Information Processing Letters 19 (1984) 31-35
North-Holland 26 July 1984

ON THE POWER OF REAL-TIME TWO-WAY MULTIHEAD FINITE AUTOMATA WITH JUMPS

Walter J. SAVITCH *

Department of Electrical Engineering & Computer Sciences, University of California, San Diego, La Jolla, CA 92093, U.S.A.

Paul M.B. VITÂNYI

Department of Computer Science, Mathematical Centre, 1098 SJ Amsterdam, The Netherlands

Communicated by K. Mehlhorn
Received 2 January 1984

We investigate the relative power of jumps, nondeterminism, and number of heads for real-time finite automata. Results include showing that jumps add power that cannot be compensated for by nondeterminism and more heads. We also show that k+1 heads are more powerful than k heads, even if the finite automaton is allowed head to head jumps.

Keywords: Finite automata, jumps, multihead, real time

CR Categories: 5.23, 5.25, 5.26

1. Introduction

Computations of 1-way multihead finite automata have been considered by Yao and Rivest [8]. They show that k + 1 heads are better than k heads for both the deterministic and the nondeterministic versions of the machine. Furthermore, they show that the k-head nondeterministic variety is strictly more powerful than the k-head deterministic one. Janiga [1] studied the analogous questions for 2-way real-time multihead deterministic, respectively nondeterministic, finite automata, from now on called 2DRTFA and 2NRTFA, respectively. He obtained, mutatis mutandis, the same results for the 2-way real-time machines as did Yao and Rivest for the 1-way (no time limit) variety. Whereas the latter used 'palindrome like' arrangements of $\binom{k}{2}$ substrings to obtain their result, for the 2-way real-time case Janiga employed strings of k palindromes. To be more precise, let PALM be the set of palindromes in $\{0,1\}^*\{2\}\{0,1\}^*$. Let $P_k = (\text{PALM}\{*\})^k$. Then P_k is recognized by a (k + 1)-head 2DRTFA but not by any k-head 2NRTFA. $\{0,1,2,*\}^* - P_k$ is accepted by a 2-head 2NRTFA but not by any k-head 2DRTFA. Kosaraju [2] has shown that the jump Turing machine as defined by Savitch and Vitânyi [6] can be simulated in real-time by multitape Turing machines. A jump Turing machine has multiple heads on its one storage tape and each head can be shifted in one step to the position scanned by any other head, irrespective of the distance in between. So Kosaraju's result says that the computational power of real-time Turing machines is invariant under placing all of the heads on the same (storage) tape and adding the head-to-head jump option. Here we show that for 2-way multihead finite automata the head-to-head jump facility does extend the class of languages accepted in real-time. This result follows as a corollary of Janiga's work. However, the proof presented here is substantially simpler. We then go on to observe that for 2-way real-time multihead finite automata, the jump option cannot be compensated for by adding more heads and nondeterminism. An extra

* This research has been supported in part by NSF Grant MCS-8001963.

head cannot be compensated for by adding jumps, nondeterminism, and bidirectionality. Nondeterminism cannot be compensated for by adding extra heads and jumps. With respect to real-time 2-way multihead finite automata it is shown that k + 1 heads are better than k. For precise definitions of the devices and the addressed issues we refer the reader to the references.

2. Jumps vs. no jumps

We give an example language which is acceptable in real-time by 2-way 2-head finite automata with jumps, but not by any real-time 2-way multihead finite automaton without jumps. In the following, let $h:\{0, 1, \bar{0}, \bar{1}\}^* \to \{0, 1\}^*$ be a homomorphism which is defined by $h(\bar{a}) = h(a) = a$ for $a \in \{0, 1\}$.

$$L_0 = \left\{\bar{w}\,\bar{v}\,\bar{a}\,av^R \mid \bar{w}\,\bar{v} \in \{0, 1, \bar{0}, \bar{1}\}^*,\right.$$
$$\left. v \in \{0, 1\}^*, a \in \{0, 1\}, h(\bar{v}) = v\right\}.$$

The reader will easily figure out more complicated examples along these lines.

