数理解析研究所講究録 1205 巻 2001 年 19-24

On the Power of Cooperating Systems of One-way Hybrid Finite Automata

山口大学 (Yamaguchi University)
王躍 (Wang Yue) 井上克司 (Katsushi Inoue) 伊藤暁 (Akira Ito)
山口東京理科大学 (Science University of Tokyo in Yamaguchi)
岡崎世雄 (Tokio Okazaki)

KeyWords: finite automaton, cooperating system, nondeterminism, complexity

1 Introduction

The concept of nondeterminism plays a fundamental role in automata theory, and there are some famous open problems regarding nondeterminsm, for example the *P* versus *NP* or the *DLBA* versus *NLBA* problem. In this paper we primarily consider nondeterminism in cooperating systems of one-way finite automata as a resource and study the power of cooperating system of one-way finite automata in terms of the number of nondeterministic finite automata in it.

The cooperating sysytems of finite automata may be considered as one of the simplest models of parallel computing systems: (in fact, any existing physical devices is finite, despite we often think in terms of models with infinite memory.) there are more than one finite automata and an input tape where these finite operate synchronously (in parallel) and can communicate with each other on the same cell of the input tape. More precisely, a *cooperating system of k finite automata*, $M = (FA_1, FA_2, \cdots, FA_k)$, cosists of k finite automata FA_1, FA_2, $\cdots$, FA_k, and a read-only input tape where these finite automata independently work step by step. Each step is assumed to require exactly one time for its completion. Those finite automata whose input heads scan the same cell of the input tape can communicate with each other, that is, every finite automaton is allowed to know internal states of other automata on the cell it is sacanning at the moment. The system M starts with each FA_i on the left endmarker ¢ in its inital state and accepts the input if each FA_i enters an accepting state and halts (when reading the right endmatker \$ of the input tape). (The reader is referred to [1] for the formal definition of a cooperating system of [one-way] finite automata.)

It has been shown that computational power of the cooperating system of one-way finite automata depends on the number of finite automata involved, and the (deterministic or nondeterministic) behavior of finite automata [1]. It is unknown, however, in general, whether computational power of the cooperating system of one-way finite automata depends on the number of nondeterministic finite automata involved. (Note that so far the finite automata considered in a cooperating system of finite automata are all deterministic or nondeterministic.) In order to investigate this problem, from now on, in general, we consider a cooperating system of finite automata in which some finite automata are deterministic and some finite automata are nondeterministic. We call it a *cooperating system of hybrid finite automata*, and investigate its power in terms of the number of nondeterministic finite automata in it.

In [2] it was actually shown that the power of cooperating systems of one deterministic one-way finite automata and one nondeterministic one-way finite automata are the same as that of systems of two nondeterministic one-way finite automata. (As a corollary the cooperating systems of one deterministic one-way finite automata and one nondeterministic one-way finite automata are more powerful than the the cooperating systems of two deterministic one-way finite automata.) The main result of this paper is as follows: for the cooperating systems of k hybrid one-way finite automata with $k \geq 3$ and for any $0 \leq j \leq k-2$, $j+1$ nondeterministic finite automata are better than j.

2 Definitions and notations

We say that the *speed* of a finite automaton in cooperating system of finite automata is $1/n$ if it moves its input head one cell every n steps. We assume that no two FA's in a finite automaton in cooperating system of finite automata have a common internal state. A *configuration* of a cooperating system of k finite automata is a $2k$-tuple $(a, \langle q_1, q_2, \cdots, q_k \rangle, t_1, t_2, \cdots, t_{k-1})$ where a is the symbol on the cell, c, read by the leading FA, $\langle q_1, q_2, \cdots, q_k \rangle$ are the internal states of FA's when FA's leave from the cell c (in the order $\langle q_1, q_2, \cdots, q_k \rangle$), respectively, and $t_1, t_2, \cdots, t_{k-1}$ are the difference between time when FA's leave from the cell c. (Clearly, the next move of a cooperating system of finite automata is only dependent on its current configuration.)

By $CS-k-FA(j)$ we denote a cooperating system of hybrid finite automata with j nondeterministic one-way finite automata and $k-j$ deterministic one-way finite automata (where $0 \leq j \leq k$).

