \bar{y} \right\}^{(d+6)} \]

\[ S^{(d+7)} = \left\{ \bar{a}_1 \ast \bar{y} \right\}^{(d+7)} \]

\[ S^{(d+8)} = \left\{ \bar{a}_2 \ast \bar{a}_1 \ast \bar{y} \right\}^{(d+8)} \]

\[ S^{(d+9)} = \left\{ \bar{a}_1 \ast \bar{y} \right\}^{(d+9)} \]

\[ S^{(d+10)} = \left\{ \bar{a}_2 \ast \bar{a}_1 \ast \bar{y} \right\}^{(d+10)} \]

\[ S^{(d+11)} = \left\{ \bar{a}_1 \ast \bar{y} \right\}^{(d+11)} \]

\[ S^{(d+12)} = \left\{ \bar{a}_2 \ast \bar{a}_1 \ast \bar{y} \right\}^{(d+12)} \]

\[ S^{(d+13)} = \left\{ \bar{a}_1 \ast \bar{y} \right\}^{(d+13)} \]

\[ S^{(d+14)} = \left\{ \bar{a}_2 \ast \bar{a}_1 \ast \bar{y} \right\}^{(d+14)} \]

\[ S^{(d+15)} = \left\{ \bar{a}_1 \ast \bar{y} \right\}^{(d+15)} \]

\[ \ldots \]

\[ S^{(2p-1)d + 6(p-1) + 3} = \left\{ \bar{a}_1 \ast \bar{a}_1 \ast \bar{y} \right\}^{(2p-1)d + 6(p-1) + 3} \]

\[ S^{(2p-1)d + 6(p-1) + 3} = \left\{ \bar{a}_1 \ast \bar{y} \right\}^{(2p-1)d + 6(p-1) + 3} \]

\[ S^{(2p+1)d + 6(p-1) + 6} = \left\{ \bar{y} \right\}^{(2p+1)d + 6(p-1) + 6} \]

\[ S^{(2p+1)d + 6(p-1) + 6+r} = \left\{ \bar{y} \right\}^{(2p+1)d + 6(p-1) + 6+r} \]

Hence in this case we obtain the required relation if we take the product \( T^{(1)} \wedge T_c \) as \( \overline{T_c} \). In all other cases we can prove, analogously, that it is sufficient to take the same product \( T^{(1)} \wedge T_c \) as \( \overline{T_c} \).

**Remark.** In general, \( \mathcal{E}(T_c) \subset \mathcal{E}(\overline{T_c}) \), hence the above lemma holds only for \( \bar{x} \in \mathcal{E}(T_c) \). If \( T \) is 'regular' in Davis' sense, then \( \mathcal{E}(T_c) = \mathcal{E}(\overline{T_c}) \), and no questions occur. Otherwise we must consider \( \mathcal{E}(T_c) \) only by observing two computers \( [T_c] \) and \( [\overline{T_c}] \).

**Proposition 2.1.** If a function \( f: \mathcal{D}(\subset \mathcal{N}) \to \mathcal{N} \) satisfying \( f(x) = 0 \) for all \( x \in \mathcal{D} \) is strongly (partially) \( C \)-computable, then it is indirectly
(partially) \((\#(A) - 1, C)\)-computable.

**Proof.** From assumption there exists a Turing transformation \(T_r\) such that \(T_r(\tilde{x}) = f(x) - 1^{(r)}\) for some \(r \geq 0\) and \(\tilde{x} \in \mathcal{C}(T_r)\). We consider a Turing computer \([T^{(1)}]\), the transition diagram of which is shown in Figure 2. Let \(S^{(1)} \equiv S^{(1)}(\tilde{x}^{(r)}; T^{(1)})\). Then we get the relations

\[
S^{(r)} = \left[ \frac{x}{\tilde{x}^{(r)}} \right]
\]

\[
S^{(r+1)} = \left[ \tilde{a}_{\#(A) - 1} \circ \frac{x}{\tilde{x}^{(r)}} \right]^{(r)}
\]

\[
S^{(r + x + 3)} = \left[ \tilde{a}_{\#(A) - 1} \circ \frac{x}{\tilde{x}^{(r)}} \circ \tilde{a}_{\#(A) - 1} \right]^{(r)}
\]

\[
S^{(r + 2x + 6)} = \left[ \tilde{a}_{\#(A) - 1} \circ \frac{x}{\tilde{x}^{(r)}} \circ \tilde{a}_{\#(A) - 1} \right]^{(r)}
\]

