stichting
mathematisch
centrum

MC

AFDELING INFORMATICA
(DEPARTMENT OF COMPUTER SCIENCE)

IW 147/80 SEPTEMBER

P.M.B. VITANYI

REAL-TIME TURING MACHINES UNDER VARYING SPECIFICATIONS II

Preprint

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics subject classification: 68C40, 68C25, 68F10

ACM-Computing Reviews-categories: 5.23, 5.25, 5.26

Real-time Turing machines under varying specifications II *)

by

Paul M.B. Vitányi

ABSTRACT

In this note we give explicit numerical detail of the proof in VITÁNYI [1979,1980] that k+1 heads are more powerful than k heads as a storage device for real-time Turing machines. The proof technique is unusual in the induction phase, and has a wider range of applicability than the chosen counter example.

KEY WORDS & PHRASES: *complexity, real-time computation, multitape Turing machines, multihead Turing machines*

---

*) This report will be submitted for publication elsewhere.

APPENDIX TO IW 140/80

The proof that k+1 heads are more powerful than k heads as a storage device for real-time Turing machines is essentially the proof that appeared in VITÁNYI [1979]. As stated in Section 5, PAUL, SEIFERAS and SIMON [1980] gave, independently, a similar proof that k+1 push-down stores cannot be simulated in real-time by k-head Turing machines with head-to-head jumps either, thus strengthening the above result. They use a simpler type of induction than we did, but their proof is less generally applicable than the one we are concerned with here. (See also the last paragraph of this Appendix.) Although both proofs may seem adequate for their respective purposes, for these hairy arguments there is something dissatisfying in lack of explicitness. Real conviction is gained through spelling out the numerical detail. Therefore, as an exercise, and to pacify readers with a taste for rigour, we supply full details for the proof of Theorem 2.1 below.

Recall that AANDERAA [1974] demonstrated that no k-tape RTTM can recognize A_{k+1} or, equivalently, no k-tape Turing machine can simulate k+1 push-down stores in real-time. Although he, as is customary, defines k-tape Turing machines starting with blank storage tapes, nowhere in his proof use is made of that fact. Therefore, his argument works as well for k-tape Turing machines with initially inscribed tapes, which is what we need. His result can be paraphrased as follows:

"There is a function $N: \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for any k-tape RTTM M having p states and q work tape symbols, whatever its initial tape contents, we can find a word w in Σ^*_{k+1} of length no more than $N(k+1,p,q)$ such that $w \in L(M)$ iff $w \notin A_{k+1}$.

On pp. 89-90 of the cited reference we find $N(k+1,p,q)$ defined by:

$$N(k+1,p,q) = r^r \cdot N_0(k+1,p,q); \tag{A1}$$

where

$$r = 32(k+1)k \cdot \rho^{k+3}, \tag{A2}$$

$$N_0(k+1,p,q) = 32(k+1)k^2 \cdot \theta^2 \cdot \rho^{2(k+1)} \cdot (\log_2 p+1)^2, \tag{A3}$$

with

(A4) $\rho = 8(k+1)k\cdot(\log_2 q+1)$,

(A5) $\theta = \rho^{-k}(\rho^{k+1}-1)/(\rho-1)$.

Therefore, from (A5):

(A6) $\theta^2\rho^{2(k+1)} = (\rho/(\rho-1))^2(\rho^{k+1}-1)^2 \leq \rho^{2k+4}$ (since $\rho > 1$).

From (A1), (A2), (A3) and (A6) it follows that it suffices to set

(A7) $N(k+1,p,q) = m(k+1,q)^{m(k+1,q)+2}\cdot(\log_2 p+1)^2$,

where by A(2) and (A4) we have

(A8) $m(k+1,q) = 4\cdot(8(k+1)k)^{k+4}\cdot(\log_2 q+1)^{k+3}$.