By $L[CS-k-FA(j)]$ we denote the class of languages accepted by $CS-k-FA(j)$'s.

In our proof below we will use Kolmogorov-complexity (K-complexity). Informally, we define the K-complexity of a string w, denoted by $K(w)$, as the length of the shortest program that prints w (only). The conditional K-complexity of x with respect to y, denoted by $K(x|y)$, is the length of the shortest program which, with extra information y, prints x (only). A string w is called *random* if $K(w) \geq |w|-1$. Then, it is easy to see that more than half of strings are random [3]. A natural number n is called *random* if the binary notation of n (without leading zeros), $bin(n)$, is random. By $K(n)$ we denote the K-complexity of a natural number n. So it is easy to see that if a natural number n is random, then $K(n) \geq \log n - 1$. Note that for any random number n, there is a number m ($n \leq m \leq 2n$), for example, $bin(m) = 10^{|bin(n)|}$, such that $K(m) = O(\log\log n)$. (For formal definition of K-complexity, see [3])

3 Simulation of $CS-2-FA(2)$ by $CS-2-FA(1)$

It is obvious that a $CS-1-FA(1)$ (i.e., a nondeterministic finite automaton) can be simulated by a $CS-1-FA(0)$ (i.e., a deterministic finite automaton). Now we show that any $CS-2-FA(2)$ can be simulated by some $CS-2-FA(1)$. We conjecture that for any $k \geq 1$ $L[CS-k-FA(k-1)] = L[CS-k-FA(k)]$. Unfortunately, we cannot find a way at present, in general, to simulate any $CS-k-FA(k)$ by $CS-k-FA(k-1)$.

Lemma 1 ([1]) *Let M be a $CS-k-FA(j)$ $(0 \leq j \leq k)$. If w is any word accepted by M, then there exsits a computaion of M on w such that M accepts w at most $C|w|$ steps*[1]*, where C is a constant dependent only on M.*

Theorem 1 *For any $CS-2-FA(2)$, M, we can construct a $CS-2-FA(1)$ to simulate M.*

Proof: Let $M = (FA_1, FA_2)$ be a $CS-2-FA(2)$. We will construct a $CS-2-FA(1)$ $M' = (FA'_1, FA'_2)$ to simulate M.

Without loss of generality, we can assume that in any accepting computaion of M, FA_1 is the one that lastly leaves the each cell of input tape. Then, By Lemma 1, FA_1 can stay on each cell of the input tape at most C steps in an accepting computaion of M where C is some constant only dependent on M.

M' acts as follows:

1. FA'_1 deterministicly moves in speed $1/C$.
2. FA'_2 keeps track in its finite control of what states FA_1 and FA_2 may be in when they read each cell of the input tape. Furthermore, for each cell of the input tape, FA'_2 nondeterministicly adjusts its speed so that the interval between the times at which FA'_1 and FA'_2 leave the cell is the same as that FA_1 and FA_2 may do.

Clearly, it is easy to verify that M' can simulate M. □

4 Hierarchies based on the number of nondeterministic finite automata

In this section we investigate how the number of nondeterministic finite automata in a cooperating system of k hybrid one-way finite automata affects its accepting power, where $k \geq 3$.

Lemma 2 ([1]) *Let $T_1 = \{w_1 1 w_2 | w_1, w_2 \in \{0,1\}^* \ \& \ |w_1| = |w_2|\}$. Then, for any $k \geq 2$,*

1. $T_1 \in L[CS-k-FA(1)]$, *and*

2. $T_1 \notin L[CS-k-FA(0)]$.

Lemma 3 *For each $k \geq 3$ and each $1 \leq j \leq k-1$, Let $T_{k,j} = \{0^{i_1}10^{i_2}\cdots 10^{i_{k-1}}2 w_{11}0^{i_1}10^{i'_1}w_{12}2w_{21}0^{i_2}10^{i'_2}w_{22}\cdots 2w_{j1}0^{i_j}10^{i'_j}w_{j2}20^{i'_1}10^{i'_2}\cdots 10^{i'_j}10^{i_{j+1}}\cdots 10^{i_{k-1}} \mid w_{11}, w_{21}, \cdots, w_{j1} \in \{\{0\}^*1\}^* \ \& \ w_{12}, w_{22}, \cdots, w_{j2} \in \{1\{0\}^*\}^*\}$. Then,*

1. $T_{k,j} \in L[CS-k-FA(j)]$, *and*

2. $T_{k,j} \notin L[CS-k-FA(j-1)]$.