\[
S^{(r + 2x + 8)} = \left[ \tilde{a}_{1} \circ \frac{x}{\tilde{x}^{(r)}} \circ \tilde{a}_{\#(A) - 1} \right]^{(r)}
\]

\[
S^{(r + 2x + 11)} = \left[ \tilde{a}_{1} \circ \frac{x}{\tilde{x}^{(r)}} \circ \tilde{a}_{\#(A) - 1} \right]^{(r)}
\]

\[
S^{(r + 2x + 14)} = \left[ \tilde{a}_{2} \circ \frac{x}{\tilde{x}^{(r)}} \circ \tilde{a}_{\#(A) - 1} \right]^{(r)}
\]

\[
S^{(r + 2x + 18)} = \left[ \tilde{a}_{3} \circ \frac{x}{\tilde{x}^{(r)}} \circ \tilde{a}_{\#(A) - 1} \right]^{(r)}
\]

If \(x < \#(A) - 2\), we have

\[
S^{(r + x + 4)} = \left[ \tilde{a}_{x} \circ \tilde{a}_{\#(A) - 1} \right]^{(r)}
\]

\[
S^{(r + x + 6)} = \left[ \tilde{a}_{x} \right]^{(r)}
\]

If \(x = \#(A) - 2\), then we have

\[
S^{(r + x + 3 + x)} = \left[ \tilde{a}_{x + 1} \circ \tilde{a}_{\#(A) - 1} \right]^{(r)}
\]

\[
S^{(r + x + 4 + x + 1)} = \left[ \tilde{a}_{x + 1} \circ \tilde{a}_{\#(A) - 1} \right]^{(r)}
\]

\[
S^{(r + x + 5 + x + 2)} = \left[ \tilde{a}_{\#(A) - 2} \circ \tilde{a}_{\#(A) - 1} \right]^{(r)}
\]

\[
S^{(r + x + 6 + x + 4)} = \left[ \tilde{a}_{\#(A) - 3} \right]^{(r)}
\]

\[
S^{(r + x + 7 + x + 5)} = \left[ \tilde{a}_{\#(A) - 4} \right]^{(r)}
\]

\[
S^{(r + x + 8 + x + 6)} = \left[ \tilde{a}_{\#(A) - 5} \right]^{(r)}
\]

\[
S^{(r + x + 9 + x + 7)} = \left[ \tilde{a}_{\#(A) - 6} \right]^{(r)}
\]

\[
S^{(r + x + 10 + x + 8)} = \left[ \tilde{a}_{\#(A) - 7} \right]^{(r)}
\]

\[
S^{(r + x + 11 + x + 9)} = \left[ \tilde{a}_{\#(A) - 8} \right]^{(r)}
\]

\[
S^{(r + x + 12 + x + 10)} = \left[ \tilde{a}_{\#(A) - 9} \right]^{(r)}
\]

\[
S^{(r + x + 13 + x + 11)} = \left[ \tilde{a}_{\#(A) - 10} \right]^{(r)}
\]

\[
S^{(r + x + 14 + x + 12)} = \left[ \tilde{a}_{\#(A) - 11} \right]^{(r)}
\]

\[
S^{(r + x + 15 + x + 13)} = \left[ \tilde{a}_{\#(A) - 12} \right]^{(r)}
\]

\[
S^{(r + x + 16 + x + 14)} = \left[ \tilde{a}_{\#(A) - 13} \right]^{(r)}
\]

\[
S^{(r + x + 17 + x + 15)} = \left[ \tilde{a}_{\#(A) - 14} \right]^{(r)}
\]

\[
S^{(r + x + 18 + x + 16)} = \left[ \tilde{a}_{\#(A) - 15} \right]^{(r)}
\]

\[
S^{(r + x + 19 + x + 17)} = \left[ \tilde{a}_{\#(A) - 16} \right]^{(r)}
\]

\[
S^{(r + x + 20 + x + 18)} = \left[ \tilde{a}_{\#(A) - 17} \right]^{(r)}
\]

\[
S^{(r + x + 21 + x + 19)} = \left[ \tilde{a}_{\#(A) - 18} \right]^{(r)}
\]

\[
S^{(r + x + 22 + x + 20)} = \left[ \tilde{a}_{\#(A) - 19} \right]^{(r)}
\]
Turing transformation and strong computability of Turing computers
Next let \((p-1)(\#(A)-1) \leq x \leq p(\#(A)-1)\) \((p=1,2,\cdots)\).