We now turn to the actual proof of Theorem 2.1. The proof shows that Aanderaa's languages A_{k+1} cannot be recognized by k-head RTTM's. It proceeds by induction on the number k of heads. The base case k = 1 follows from Aanderaa's result that A_2 cannot be recognized by a 1-tape (= 1-head) RTTM. In the induction phase, the truth of the theorem for k > 1 follows from the truth of the theorem for all values 1,2,...,k-1, rather than, as is more usual in induction arguments, from the truth of the theorem for just k-1. That this is a necessary nuisance follows from the fact that although A_k cannot be recognized by a (k-1)-head RTTM, this does not imply that in a k-head RTTM recognizing A_k all heads get *pairwise* arbitrary for apart. This we need, since we want to spring at the appropriate moment the recognition problem of A_{k+1} on the k-head RTTM and then use Aanderaa's result as if the k heads were on separate tapes. However, all k heads may be necessary to recognize A_k, but this still allows the case where at all times a pair of heads is close together. Superficially, it would seem that in such a case A_k could also be recognized by a (k-1)-head RTTM with "fat" heads, contradicting the induction assumption. This line of argument holds indeed for k = 1,2,3 but breaks down

for k = 4, as shown by Figure A1. Therefore, for each k > 1 we need to appeal to the truth of the Theorem for all values 1,2,...,k-1 so as to pry the heads pairwise far apart.

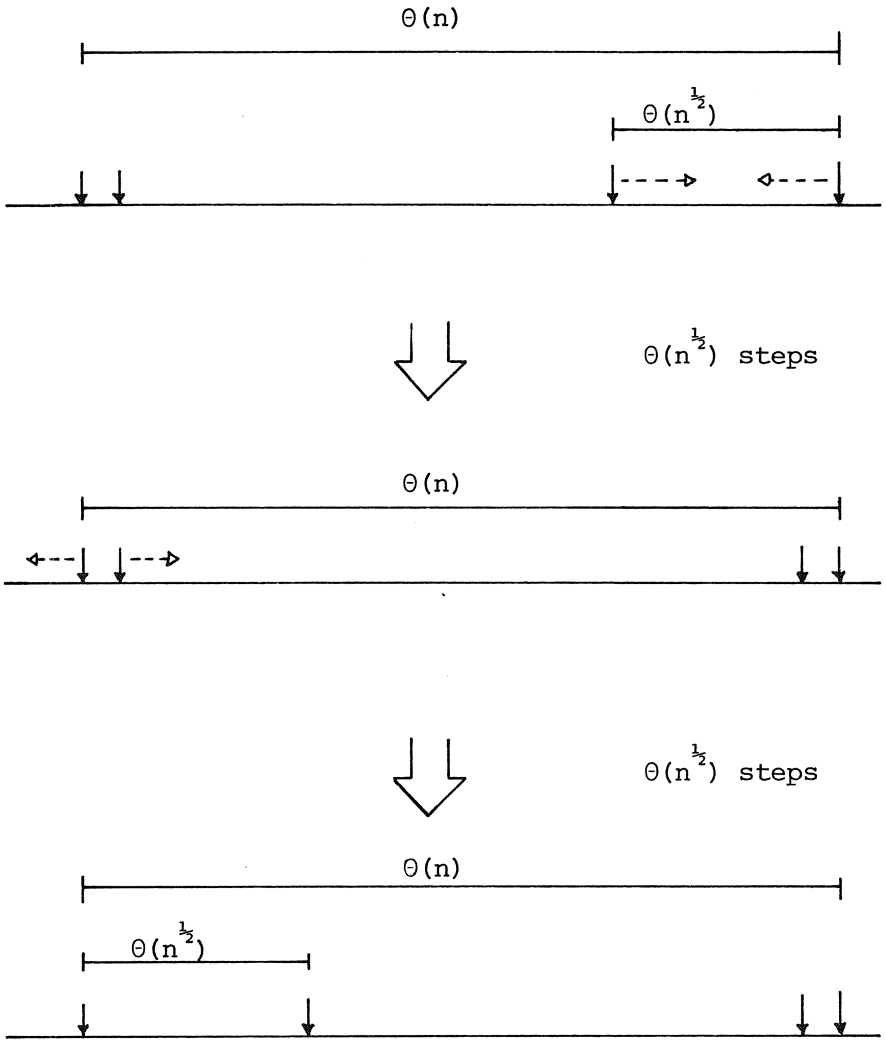

Figure A1. We cannot a priori assume that this behavior can be simulated by 3 "fat" heads.