Lemma 1. L_0 *is accepted by a deterministic real-time 2-way 2-head finite automaton with jumps.*

Proof. Let M be a 2-way 2-head finite automaton with jumps as follows. The front head reads from left to right one letter at a time. Whenever this first head reads a barred letter it calls the second head to its present position. This second head starts reading from right to left one letter at a time. So M is able to recognize L_0. □

Lemma 2. L_0 *is not accepted by any deterministic real-time 2-way multihead finite automaton without jumps.*

Proof. Suppose the lemma is false. Let k be the minimum number such that L_0 is recognized by a k-head real-time 2-way finite automaton M_k but not by any (k − 1)-head one. Since L_0 is not regular, such a k must be greater than 1. Since M_k is real-time, there must be at least one head which moves right at each step. For each constant c we can find an input word w such that, during the processing of w by M_k, some head lags behind the vanguard head more than c squares. If this were not so, then all heads are at all times with c squares of the vanguard head, and we could replace M_k by an ordinary finite automaton with a larger finite-state control which keeps track of the symbols under the k − 1 nonvanguard heads of the simulated machine. This would imply that L_0 is regular, which would be a contradiction. Since by assumption L_0 is not recognizable by a (k − 1)-head real-time 2-way finite automaton, for each constant c we can find an input word w such that, during the processing of w by M_k, all k − 1 heads lag behind the vanguard head more than c squares. For suppose this were not the case. Since the vanguard head moves right at each step, at least one particular head must be at all times within c squares of the vanguard head, and similarly to the above, we would be able to replace M_k by a (k − 1)-head machine M_{k-1} with a finite-state control which also keeps track of the symbol under the neighboring head of the vanguard head. Contrary to the assumption, this would imply the falsehood of the lemma for k − 1.

So suppose that, subsequent to processing an input prefix, all other heads of M_k lag behind the vanguard head more than c squares, and the vanguard head now starts to read a suffix $u \in \{0, 1, \bar{0}, \bar{1}\}^*$, such that $|u| \leqslant c + 1$. In this situation, no other head of M_k will ever scan a symbol from u. Let the input prefix, which forces the k − 1 nonvanguard heads more than c squares behind the vanguard head, be z. At time $|z| + 1$, all these (k − 1)-heads scan specific letters of z. Set $\ell = (c/2) - 1$. We next consider how M_k will behave on suffixes chosen from the set $\{0, 1\}^{\ell}\,\bar{0}0\{0, 1\}^{\ell}$. The constant c is chosen to be even and to be large enough to complete the argument. The number of distinct positions on z which these k − 1 heads can reach, multiplied by the number of distinct states which the finite control can attain when the vanguard head crosses $\bar{0}$, is bounded above by $c^{k-1} \times n$, where n is the number of states of the finite control of M_k. The number of strings in $\{0, 1\}^{\ell}$ is 2^{ℓ}. If $2^{\ell} \geqslant c^{k-1} \times n$, which happens for c large enough, two distinct such strings, say u_1 and u_2, lead to the same instantaneous description of

M_k after processing zu_1 and zu_2. Therefore, M_k accepts either both $zu_1\bar{0}0u_1^R$ and $zu_2\bar{0}0u_1^R$ or rejects them both. Since $u_1 \neq u_2$ it follows that M_k does not accept L_1. □

Lemmas 1 and 2 immediately yield the following theorem.

Theorem 3. *There is a language recognized by a real-time 2-way 2-head deterministic finite automaton with jumps which is not recognized by any real-time 2-way deterministic multihead finite automaton without jumps.*

The language L_0 which witnesses Theorem 3 is quite simple and the proof of Theorem 3 is also fairly simple. By appealing to a result by Janiga (with a more complex proof), we observe that Theorem 3 can be strengthened to allow the machines without jumps to be nondeterministic as well. Recall the discussion in the Introduction and consider the language $P = \bigcup_{k-1}^{\infty} P_k$. It is easy to see that P is recognized by a 2-head 2DRTFA with jumps. However, Janiga [1] showed that P is not accepted by any multihead 2NRTFA without jumps. Hence we get the following theorem.

Theorem 4. *There are languages recognized by real-time 2-way 2-head deterministic finite automata with jumps which are not recognized by any real-time 2-way nondeterministic multihead automaton without jumps.*

3. Head count hierarchy for jump machines

We next show that the well-known maxim "k + 1 heads are better than k heads" remains true even if the real-time finite automaton is allowed to have head-to-head jumps. Indeed, this result indicates that almost nothing, including jumps and nondeterminism, can make up for the power of an extra head.