We shall now study the case when \(p=2\). Other cases can be treated analogously.

\[\begin{align*}
\mathcal{S}^{(r+2x+8+\{\#(A)\}-4+\{\#(A)\}+1)} &= \left\{ \bar{a}_{\#(A)-1} \circ \bar{a}_{\#(A)-1} \circ \bar{a}_{\#(A)-1} \right\}^{(r)} \\
\mathcal{S}^{(r+2x+8+\{\#(A)\}-3+\{\#(A)\}+2)} &= \left\{ \bar{a}_{\#(A)-2} \circ \bar{a}_{\#(A)-2} \circ \bar{a}_{\#(A)-2} \right\}^{(r)} \\
\mathcal{S}^{(r+2x+8+\{\#(A)\}-3+\{\#(A)\}+2+2\#(A))} &= \left\{ \bar{a}_{\#(A)-3} \circ \bar{a}_{\#(A)-3} \circ \bar{a}_{\#(A)-3} \right\}^{(r)} \\
\mathcal{S}^{(r+2x+8+\{\#(A)\}-3+\{\#(A)\}+2+2\#(A)+1)} &= \left\{ \bar{a}_{\#(A)-4} \circ \bar{a}_{\#(A)-4} \circ \bar{a}_{\#(A)-4} \right\}^{(r-1)} \\
\mathcal{S}^{(r+2x+8+\{\#(A)\}-1+8+\{\#(A)\}-3+\{\#(A)\}+2+2\#(A)+1+2\#(A)+3)} &= \left\{ \bar{a}_{1} \right\}^{(r)} \\
\end{align*}\]

If \(x=\#(A)-1\), then we have

\[\mathcal{S}^{(r+2x+8+\{\#(A)\}-1+8+\{\#(A)\}-3+\{\#(A)\}+2+2\#(A)+1+2\#(A)+3)} = \left\{ \bar{a}_{1} \right\}^{(r-1)}.
\]

If \(\#(A)-1 < x < 2(\#(A)-1)\), then we have the following chain of expressions.

\[\begin{align*}
\cdots & \rightarrow \bar{a}_{1} \circ \bar{a}_{\#(A)-1} \circ \left( \bar{a}_{\#(A)-1} \right) \circ \bar{a}_{\#(A)-1}^{(5)} \\
\cdots & \rightarrow \bar{a}_{1} \circ \bar{a}_{\#(A)-1} \circ \left( \bar{a}_{\#(A)-1} \right) \\
\cdots & \rightarrow \bar{a}_{1} \circ \bar{a}_{\#(A)-1} \circ \left( \bar{a}_{\#(A)-1} \right) \\
\cdots & \rightarrow \bar{a}_{1} \circ \bar{a}_{\#(A)-1} \circ \bar{a}_{\#(A)+1} \circ \bar{a}_{\#(A)-1} \\
\cdots & \rightarrow \bar{a}_{1} \circ \bar{a}_{\#(A)-1} \circ \bar{a}_{\#(A)+1} \\
\end{align*}\]

The same procedures hold for \(p(\#(A)-1) \leq x \leq (p+1)(\#(A)-1)\). Taking the product \(T^{(1)} \wedge T_{c}\), we get for any \(\bar{x} \in C(T_{c})\)

\[
\begin{align*}
\rho_{\#(A)-1} \left[ T^{(1)} \wedge T_{c}(\bar{x}) \right] &= \rho_{\#(A)-1} \left[ T^{(1)} \left[ f(\bar{x})-1 \right] \right] \\
&= \rho_{\#(A)-1} \left[ f(\bar{x})-1 \right] \cdot (\#(A)-1)^{t} \text{ for some } t \in Z \\
&= [f(\bar{x})]_{\#(A)-1} (\#(A)-1)^{t} \text{ for some } s \in Z.
\end{align*}\]

Hence the conclusion.

5) We omit a position of computer and a transition index of a map.
§3. Some elementary operations on Turing computers

We know various results on elementary operations on Turing machines in ordinary sense ([3]). Whether these results hold or do not hold in a Turing computer just defined in this paper must, however, be studied in detail. Now we shall study under what conditions and how the results hold in our case.

Let $A^{x}_{T_{c}}$ (or simply $A^{x}$ when confusion does not occur) be a subset of $A$ such that its element really appears as an element in a transition diagram of the Turing computer $[T_{c}]$ under consideration. The following lemma is easily obtained from lemma 2.1.

**Lemma 3.1.** If an expression $E$ is in $S(T_{c})$ and $T_{c}(E)=k^{r}$ for some $r \in Z (\neq 0)$, then there exists a Turing transformation $T_{c}^{*}$ such that $T_{c}^{*}(E)=k$.

A transformation $T_{c}^{*}$ is called a regular extension of $T_{c}$.