Recall, that in the proof of Theorem 2.1 we presented the k-head RTTM M_k, alleged to recognize H_{k+1} (equivalently A_{k+1}), with a string of the form:

$$w = a_1^{(2)} a_2^{(2)} \ldots a_{n_2}^{(2)} * a_1^{(3)} a_2^{(3)} \ldots a_{n_3}^{(3)} * \ldots * a_1^{(k+1)} a_2^{(k+1)} \ldots a_{n_{k+1}}^{(k+1)}$$

$$= w_2 * w_3 * \ldots * w_{n_{k+1}},$$

such that w_i is over the alphabet of A_i, $2 \leq i \leq k+1$. During the processing of w_i, M_k must recognize A_i, $2 \leq i \leq k+1$. At each such stage i, it suffices to show that a (i-1)-head RTTM which is able to buffer in its finite control i-1 tape segments of length c_i, with k-i+1 heads distributed over these tape segments, will be fooled by some input word over Σ_i^* of length no more than n_i, where we choose $n_i = N(i,p_i,q_i)$ and p_i and q_i are the sizes of the state set and work tape alphabet, respectively, of the (i-1)-head RTTM $M_k^{(i)}$ thus constructed from M_k, see Figure A2. Clearly, we need not keep more than $\min\{i-1,k-i+1\} \leq \lceil k/2 \rceil$ of such tape segments in the finite control of $M_k^{(i)}$. In $M_k^{(i)}$ each head on the work tape t_i will scan the equivalent of two tape squares of M_k's tape, in between which the cut out tape segment can be thought to belong. The swapping of tape contents from the buffers in $M_k^{(i)}$'s finite control with tape t_i, so as to enable $M_k^{(i)}$ to continue simulating M_k's behavior whatever the head movements, is handled in the obvious way.

Let M_k have state set Q, tape t and work tape alphabet Γ. The machine $M_k^{(i)}$ then has state set Q_i, work tape t_i and work tape alphabet Γ_i defined as follows:

$$Q_i = Q \times \{0,1,\ldots,c_i-1\}^{k-i+1} \times \{1,2,\ldots,k\} \times \{0,1,\ldots,\lceil k/2 \rceil\} \times \Gamma^{c_i \lceil k/2 \rceil},$$

where Q is the original finite-state control of M_k; $\{0,1,\ldots,k\} \times \{0,1,\ldots,\lceil k/2 \rceil\}$ tells on which tape segment j, $0 \leq j \leq \lceil k/2 \rceil$, head i, $1 \leq i \leq k$, is positioned (tape segment 0 indicating that the head is not positioned on a buffer tape segment but on the real work tape t_i); $\{0,1,\ldots,c_i-1\}^{k-i+1}$

---

1) We denote the integer floor by $\lfloor \cdot \rfloor$ and the integer ceiling by $\lceil \cdot \rceil$.