Proof:

1. The language $T_{k,j}$ is accepted by the $CS-k-FA(j)$ $M = (FA_1, FA_2, \cdots, FA_k)$ which acts as follows: Consider the case when an input tape

$$¢0^{i_1}10^{i_2}\cdots 10^{i_{k-1}}2w_1 2w_2\cdots 2w_j 20^{i'_1}10^{i'_2}\cdots 10^{i'_{k-1}}\$$$

is presented to M, where $w_1, w_2, \cdots, w_j \in \{\{0\}^*1\}^*\{0\}^*$. (Input words in the form different from the above can be easily rejected by M.)

[1] For any string w, let $|w|$ denote the length of w.

(a) FA_k (deterministicly) sweeps the input tape at the same speed $1/2$.

(b) For each $1 \leq h \leq j$, FA_h sweeps the subword 0^{i_h} at speed 1, nondeterministicly sweeps some subword $0^{l_h}10^{1'_h}$ of w_h (if exists) at speed $1/3$ for 0^{l_h} and at speed 1 for $0^{1'_h}$, sweeps the subword $0^{i'_h}$ at speed $1/3$, and sweeps the others at speed $1/2$.

(c) For each $j+1 \leq g \leq k-1$, FA_g (deterministicly) sweeps the subword 0^{i_g} at speed 1, sweeps the subword $0^{i'_g}$ at speed $1/3$, and sweeps the others at speed $1/2$.

(d) M accepts the input if and only if FA_h and FA_k scan the same cell just after FA_k sweeps the subwords 0^{l_h} and $0^{i'_h}$, respectively, for each $1 \leq h \leq j$, and FA_g and FA_k scan the same cell just after FA_k sweeps the subword $0^{i'_g}$ for each $j+1 \leq g \leq k-1$.

Note that for each $1 \leq h \leq j$, $i_h = l_h$ and $i'_h = l'_h$ if and only if FA_h and FA_k scan the same cell just after FA_k sweeps the subwords 0^{l_h} and $0^{i'_h}$, respectively, and for each $j+1 \leq g \leq k-1$, $i_g = i'_g$ if and only if FA_g and FA_k scan the same cell just after FA_k sweeps the subword $0^{i'_g}$. Thus M accepts an input words w if and only if $w \in T_{k,j}$.

2. We only give a proof for the case $k = 3$, and leave the others to the reader. (Although it seems not easy to do it, we believe that the techniques used in the proof can be generalized for the other cases.) Suppose that there is some $CS-3-FA(1)$ $M = (FA_1, FA_2, FA_3)$ which accepts $T_{3,2}$. For each $n \geq 1$, let

$$V_{3,2}(n) = \{0^{m_1}10^{m_2}20^{m'_1}10^{m'_3}10^{m''_1}10^{m''_3}20^{m'_2}10^{m'_4}10^{m''_2}10^{m''_4}20^{m_3}10^{m_4}\}$$

where $n \leq m_1, m_2, m_3, m_4, m'_1, m'_2, m'_3, m'_4, m''_1, m''_2, m''_3, m''_4 \leq Cn$ for some constant C. For a large enough n, we can select $m_1, m_2, m_3, m_4, m'_1, m'_2, m'_3, m'_4, m''_1, m''_2, m''_3, m''_4$ as follows:

(a) m_1, m_2, m_3, and m_4 are random.

(b) For any $1 \leq i, j \leq 4$ and $i \neq j$, $K(m_i|m_j) \geq K(m_i) - O(1)$.

(c) There is only one element in $\{m'_1, m''_1\}$ which is equal to m_1, and the other m satisfies the condition $K(m_i|m) \geq K(m_i) - O(1)$ for each $1 \leq i \leq 4$.