**Proposition 3.1.** Let $\#(A^{x}) \leq \#(A)-2$. If a function $f: D (\subseteq N') \rightarrow N$ is (partially) $C$-computable, then it is strongly (partially) $C$-computable.

**Proof.** When $D=D_{1}=\{-1, 0, +1\}$, the proposition was proved by M. Davis ([1], [3]). Here we shall prove the general case. Firstly we consider the following Turing computer $[T^{(1)}]$ whose transition diagram is shown in Figure 3. In this diagram, let $d=\{d_{i}\} \subseteq D$ and let $a_{2}$ and $a_{r}$ be two elements which do not belong to $A^{x}$. Such elements exist by assumption. We write symbolically by $\tilde{a}_{t}$ ($t \neq 0, 1$) an element $E \in \tilde{A}^{x}$ such that $E(i)=a_{0}(i \neq 0)$ and $=a_{0}(i=0)$. Then we have

![Diagram](Fig. 3)
(3.1) \[ T^{i+1}(\overline{x}) = \{(\overline{a}_1)^d \circ \overline{a}_2 \circ (\overline{a}_x)^{i-d} \} \] for all \( x \in N^i \),
where
\[ (\overline{a}_x)^i = \overline{a}_3 \circ \overline{a}_2 \circ \cdots \circ \overline{a}_1. \]

By assumption there exists a Turing transformation \( T_c \) such that \( f(x) = \langle T_c(\overline{x}) \rangle \). Let \( [T_c^{(w)}] \) be a Turing computer, the transition diagram of which consists of the transition diagram of \([T_c] \), when the transition diagrams shown in Figure 4 are added to each state \( q_i (i = 1, 2, \ldots, |Q_T|) \) of the computer \([T_c] \).

In that figure \( \omega \) is the order of a state-set of the computer \([T_c] \).

Let \( \overline{a}_x \) and \( \overline{a}_n \), be, respectively, a set which does not contain \( a_1 \) and \( a_x \).

Let \( S^{(1)} \equiv S(\overline{a}_x; T_c^{(w)}) \). Then we have
\[
S^{(1)} = \left\{ \emptyset \right\}, \quad S^{(2)} = \left\{ \overline{a}_3 \right\}^{(i-1)}, \quad S^{(3)} = \left\{ (d-2) \right\}^{(i-1)}.
\]

![Fig. 4](image-url)

---

Seiiti Huzino
Turing transformation and strong computability of Turing computers

\[ S^{(3+2d)} = \left\{ (\overline{a}_2)^{d+1} \right\}_0^{(4-d)} \]

\[ S^{(5+2d)} = \left\{ (\overline{a}_2)^{d+1} \right\}_0^{(4-d)} \]

\[ S^{(6+2d)} = \left\{ (\overline{a}_2)^{d} \right\}_0^{(4-d)} \]

Similarly we get

\[ S^{(8+2d)}(\overline{a}_r) ; T_c^{(2)} = \left\{ (\overline{a}_r)^{d} \right\}_0^{(1)} \]

Thus it follows that

\[ T_c^{(3)} \left( \left( (\overline{a}_2)^{d} \circ \overline{x} \circ (\overline{a}_r)^{d} \right)^{(4-d)} \right) = \left[ (\overline{a}_2)^{d} \circ \left( T_c(\overline{x}) \right)^{(r)} \circ \left( (\overline{a}_r)^{d} \right)^{(s)} \right]^{(1)} \]

for some \( r, s, t \geq 0 \) \((t \neq 0)\).

Next we construct a Turing computer \([T^{(3)}]\) shown in Figure 5, where \( a \) represents all elements except \( a_0, a_1, a_2 \) and \( a_r \). Then we have

\[ T_c^{(3)} \left( \left( (\overline{a}_2)^{d} \circ \left( T_c(\overline{x}) \right)^{(r)} \circ \left( (\overline{a}_r)^{d} \right)^{(s)} \right)^{(1)} \right) = f(x) - 1^{(r)} \]

for some \( l \in Z \).

To prove the proposition it is sufficient to take \( T_c^{(3)} = T_c^{(3)} \land T_c^{(2)} \land T^{(1)} \), since in this case lemma 3.1 holds if we take a regular extension of \( T_c^{(3)} \), if necessary.

![Fig. 5](image)

**Proposition 3.2.** If there exists a Turing transformation \( T_c^* \) such that \( T_c^* (\overline{x}) = \overline{y} \) \((x \in N^t, y \in N^s)\), then there exists a Turing transformation \( T_c \) such that

\[ T_c (\overline{z} \circ \overline{x}) = \{ \overline{z} \circ \overline{y} \}^{(r)} \]

for some \( r \in Z \) and \( z \in N^k \) \((\langle \overline{z} \rangle \geq d)\), where \( d = \max_i \left\{ \| d_i \| \right\} (d_i \in D_{T_c}) \).