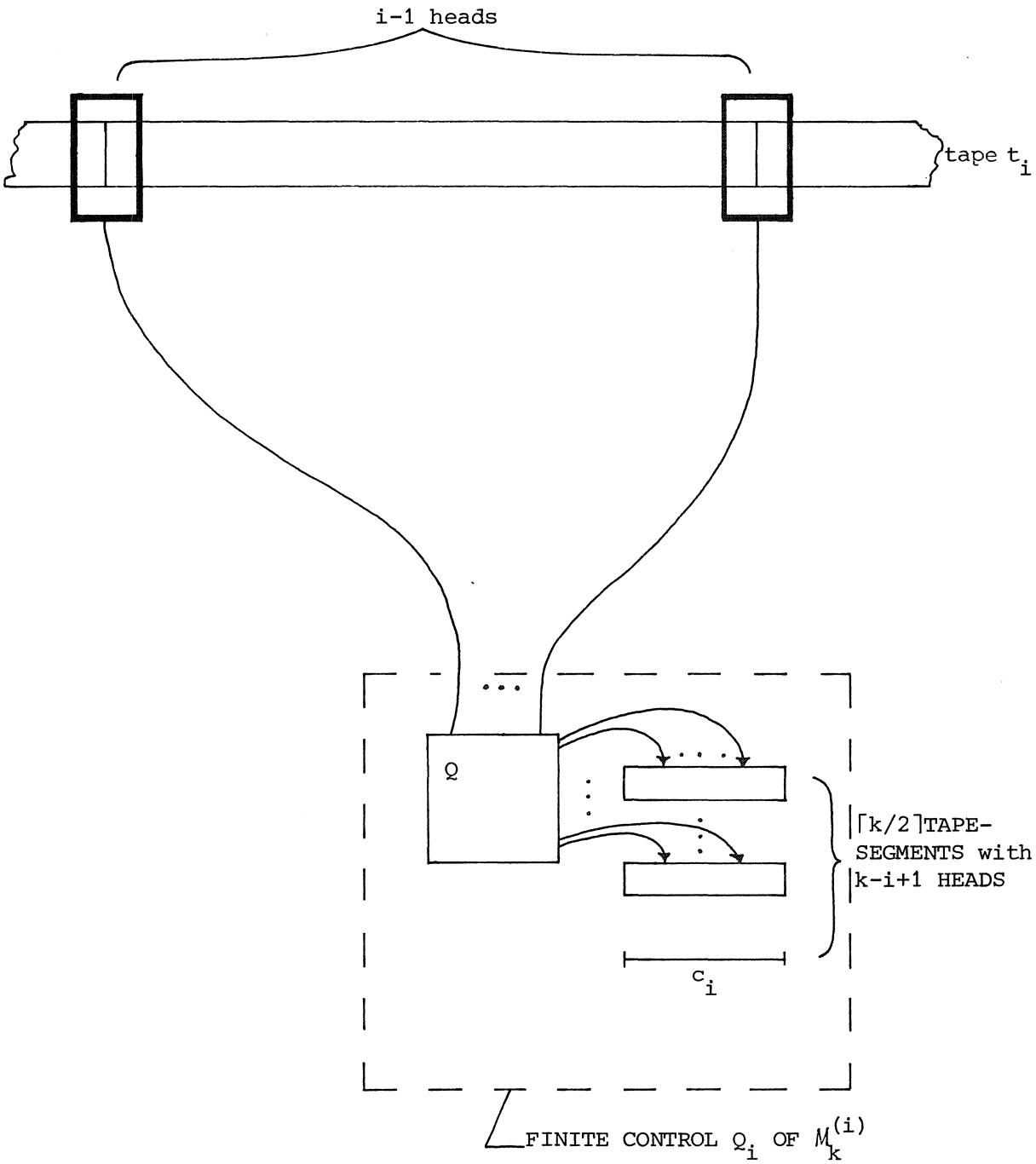

Figure A2. The construction of $M_k^{(i)}$ from M_k.

keeps track of the positions of the k-i+1 heads on the cut out tape segments; and $\Gamma^{c_i\lceil k/2\rceil}$ records the contents of the (at most) $\lceil k/2\rceil$ buffer tape segments of length c_i.

$$\Gamma_i = \Gamma \times \Gamma$$

So the number of states of $M_k^{(i)}$ is bounded above by any p_i' such that

$$p_i' \geq p \cdot c_i^{k-i+1} \cdot k \cdot \lceil k/2\rceil \cdot q^{c_i\lceil k/2\rceil}.$$

Assuming that for c_i large enough we have

$$\text{(A9)} \qquad 2 \cdot p \cdot c_i^{k-i+1} \cdot k \cdot \lceil k/2\rceil \leq q^{c_i\lfloor k/2\rfloor},$$

we observe that

$$\text{(A10)} \qquad p_i = q^{c_i k}/2$$

suffices as the number of states of $M_k^{(i)}$, $2 \leq i \leq k$.

Recall that the conditions and inequalities in the proof of Theorem 2.1 are certainly fulfilled if

(A11) n_i is large enough to fool $M_k^{(i)}$ for some input word over Σ_i of length not greater than n_i, $2 \leq i \leq k+1$;

and

$$\text{(A12)} \qquad c_i \geq 2k \sum_{j=i+1}^{k+1} (n_j+1), \qquad 2 \leq i \leq k.$$

By (A7), (A11), (A12) and the construction of $M_k^{(i)}$, $2 \leq i \leq k$, and setting $n_{k+1} = N(k+1,p,q)$, we can compute $n_{k+1}, n_k, \ldots, n_2$, in that order, by setting

$$\text{(A13)} \qquad n_i = N(i,p_i,q^2), \qquad 2 \leq i \leq k.$$

By (A12) and noting that $n_{i+1} + 1 > \Sigma_{j=i+2}^{k+1} (n_j+1)$ we see that the following

choice of c_i suffices:

