Multicontrol Turing machines

By Gy. Révész

Multitape and multihead Turing machines are well-known generalizations of the basic model. All these generalizations have the common feature that they have a single finite state control and thus, they work in a synchronous manner. That is, the finite state control device coordinates the moves of the read-write heads so that they work at the same speed [1].

The new model introduced in the present paper can be considered as an abstract model of multi-processor systems, where the working speeds of the individual processors are independent from each-other. The only restriction is the exclusion of a symultaneous acces to the same storage location, i.e., each storage location can be accessed by only one processor at a time. This limitation is quite reasonable whenever more than one processors share the storage device. The model is asynchronous as there is no other connection between the individual processors except the common storage device through which they can pass information.

Suppose we have two Turing machines sharing a single tape (see Figure 1).

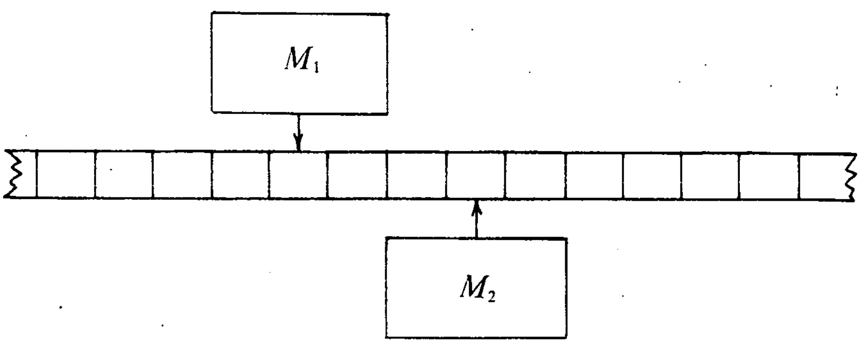

Figure 1

Each finite state control works at its own speed that may also vary in the course of the computation. If they attempt to access the same square on the tape one of them will be delayed until the other completes one step. If this step does not change the position of the corresponding read-write head then another choice is made for the finite control to be the next. The choice can be made in many different ways depending on some preassigned priorities or on the basis of some probabilities, etc. Here we are not concerned with the details of how to make this choice, but we are interested in the computational power of the general model. We will show that the computational power of multicontrol Turing machines is the same as that of the basic model. For this purpose we shall give first our formal definitions.

Definition. A single-control mondeterministic Turing machine is a quintuple $T=(K, Z, q_0, B, M)$, where K is a finite nonvoid set of *internal states*, Z is a finite nonvoid set of *tape symbols*, $q_0 \in K$ is the *initial state*, $B \in Z$ is the *blank symbol*, M is a mapping from $K \times Z$ into the subsets of $K \times (Z-\{B\}) \times \{-1, 0, +1\}$, called *moves*.

Definition. A configuration of a single-control nondeterministic Turing machine is a word XqY, where X and Y are words over $Z-\{B\}$ and $q \in K$.

The configuration denotes the nonblank portion of the tape, the actual position of the read-write head, and the actual internal state of the finite control.

The symbol scanned by the read-write head in this configuration is the first symbol of Y, or B if Y is the empty word.

A move of the Turing machine will change its configuration in the following steps.

a) The internal state of the finite state control is changed.

b) A symbol of $Z-\{B\}$ is printed on the tape in place of the scanned symbol.

c) The read-write head moves one square to the left, remains unchanged, or moves one square to the right as expressed by the values -1, 0, or $+1$, respectively.

The transformation of the configuration induced by the mapping M defines the relation $\Rightarrow$ such that $X_1q_1Y_1 \Rightarrow X_2q_2Y_2$ iff there is a move in M that changes the configuration $X_1q_1Y_1$ into $X_2q_2Y_2$. In fact, the mapping M can be given as a set of rewriting rules. Namely.

(i) $qy \to pz \in M$ iff $(p, z, 0) \in M(q, y)$,

(ii) $xqy \to pxz \in M$ for all $x \in Z$ iff $(p, z, -1) \in M(q, y)$,

(iii) $qy \to zp \in M$ iff $(p, z, +1) \in N(q, y)$.

The reflexive and transitive closure of the relation $\Rightarrow$ will be denoted as usual by $\stackrel{*}{\Rightarrow}$.

Definition. A configuration XqY is final if for the first symbol y of Y the set $M(q, y)=\emptyset$, or if Y is the empty word and $M(q, B)=\emptyset$.