**Proof.** M. Davis has proved the proposition without the condition \( \langle \overline{z} \rangle \geq d \) in the case when \( D = D_1 = \{-1, 0, +1\} \) \(([3])\). We shall prove it under the condition \( \langle \overline{z} \rangle \geq d \), without which the conclusion will not hold in
the general case. We construct a Turing computer \([T^{(D)}]\) shown in Figure 6.

Fig. 6

Fig. 7
Then it follows that for \( z = (z_1, z_2, \ldots, z_h) \)
\[
(3.6) \quad T^{(2)}(\overline{z} \oplus \overline{x}) = \left\{ \overline{a}_1 \circ (\overline{a}_p)^{\nu_1} \circ \overline{a}_2 \circ (\overline{a}_p)^{\nu_2} \circ \cdots \circ \overline{a}_r \circ (\overline{a}_p)^{\nu_r} \circ \overline{a}_s \circ \overline{x} \right\}_{\overline{x} = \overline{x}^{\sum_{i=1}^h \nu_i}}.
\]

Let \([T_c^{(2)}]\) be a Turing computer which consists of the computer \([T_c^{(1)}]\) and a part of "recopying mechanism" associated with each state \(q_i\) of the computer \([T_c^{(1)}]\). The transition diagram of a recopying mechanism of \([T_c^{(2)}]\) is shown in Figure 7. Then we have
\[
(3.7) \quad T_c^{(2)}(\{ \overline{a}_1 \circ (\overline{a}_p)^{\nu_1} \circ \overline{a}_2 \circ (\overline{a}_p)^{\nu_2} \circ \cdots \circ \overline{a}_r \circ (\overline{a}_p)^{\nu_r} \circ \overline{a}_s \circ \overline{x} \}) = \left\{ \overline{a}_1 \circ (\overline{a}_p)^{\nu_1} \circ \overline{a}_2 \circ (\overline{a}_p)^{\nu_2} \circ \cdots \circ \overline{a}_r \circ (\overline{a}_p)^{\nu_r} \circ \overline{a}_s \circ \overline{y} \right\}_{(c)}^{(r)}.
\]
for some $r$ and $s \geq 0 \ (s = \sum_{i=1}^{k} z_{i} + 2k + r)$.

Next we construct a Turing computer $[T^{(3)}]$, the transition diagram of which is shown in Figure 8. Then it follows that

$$T^{(3)} \Bigg\{ a_{1} \circ (a_{s})y_{1} \circ a_{r} \circ (a_{s})y_{2} \circ a_{r} \circ \cdots \circ a_{r} \circ (a_{s})y_{k+1} \circ a_{s} \ast \bar{y}_{(r)} \Bigg\}_{(s)}^{(r)} = \left[ \bar{z} \ast \bar{y}_{(r)} \right]_{\left( \sum_{i=1}^{k} + 2k \right)}^{(r)}.$$

Hence the conclusion follows if we take a Turing transformation $T_{e} \equiv T^{(3)} \land T_{c}^{(2)} \land T^{(1)}$.

**Corollary 3.2.1.** If a function $f : \mathcal{D}(\subset N^{i}) \rightarrow N$ is C-computable and $y$ is an expression such that $y \in N^{s}$ and $<y> \geq d \ (d = \max \{ |d_{i}|, d \in D \}$, then there exists a Turing transformation $T_{e}$ such that

$$T_{e}(y \ast \bar{x}) = (y \ast (f(x) - 1))^{(r)} \text{ for some } r \geq 0.$$

The next propositions which M. Davis has proved are important for our later development, and the proofs run similarly in our case, so we write them here omitting the proof (3]).

**Proposition 3.3.** If there exists a Turing transformation $T_{c}^{(r)}$ such that $T_{c}^{(r)}(\bar{x}) = y$ for $x \in N^{i}$ and $y \in N^{s}$, then there exists a Turing transformation $T_{e}$ such that $T_{e}(\bar{x}) = (y \ast \bar{x})^{(r)}$ for some $r \geq 0$.