The witness languages for this head hierarchy are denoted S_k and their definition requires one other preliminary definition. For each $k \geqslant 1$, define a partial function f_k from k tuples of strings over the alphabet {0, 1} to strings over {0, 1}. Specifically, $f_k(x_1, x_2, \ldots, x_k) = z$ provided that $|x_1| = |x_2| = \cdots = |x_k| = |z|$ and, for $1 \leqslant i \leqslant |z|$, $(z)_i = \Sigma_{j-1}^{k}(x_j)_i \bmod 2$. Here $(w)_i$ denotes the *ith* digit of a string w. Less formally, if we regard each x_j and z as a vector of some number of 0's and 1's then $f_k(x_1, x_2, \ldots, x_k) = z$ provided that z is the bitwise mod 2 sum of all the vectors x_j. For each $k \geqslant 1$, define

$$\begin{aligned} S_k = \{x_1 * x_2 * \cdots * x_k * z \mid \\ &|x_1| = |x_2| = \cdots = |x_k| \text{ and} \\ &f_k(x_1, x_2, \ldots, x_k) = zy, \text{ for some } y\}. \end{aligned}$$

Theorem 5. *For any* $k \geqslant 1$, S_k *is accepted by a* (k + 1)*-head* 1DRTFA (*without jumps*) *but not by any* k*-head* 2NRTFA *with jumps.*

Proof. It is straightforward to produce a (k + 1)-head 1DRTFA which accepts S_k. So we need only show that no k-head 2NRTFA with jumps can accept S_k.

For the case k = 1, note that 1-head 2NRTFA's accept only regular sets, and S_1 is not regular.

Next suppose k > 1. Let M be a k-head 2NRTFA with jumps which is claimed, for purposes of deriving a contradiction, to accept S_k. Let ℓ be a fixed, sufficiently large, integer, We will consider how M computes in accepting computations on inputs of the form $x_1 0^{2\ell} * x_2 0^{2\ell} * \cdots * x_k 0^{2\ell} * z$ where z and all the x_i's are of length ℓ and $f(x_1 0^{2\ell}, x_2 0^{2\ell}, \ldots, x_k 0^{2\ell}) = z0^{2\ell}$. For each such input, we fix one accepting computation of M on that input and consider the configuration of M as the vanguard head reads the final $*$.

By a configuration we mean the state of the finite control and the position of the tape heads. When the vanguard reads the final $*$, M must be in one of $s * (k(3\ell + 1))^{k-1}$ configurations, where s is the number of states in the finite control of M. Set $c(\ell) = s * (k(3\ell + 1))^{k-1}$. There are $2^{k\ell}$ such inputs. So $2^{k\ell}/c(\ell)$ such inputs must leave M in the same configuration when the vanguard head reads the final $*$.

We now focus on these inputs and their fixed accepting computations, all of which leave M in the same configuration when the vanguard head reaches the last $*$. In this one configuration, some block i_0 ('some x_{i_0}') has all heads at least ℓ squares away from x_{i_0}. There are $2^{(k-1)\ell}$ choices for the other x_j, $j \neq i_0$, and $2^{k\ell}/c(\ell)$ inputs all together which leave M in this configuration. But, for sufficiently large ℓ, $2^{k\ell}/c(\ell) > 2^{(k-1)\ell}$. So, there must be two such inputs that differ only in block x_{i_0}. For notational convenience suppose $i_0 = 1$; the proof is similar for any other i_0. With $i_0 = 1$, there are two inputs:

$$x_1 0^{2\ell} * x_2 0^{2\ell} * \cdots * x_k 0^{2\ell} * z_1,$$

$$y_1 0^{2\ell} * x_2 0^{2\ell} * \cdots * x_k 0^{2\ell} * z_2,$$

such that: $x_1 \neq y_1$ and, in the accepting computations on these inputs, both computations leave M in the same configuration when the vanguard head reads the final $*$. Furthermore, in that configuration all heads are at least ℓ squares away from x_1 or y_1.

Since no head is within ℓ squares of x_1 or y_1, the string in that first position cannot effect the next ℓ moves of M in either computation. Hence, by a standard 'cut and paste' argument, M accepts

$$x_1 0^{\ell} * x_2 0^{\ell} * \cdots * x_k 0^{\ell} * z_2.$$

However, this is a contradiction since this string is not in S_k. □

4. Nondeterminism

For Theorem 5 we observe that the result that nondeterminism adds to the power of 2-RTFA holds for machines with jumps. The witness languages is $L = \{xzy * z^R \mid x, y, z \in \{0, 1\}^*\}$. It is easy to see that L is accepted by a 2-head 2NRTFA, even without jumps. However, Rosenberg [5] has shown that L is not accepted by any deterministic real-time Turing machine, and the results of Kosaraju [2] showed that a 2DRTFA with jumps can be simulated in real-time by a deterministic real-time Turing machine. Hence, L is not accepted by any 2DRTFA with jumps and so we get the following theorem.