(d) There is only one element in $\{m'_2, m''_2\}$ which is equal to m_2, and the other m satisfies the condition $K(m_i|m) \geq K(m_i) - O(1)$ for each $1 \leq i \leq 4$.

(e) If $m'_1 = m_1$, then $m'_3 = m_3$, and m''_3 satisfies the condition $K(m_i|m''_3) \geq K(m_i) - O(1)$ for each $1 \leq i \leq 4$; If $m''_1 = m_1$, then $m''_3 = m_3$, and m'_3 satisfies the condition $K(m_i|m'_3) \geq K(m_i) - O(1)$ for each $1 \leq i \leq 4$.

(f) If $m'_2 = m_2$, then $m'_4 = m_4$, and m''_4 satisfies the condition $K(m_i|m''_4) \geq K(m_i) - O(1)$ for each $1 \leq i \leq 4$; If $m''_2 = m_2$, then $m''_4 = m_4$, and m'_4 satisfies the condition $K(m_i|m'_4) \geq K(m_i) - O(1)$ for each $1 \leq i \leq 4$.

So, each word w in $V_{3,2}(n)$ is in $T_{3,2}$, and is accepted by M. With each $w \in V_{3,2}(n)$, we associate one fixed accepting computation, $c(w)$, of M on

Statement 1 *For each $w \in V_{3,2}(n)$ above, there must exist two FA such that one of them is nondeterministic and the nondeterministic FA meets with another when it reads the subwords $0^{m'_1}10^{m'_3}10^{m''_1}10^{m''_3}$ and $0^{m'_2}10^{m'_4}10^{m''_2}10^{m''_4}$ of w, respectively, in the computaion $c(w)$.*

(Suppose that Satement 1 is not valid. That is, the nondeterministic FA_1 does not meet with either FA_2 or FA_3 when it reads the subwords $0^{m'_1}10^{m'_3}10^{m''_1}10^{m''_3}$ and $0^{m'_2}10^{m'_4}10^{m''_2}10^{m''_4}$. Let l_1 and l_2 be the distance between FA_1 and FA_2 and the distance between FA_2 and FA_3, respectively.

Clearly, since FA_2 and FA_3 move deterministicly, l_2 cannot be changed when FA_2 (or FA_3) reads the subword $0^{m'_1}$ or $0^{m'_2}$. Otherwise, there must exist a sufficiently large m'_1 ($\neq m_1$) or a sufficiently large m'_2 ($\neq m_2$) such that $K(l_2) \leq O(\log \log n)$ in some location of $0^{m'_1}$ (or $0^{m'_2}$). However, this would mean that

$$K(m_1) + K(m_2) \leq \log n + O(\log \log n),$$

or

$$K(m_2) + K(m_3) \leq \log n + O(\log \log n),$$

since one can reconstruct m_1, m_2 (or m_2, m_3) by simulating $c(w)$, given the information on that configuration of M. This contradicts our assumption $K(m_1) + K(m_2) \geq 2 \log n - O(1)$.

However, if l_2 is not changed when FA_2 (or FA_3) reads the subword $0^{m'_1}$ or $0^{m'_2}$, then by a simple 'cut and paste' argument (on $0^{m'_1}$ and $0^{m''_1}$), one can 'fool' M to accept some word not in $T_{3,2}$.)

Furthermore, we have

Statement 2 *FA_1 can only meet with another FA in some constant (dependent only on M) area of the input tape, $0^{m'_{12}}10^{m'_{31}}$ ($|0^{m'_{12}}10^{m'_{31}}| = O(1)$, where $0^{m'_{12}}$ is a suffix of $0^{m'_1}$ and $0^{m'_{31}}$ is a prefix of $0^{m'_3}$), located at the boundary of $0^{m'_1}$ and $0^{m'_3}$ if $m'_1 = m_1$, or $0^{m''_1}10^{m''_3}$ ($|0^{m''_{12}}10^{m''_{31}}| = O(1)$, where $^{m''_{12}}$ is a suffix of $0^{m''_1}$ and $0^{m''_{31}}$ is a prefix of $0^{m''_3}$), located at the boundary of $0^{m''_1}$ and $0^{m''_3}$ if $m''_1 = m_1$.*

(Suppose that Satement 2 is not valid. Then, for example, in the case $m'_1 = m_1$, there some suffix of $0^{m'_1}$, $^{m'_{12}}$ ($m'_{12} > O(1)$) such that FA_1 would meet with another FA when it begins to read $0^{m'_{12}}$. This would mean that $K(m'_{12}) + K(m_2) \leq \log n + O(1)$. Howevere, it is easy to see that $K(m'_{12}) + K(m_2) > \log n + O(1)$.)