**Corollary 3.3.1.** If a function $f : N^{i} \rightarrow N$ is C-computable, then there exists a Turing transformation $T_{e}$ such that

$$T_{e}(\bar{x}) = (f(x) - 1) \ast \bar{x}^{(r)} \text{ for some } r \geq 0.$$

**Proposition 3.4.** If functions $f_{i} : N^{i} \rightarrow N \ (i = 1, 2, \ldots, s)$ are (partially) C-computable, then there exists a Turing transformation $T_{e}$ such that

$$T_{e}(\bar{x}) = (f_{1}(x) - 1) \ast (f_{2}(x) - 1) \ast \cdots \ast (f_{s}(x) - 1)^{(r)} \text{ for some } r \in \mathbb{Z}.$$

Here we assume that $<f_{i}(x) - 1> \geq d$ for all $x \in N^{i}$.

Lastly from propositions 2.1 and 3.1 it follows:

**Proposition 3.5.** If a function $f : \mathcal{D}(\subset N^{i}) \rightarrow N$ satisfying $f(x) \neq 0 \text{ for all } x \in \mathcal{D}$ is partially C-computable, then it is directly $(\mathcal{D}(\mathcal{D}) - 1, C)$-computable.

§4. Some results on computability and decision mechanism

**Proposition 4.1.** Let functions $f_{i} : \mathcal{D}(f_{i}) \subset N^{i} \rightarrow N$ be partially
C-computable and let $\mathcal{D}(f_i) = \mathcal{D}$ (i=1, 2, ..., s). Let a function $M: \mathcal{D} \rightarrow N$ be such that $M(x) = \max_{i=1,2,...,s} \{f_i(x)\}$ for all $x \in \mathcal{D}$. If $<f_i(x) - 1> \geq d$ for all $x \in \mathcal{D}$ and some $i$, then there exists a Turing transformation $T_c$ such that $T_c(\overline{x}) = \overline{M(x) - 1}^{(r)}$ ($r \geq 0$), i.e. $M(x)$ is also strongly partially C-computable.

Proof. Let $M_i(x) = \max_{i=1,2,...,s} \{f_i(x)\}$. Without loss of generality we can assume that $<f_i(x) - 1> \geq d$ for all $x \in \mathcal{D}$.

When $s=1$, the existence of a Turing transformation $T_c$ such that $T_c(\overline{x}) = \overline{M_1(x) - 1}^{(r)}$ ($r \geq 0$) follows from proposition 3.1.

Assume that the conclusion holds for $s=k$. Now we shall show that the conclusion holds also for $s=k+1$. By assumption there exists a Turing transformation $T_c^{(1)}$ such that $T_c^{(1)}(\overline{x}) = \overline{M_k(x) - 1}^{(r_1)}$ ($r_1 \geq 0$), therefore, by lemma 2.1 there exists a Turing transformation $T_c^{(1)}$ having the property $T_c^{(1)}(\overline{x}) = \overline{M_k(x) - 1}$.

Fig. 9
A function $f_{k+1}$ being $C$-computable, by lemma 2.1 and proposition 3.2 there exists a Turing transformation $\tilde{T}_c(3)$ such that
\begin{equation}
\tilde{T}_c(3)(M_k(x)-1 \ast \tilde{x}) = M_k(x)-1 \ast f_{k+1}(x)-1,
\end{equation}
since $<M_k(x)-1 \geq f_{k+1}(x)-1 \geq d$ for all $k=1, 2, \cdots, s$.

And the existence of a Turing transformation $T_c(2)$ such that $T_c(2)(\tilde{x}) = (M_k(x)-1 \ast \tilde{x})^{e_2}(r_2 \in Z)$ follows from proposition 3.3, since $\tilde{T}_c(4)$ exists. Using lemma 2.1 again, we can verify the existence of a Turing transformation $\tilde{T}_c(2)$ such that $\tilde{T}_c(2)(\tilde{x}) = M_k(x)-1 \ast \tilde{x}$. Construct a Turing computer $[T_c(4)]$, the transition diagram of which is shown in Figure 9.

Let $\mathcal{S}(\tilde{x} \ast y ; T_c(4)) (x, y \in N)$. Firstly suppose that $x \geq y$.