Theorem 6. *There are languages accepted by 2-head* 2NRTFA (*without jumps*) *but not accepted by any multihead* 2DRTFA *with jumps.*

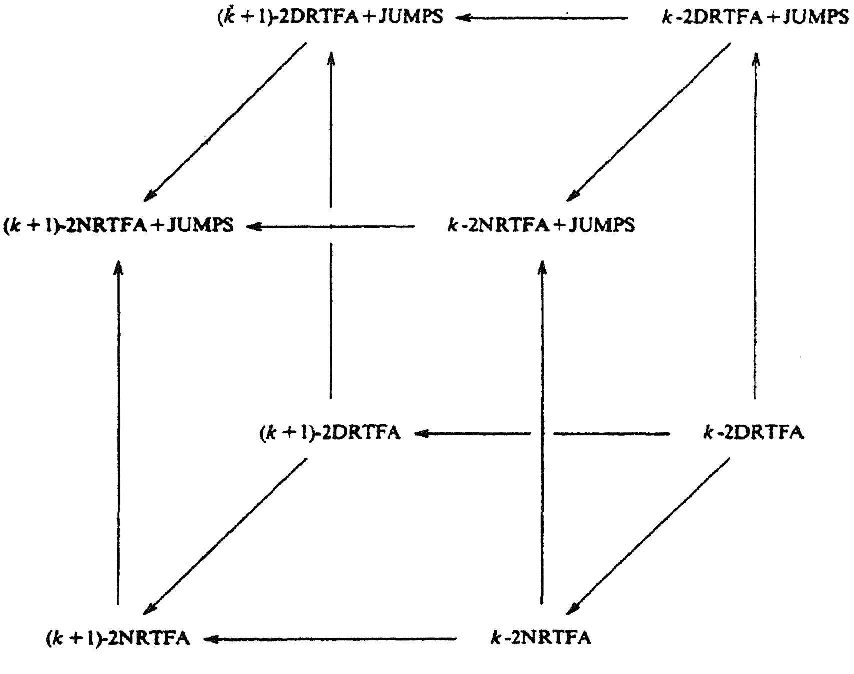

Fig. 1. Inclusion diagram for the computing power of real-time 2-way multihead finite automata according to number of heads, determinism, and jump option.

5. Summary

All results above hold whether or not we assume end markers or that the heads can detect coincidence.

Fig. 1 is a summary of the inclusions which hold for 2DRTFA's and 2NRTFA's. All inclusions are proper. Classes which are not connected by a sequence of directed arrows are incomparable. Hence we see that there are 3 distinct parameters: nondeterminism–determinism, jumps–no jumps, and the number of heads. Looking back at the theorems in more detail, we observe that jumps plus nondeterminism cannot make up for an additional head; additional heads plus nondeterminism cannot make up for jumps; and jumps plus additional heads cannot make up for nondeterminism.

References

[1] L. Janiga, Real-time computations of two-way multihead finite automata, in: L. Budach, ed., Fundamentals of Computation Theory (FCT'79) (Akademie Verlag, Berlin, 1979) pp. 214–218.

[2] S.R. Kosaraju, Real-time simulation of concatenable double-ended queues by double-ended queues, Proc. 11th Ann. ACM Symp. on Theory of Computing (1979) pp. 346–351.

[3] B.L. Leong and J.I. Seiferas, New real-time simulations of multihead tape units, Proc. 9th Ann. ACM Symp. on Theory of Computing (1977) pp. 239–248.

[4] M.O. Rabin, Real-time computation, Israel J. Math. 1 (1963) 203–211.

[5] A.L. Rosenberg, Real-time definable languages, J. ACM 14 (1967) 645–662.

[6] W.J. Savitch and P.M.B. Vitányi, Linear time simulation of multihead Turing machines with head-to-head jumps, Lecture Notes in Comput. Sci. (ICALP 4) 52 (Springer, Berlin, 1977) pp. 453–464.

[7] P.M.B. Vitányi, Multihead and multitape real-time Turing machines, Tech. Rept. IW 111, Mathematisch Centrum, Amsterdam, 1979.

[8] A. Yao and R. Rivest, $k+1$ heads are better than k, J. ACM 25 (1978) 337–340.