Definition. A computation of a Turing machine is a sequence of moves $q_0P \stackrel{*}{\Rightarrow} XqY$, where q_0 is the initial state, P is a word over Z, and XqY is final. The input value of the computation is P while the output is represented by XY, the final contents of the storage tape.

Definition. A two-control mondeterministic Turing machine is an 8-tuple $T=(K', K'', Z, q_0', q_0'', B, M', M'')$, where $T'=(K', Z, q_0', B, M')$ and $T''=(K'', Z, q_0'', B, M'')$ are single-control Turing machines such that $K' \cap K''=\emptyset$.

Definition. A configuration of a two-control Turing machine is a word $Xq'Uq''Y$, where X, U, Y are words over $Z-\{B\}$, $q'\in K'$ and $q''\in K''$.

The relations $\Rightarrow$ and $\overset{*}{\Rightarrow}$ can be defined in a similar fashion as in the case of single-control Turing machines, taking into account that a symbol of K' will be adjacent to a symbol of K'' whenever the two read-write heads are scanning the same square. In such cases either of the two controls (but only one of them) may perform the next move as if the other were not there. Otherwise they work parallel and do not disturb each-other. Now we will show the following.

Theorem. Every computation performed by a two-control Turing machine can be performed by a single-control one.

Proof. In order to prove this theorem we will simulate the computations of a two-control Turing machine with the aid of a single-control one.

The essential feature of the simulation is that the parallel moves of the two control devices will be performed in a serial manner such that only one of them will be activated at a time while the other will be frozen. But the active and the frozen status can be exchanged between them any time that makes the simulation of every parallel computation possible.

Let $T_2=(K', K'', Z, q_0', q_0'', B, M', M'')$ be a two-control Turing machine. Then let $T_1=(K, Z_1, q_0, B, M)$ be defined such that

$$K=((K'\cup\{1,2\})\times(K''\cup\{1,2\}))\cup\{L, R, \lrcorner, Я, F, ꟻ, H\},$$

$$Z_1=Z\cup((K'\cup K'')\times Z).$$

Internal states of the form $[q', q'']$ represent the coincidence of the two read-write heads. A pair of the form $[q', 1]$ or $[q'', 1]$, ($[q', 2]$ or $[q'', 2]$) means that the corresponding control device is acive and it is currently to the left (to the right) of the other. The meanings of the special state symbols are the following:

L (R): activating is passing over to the left (right),
$\lrcorner$ (Я): activating is passing over to the left (right) leaving a final configuration frozen behind,
F (ꟻ): completely final configuration obtained on the left (right),
H: simulation halted.

The actual state of the frozen control device will be encoded onto the tape as a tape symbol of the form $[q', y]$ or $[q'', y]$. Now we have to specify the mapping M.

1) $M([q', q''], y)=\emptyset$ iff $M'(q', y)=M''(q'', y)=\emptyset$, and

$M([q', q''], y)=M([q'', q'], y)$ for all $q'\in K', q''\in K''$ and $y\in Z$,

2a) $([p', q''], z, 0)\in M([q', q''], y)$ iff $(p', z, 0)\in M'(q', y)$,

2b) $([q', p''], z, 0)\in M([q', q''], y)$ iff $(p'', z, 0)\in M''(q'', y)$.

From here on a pair of related specifications like 2a and 2b will be given by describing just the first of them.

3*a*) $([p', 1], [q'', z], -1) \in M([q', q''], y)$ iff $(p', z, -1) \in M'(q', y)$.

4*a*) $([p', 2], [q'', z], +1) \in M([q', q''], y)$ iff $(p', z, +1) \in M'(q', y)$.

5*a*) For $i = 1, 2$ and $j = -1, 0, +1$,

$([p', i], z, j) \in M([q', i], y)$ iff $(p', z, j) \in M'(q', y)$.

6*a*) $M([q', 1], [q'', y]) = M([q', 2], [q'', y]) = M([q', q''], y)$

for all $q' \in K', q'' \in K''$ and $y \in Z$.

7*a*) $(R, [q', y], +1) \in M([q', 1], y)$ and $(L, [q', y], -1) \in M([q', 2], y)$

iff $M(q', y) \neq \emptyset$.

8) $(R, y, +1) M(R, y)$ and $(L, y, -1) \in M(L, y)$ for all $y \in Z$.

9*a*) $(\text{Я}, [q', y], +1) \in M([q', 1], y)$ and $(\text{⅃}, [q', y], -1) \in M([q', 2], y)$

iff $M(q', y) = \emptyset$.