Then we have the following instantaneous expressions:
\begin{align*}
\mathcal{S}(1) &= \left\{ \bar{a}_3 \circ x-1 \circ y \right\}_0, \\
\mathcal{S}(2+y+6) &= \left\{ \bar{a}_3 \circ x-1 \ast y-1 \circ \bar{a}_i \right\}_{x+y+6}, \\
\mathcal{S}(2(x+y)+10) &= \left\{ (\bar{a}_3)^2 \circ (x-2) \ast \bar{a}_i \right\}_1, \\
\mathcal{S}(2(x+y)+13) &= \left\{ (\bar{a}_3)^2 \circ (x-2) \ast (\bar{a}_i)^2 \right\}_{x+y+13}, \\
\mathcal{S}(4(x+y)+15) &= \left\{ (\bar{a}_3)^3 \circ (x-3) \ast (\bar{a}_i)^2 \right\}_{x+y+15}, \\
\mathcal{S}(5(x+y)+16) &= \left\{ (\bar{a}_3)^5 \circ (x-3) \ast (\bar{a}_i)^3 \right\}_{x+y+16}, \\
& \quad \cdots \\
\mathcal{S}(2xy+x+10y+6) &= \left\{ (\bar{a}_3)^{y+1} \circ (x-y) \ast (\bar{a}_i)^{y+1} \right\}_{x+y+6}, \\
\mathcal{S}(2xy+2x+9y+10) &= \left\{ (\bar{a}_3)^{y+2} \circ (x-y) \ast (\bar{a}_i)^{y+1} \right\}_{x+y+10}, \\
\mathcal{S}(2xy+4x+10y+16) &= \left\{ (\bar{a}_3)^{y+2} \circ (x-y) \ast \bar{a}_i \right\}_{x+y+16}, \\
\mathcal{S}(2xy+5x+12y+20) &= \left\{ \bar{a}_i \right\}_{x+y+20}.
\end{align*}

If $x \leq y$, it holds that
\begin{align*}
\mathcal{S}(p)(\tilde{x} \ast y ; T_c(4)) &= \left\{ y \right\}_{x+y+2}^{(q)} \text{ for some } p \text{ and } t \in N,
\end{align*}
which can be shown in the same way as above.

Then we have the relation
(4.2) \[ T^*_e(M_0(x)-1 \ast f_{e+1}(x)-1) = M_0(x)+(f_{e+1}(x)-M_0(x)) \cdot 1^{(r_e)}(r_e \geq 0) = M_{e+1}(x)-1^{(r_e)} \]

Hence the conclusion follows easily if we take \( T_e \equiv T^*_e(4) \cap \bar{T}^*_e(3) \cap \bar{T}^*_e(2) \).

**Corollary 4.1.1.** Let functions \( f_i(i=1, 2, \cdots, s) \) satisfy the assumptions in proposition 4.1. Let \( m(x) = \min_{i=1, 2, \cdots, s} \{ f_i(x) \} \). Then \( m(x) \) is strongly (partially) C-computable.

**Proposition 4.2.** Let \( f_i: \Xi \rightarrow N(i=1, 2, \cdots) \) be partially C-computable functions and \( \langle f_i(x)-1 \rangle \geq d \), where \( d = \max \{|d_i|/d_i \in D_1 \cup D_2\} \). Then there exists a Turing computer \([ T_d ]\) such that the Turing transformation \( T_e \) induced from it has the property:

(4.3) \[ T_e(\bar{x}) = \begin{cases} a_1 & \text{(if } f_1(x) \equiv f_2(x)) \\ \phi & \text{(if } f_1(x) < f_2(x)) \end{cases} \]

**Proof.** By proposition 3.4, there exists a Turing transformation \( T^{(1)}_e \) such that \( T^{(1)}_e(\bar{x}) = \langle f_1(x)-1 \ast f_2(x)-1 \rangle^{(r)}(r \in Z) \). By an analogous proof as in lemma 2.1, we can prove the existence of a Turing computer \([ T^{*}_e ^*] \) such that the Turing transformation induced from it has the property:

(4.4) \[ T^{*}_e(\langle f_1(x)-1 \ast f_2(x)-1 \rangle^{(r)} ) = f_1(x)-1 \ast f_2(x)-1 \]

We consider a Turing computer \([ T^{(2)}_e ]\), the transition diagram of which is shown in Figure 10. Let \( S^{(2)} = [ x ^{(-)} \ast y ; T^{(2)}_e ] \) \( (x \geq y, x, y \in N) \). Then we have the following instantaneous expressions:

\[ S^{(2)} = \left\{ \begin{array}{l} x^{(-)} \ast y \end{array} \right\}_1 \]
\[
\begin{align*}
\mathcal{S}(x+y+6) &= \left\{ \frac{x-1 \ast y-1}{x+y+2} \right\}_{1}^{(1)}, \\
\mathcal{S}(2(x+y)+11) &= \left\{ \frac{x-2 \ast y-1}{x+y+2} \right\}_{2}^{(2)}, \\
\mathcal{S}(3(x+y)+13) &= \left\{ \frac{x-2 \ast y-2}{x+y+2} \right\}_{1}^{(3)}, \\
\mathcal{S}(4(x+y)+16) &= \left\{ \frac{x-3 \ast y-2}{x+y+2} \right\}_{3}^{(3)}, \\
&\ldots \\
\mathcal{S}(2xy+11y+2) &= \left\{ \frac{x-y-1 \ast 0}{x+y+2} \right\}_{y+1}^{(y+1)}, \\
\mathcal{S}(2xy+x+10y+6) &= \left\{ \frac{x-y-1}{x+y+2} \right\}_{x+1}^{(y+1)}, \\
\mathcal{S}(2xy+3x+8y+9) &= \left\{ \frac{\bar{a}_1}{x+y} \right\}_{y}^{(x)},
\end{align*}
\]