(A14) $$c_i = 5kn_{i+1}, \qquad 2 \le i \le k.$$

Now, we have, for $2 \le i \le k$,

(A15)
$$\begin{aligned} n_i &= N(i,p_i,q^2) && \text{(by (A13))} \\ &= m(i,q^2)^{m(i,q^2)+2}\cdot(\log_2 p_i+1)^2 && \text{(by (A7))} \\ &= m(i,q^2)^{m(i,q^2)+2}\cdot(c_i k)^2\cdot(\log_2 q)^2 && \text{(by (A10))} \\ &= m(i,q^2)^{m(i,q^2)+2}\cdot(5\cdot k^2)^2\cdot n_{i+1}^2\cdot(\log_2 q)^2 && \text{(by (A14));} \end{aligned}$$

and

(A16)
$$\begin{aligned} n_{k+1} &= N(k+1,p,q) \\ &= m(k+1,q)^{m(k+1,q)+2}\cdot(\log_2 p+1)^2. \end{aligned}$$

We are now in the position to estimate a crude upper bound on the size of n_2, and, since $n_2 \ge \Sigma_{j=3}^{k+1}(n_j+1)$, on the size of n, the length of a word in Σ_{k+1}^* which is certain to fool M_k.

First note

$$m(k+1,q^2) \ge \max_{2\le i\le k+1}\{m(i,q),m(i,q^2)\}.$$

Set

(A17) $$\alpha = m(k+1,q^2)^{m(k+1,q^2)+2}\cdot(5\cdot k^2)^2\cdot(\log_2 q)^2;$$

then it follows from (A15) and (A16) that

(A18) $$n \le 2\cdot n_2 \le 2\cdot\alpha^{2^k-1}\cdot(\log_2 p+1)^{2^k}.$$

Hence we have that M_k is fooled by a word over the alphabet of H_{k+1} (equivalently A_{k+1}) of length not exceeding

$$n = 2 \cdot \alpha^{2^k - 1} \cdot (\log_2 p+1)^{2^k},$$

assuming that we have chosen c_i large enough that (A9) is satisfied, $2 \leq i \leq k$.

PAUL, SEIFERAS and SIMON [1980] noted that a proof similar to the one of Theorem 2.1 can be used to show that A_{k+1} cannot be recognized by k-head RTTMs with head-to-head jumps. Contrary to us, they establish in the induction step the truth of the proposition for $k > 1$ from the truth of the proposition for k-1 alone. In their case this works for two reasons: first because the recognition of A_{k+1} implies the simulation of k+1 pushdown stores and popping and pushing are inverse operations; second because of the ability of the jump Turing machine to jump. Our method does not depend on particular features of Aanderaa's languages, but holds more generally for any sequence of languages $L_2, L_3, \ldots, L_{k+1}, \ldots$ such that L_{k+1} cannot be recognized by a k-tape RTTM with initially inscribed tapes. More precisely,

THEOREM. *Let* $L = L_2, L_3 \ldots$ *be a sequence of languages for which there is a total function* $N_L: \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *such that for every* k*-tape* RTTM M*, with initially inscribed tapes and state set of size* p *and work tape alphabet of size* q*,* M *is fooled by an input word of length not exceeding* $N_L(k+1,p,q)$ *when it tries to recognize* L_{k+1}*, then, for the sequence of languages* $H = H_2, H_3, \ldots, H_{k+1}, \ldots$, $H_{i+1} = H_i \cup H_i * L_{i+1}$ *and* $H_2 = L_2$*, we can find a total function* $N_H: \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *such that for every* k*-head* RTTM *with state set of size* p *and work tape alphabet of size* q*, the machine is fooled by an input word of length not exceeding* $N_H(k+1,p,q)$ *when it tries to recognize* H_{k+1}.

REFERENCES

AANDERAA, S.O. (1974), *On* k*-tape versus* (k-1)*-tape real-time computation,* SIAM-AMS Proceedings, Vol. 7 (Complexity of Computation), 75-96.

PAUL, W.J., J.I. SEIFERAS & J. SIMON (1980), *An information-theoretic approach to time bounds for on-line computation,* Techn. Rept. TR 64, Department of Computer Science, University of Rochester, New York, February 1980.

VITÁNYI, P.M.B. (1979), *Multihead and multitape real-time Turing machines,* Techn. Rept. IW 111, Mathematisch Centrum, Amsterdam, June 1979.

VITÁNYI, P.M.B. (1980), *Real-time Turing machines under varying specifications,* Techn. Rept. IW 140, Mathematisch Centrum, Amsterdam, July 1980.