10) $(\text{Я}, y, +1) \in M(\text{Я}, y)$ and $(\text{⅃}, y, -1) \in M(\text{⅃}, y)$ for all $y \in Z$.

11*a*) $([q', 2], y, 0) \in M(R, [q', y])$ and $([q', 1], y, 0) \in M(L, [q', y])$

for all $q' \in K'$ and $y \in Z$.

12*a*) $([q', 2], y, 0) \in M(\text{Я}, [q', y])$ and $([q', 1], y, 0) \in M(\text{⅃}, [q', y])$

iff $M'(q', y) \neq \emptyset$.

13*a*) $(\text{ꟻ}, y, -1) \in M(\text{Я}, [q', y])$ and $(F, y, +1) \in M(\text{⅃}, [q', y])$

iff $M'(q', y) = \emptyset$.

14) $(\text{ꟻ}, y, -1) \in M(\text{ꟻ}, y)$ and $(F, y, +1) \in M(F, y)$ for all $y \in Z$.

15) $(H, y, 0) \in M(\text{ꟻ}, [q', y])$ and $(H, y, 0) \in M(F, [q', y])$

for all $q' \in K'$ and $y \in Z$.

Now let us see how T_1 simulates T_2. Suppose we have a computation of T_2 starting with a configuration $q_0' q_0'' P$ and ending with a final configuration $Xq'Uq''Y$. Then T_1 will be started with $[q_0' q_0''] P$. As long as the two read-write heads are scanning the same square the simulation is guaranted by specifications 1, 2*a* and 2*b*. As soon as they are parted one of them becomes active while the other becomes frozen (3*a*—*b*, 4*a*—*b*) and the control device of T_1 follows exactly the actions of the active control of T_2 (5*a*—*b*).

If the frozen control is encountered by the active one, either of them will be enabled to proceed (6*a*—*b*). The simulation of the active control may be suspended any time by T_1 so that T_1 switches over to the other. This is realized by T_1 with the aid of special states $R, L,$ Я and ⅃ which cause a search for the frozen control on the tape. T_1 always remembers the correct direction for the search via the state-suffix 1 or 2. (see 7*a*—*b*, 9*a*—*b*, 8, 10). The search is ended by activating the previously frozen control device (11*a*—*b*, 12*a*—*b*) unless both of them happen to be in a final configuration. In the latter case the frozen state will be cancelled from the tape and a message will be sent back to cancel the other encoded state as well and to stop the simulation (13*a*—*b*, 14, 15). If the final configuration of T_2 is of the form $Xq'q''Y$ then the simulation in T_1 will be finished with the configuration $X[q', q'']Y$. In this case the resulting tape inscription of T_1 will be exactly the same as that of T_2, but in other cases we must first get rid of the encoded final states (13*a*—*b*, 15).

The nondeterministic order of parallel steps in T_2 is properly simulated by the nondeterministic active — frozen status switching in T_1, and this completes the proof.

It can be observed that T_1 makes use of the same amount of tape squares as T_2 does. In particular, if T_2 is linearly bounded then so is T_1. On the other hand the above theorem can be extended to more than two control devices and thus, we have the following:

Corollary. Every asynchronous parallel computation performed by a multicontrol Turing machine can be also performed by a single-control Turing machine using the same amount of tape.

A number of interesting special cases of multicontrol Turing machines can be considered. One of them could be an abstract model of the so called pipeline processing where a streamlike information flow is processed simultaneously at different stages by several control units. The tape alphabet Z can be partitioned for this purpose in such a way that the input alphabet of each control unit forms a subset of the output alphabet of the previous one. This means that each control unit would work at full speed as long as the tape inscription permits.

Finally it should be mentioned that the simulation of asynchronous parallel processes was given above by a nondeterministic model even if the individual control units are deterministic. It is known that nondeterminism can be reduced to deterministic operation in case of Turing machines, but this would very likely require additional tape [2].

References

[1] Hartmanis, J. & R. E. Stearns, On the computational complexity of algorithms, *Trans. Amer. Math. Soc.*, v., 117, 1965, pp. 285—306.

[2] Savitch, W. J., Relationships between nondeterministic and deterministic tape complexities, *J. Comput. System Sci.*, v. 4, 1970, pp. 177—192.

[3] Stoss, H. J., k-Band-Simulation von k-Kopf-Turing-Maschinen, *Computing*, v. 6, 1970, pp. 309—317.

(Received May 12, 1976)