And if \( x < y \), we get the relations
\[
\begin{align*}
\mathcal{S}(2xy+13x-2y-11) &= \left\{ 0 \ast y-x+1 \right\}_{x}^{(x)}, \\
\mathcal{S}(2xy+12x-y-9) &= \left\{ 0 \ast y-x \right\}_{y}^{(x)}, \\
\mathcal{S}(2xy+11x+1) &= \left\{ y-x \right\}_{x+1}^{(x+2)}, \\
\mathcal{S}(2xy+10x+y+2) &= \left\{ y-x-1 \right\}_{x+2}^{(x+2)}, \\
\mathcal{S}(2xy+7x+4y+10) &= \left\{ \phi \right\}.
\end{align*}
\]

Thus it follows that
\[
(4.5) \quad \mathcal{T}_c(\bar{x}) \left( f_1(x) - 1 \ast f_2(x) - 1 \right) = \begin{cases} \bar{a}_1 & (f_1(x) \geq f_2(x)) \\ \phi & (f_1(x) < f_2(x)) \end{cases}.
\]

**Proposition 4.3.** Under the assumptions of the proposition 4.2, there exists a Turing computer \([T_c]\) such that the Turing transformation induced from it has the property:
\[
(4.6) \quad \mathcal{T}_c(\bar{x}) = \begin{cases} \bar{a}_1 & \text{if } f_1(x) = f_2(x) \\ \phi & \text{if } f_1(x) \neq f_2(x) \end{cases}.
\]

We can prove the above proposition analogously as in proposition 4.2.

**Proposition 4.4.** If functions \( f_i : \mathbb{N}^i \rightarrow \mathbb{N} (i = 1, 2, \ldots, s) \) are \( C \)-computable and \( \langle f_i(x) - 1 \rangle \geq d \), then there exists a Turing transformation
Turing transformation and strong computability of Turing computers

\( T_e \) such that \( T_e(i \ast x) = f_i(x) - 1^{cr_i} \) for some \( r_i \in \mathbb{Z} \) \((i = 1, 2, \ldots, s)\).

**Proof.** From assumptions and proposition 3.4 there exists a Turing computer \([T_e^{(1)}]\) such that \( T_e^{(1)}(x) = \left( \bigcap_{j=1}^{s} f_j(x) - 1 \right)^{cr_i} (r \in \mathbb{Z})\).

![Diagram](image-url)
We can prove easily the existence of two Turing computers \([T_e^{(2)}]\) and \([T_e^{(3)}]\) such that
\[
T_e^{(2)}(x \ast y) = \{x + d \ast y\}^{(t)} \quad (t \in Z, \; x \in N, \; y \in N'),
\]
\[
T_e^{(3)}(x + d \ast y) = \{x \ast y\}^{(p)} \quad (p \in Z, \; x \in N, \; y \in N'),
\]
where \(d = \max \{d_j \mid D_j \in \bigcup_{i=1}^s D_i\}\).

Now the existence of a Turing computer \([T_e^{(4)}]\) such that
\[
T_e^{(4)}(i + d \ast x) = \{i + d \ast \Pi_{j=1}^s (\ast f_j(x) - 1)\}^{(q)} \quad (q \geq 0)
\]
can be proved by the existence of \(T_e^{(1)}\) and proposition 3.2.

Consider a Turing computer \([T_e^{(5)}]\) shown in Figure 11. Then it follows that
\[
T_e^{(5)} \land T_e^{(3)} \land T_e^{(4)} \land T_e^{(2)}(i \ast x) = \{i \ast f_i(x) - 1\}^{(p')} (r' \geq 0, \; r'' \in Z).
\]

We can verify easily the existence of a Turing computer \([T_e^{(6)}]\) such that
\[
T_e^{(6)}(\{i \ast (f_i(x) - 1)\}^{(p''}) = \{f_i(x) - 1\}^{(p''')} (r''' \in Z). \; \text{Hence the required transformation is} \; T_e^{(6)} \land T_e^{(4)} \land T_e^{(3)} \land T_e^{(4)} \land T_e^{(2)}.
\]

References