Similarly, we also have

Statement 3 *FA_1 can only meet with another FA in some constant (dependent only on M) area of the input tape, $0^{m'_{22}}10^{m'_{41}}$ ($|0^{m'_{22}}10^{m'_{41}}| = O(1)$, where $0^{m'_{22}}$ is a suffix of $0^{m'_2}$ and $0^{m'_{41}}$ is a prefix of $0^{m'_4}$), located at the boundary of $0^{m'_2}$ and $0^{m'_4}$ if $m'_2 = m_2$, or $0^{m''_{22}}10^{m''_{41}}$ ($|0^{m''_{22}}10^{m''_{41}}| = O(1)$, where $0^{m''_{22}}$ is a suffix of $0^{m''_2}$ and $0^{m''_{41}}$ is a prefix of $0^{m''_4}$), located at the boundary of m''_2 and m''_4 if $m''_2 = m_2$.*

Thus, without loss of generality, we can assume that FA_1 meets with FA_2 in the subword $0^{m'_1}10^{m'_3}10^{m''_1}10^{m''_3}$.

It is easy to see that when FA_2 and FA_3 read the subword $0^{m'_2}$ in the computaion $c(w)$, their action will be deterministic (until one of them meets with FA_1). We consider the following three cases, when FA_2 and FA_3 read the subword $0^{m'_2}$:

(a) l_2 is not changed. For the case $m'_2 = m_2$, by Statements 2 and 3, this would mean that ($K(m_3|l_2) = O(1)$ and) $K(m_2|(m''_1, m''_3)) = O(\log\log n)$. However, by our assumption before, we have $K(m_2|(m''_1, m''_3)) \geq \log n - O(1)$. This is a contradiction.

(b) l_2 decreases. Then, there exists a sufficiently large m'_2 ($\neq m_2$) such that FA_2 meets with FA_3 in some location of the subword $0^{m'_2}$. However, this would mean that $K(m_2) + K(m_3) \leq \log n + O(\log\log n)$. This contradicts our assumption that $K(m_2) + K(m_3) \geq 2\log n - O(1)$.

(c) l_2 increases. Then, there exists a sufficiently large m'_2 ($\neq m_2$) such that l_2 must become $K(l_2) \leq O(\log\log n)$ sometime. Again, this would mean that $K(m_2) + K(m_3) \leq \log n + O(\log\log n)$. This contradicts our assumption that $K(m_2) + K(m_3) \geq 2\log n - O(1)$. □

From Lamma 3, we get the following theorem.

Theorem 2 *For any $k \geq 3$ and any $0 \leq j \leq k-1$, $L[CS-k-FA(j-1) \subsetneq L[CS-k-FA(j)]$.*

5 Concluding Remarks

From Lemma 3 (and the results in [1]), one can easily derive the following theorem.

Theorem 3 *For any $k \geq 2$ and any $0 \leq j \leq k$, $L[CS-k-FA(j)]$ is not closed under intersection.*

References

[1] Y. Wang, K. Inoue and I. Takanami, "Some propertis of cooperating systems of one-way finite automata", Trans. IEICE, vol.J75-D-I, pp.391-399, 1992.

[2] Y. Wang, K. Inoue and I. Takanami, "A note on one-way multicounter machines and cooperating systems of one-way finite automata", IEICE TRANS. INF. & SYST. vol.E76-D, pp.1302-1306, 1993.

[3] M. Li and P. Vitanyi, "Kolmogorov complexity and its applications", Handbook of Thepretical Computer Science, vol.A, pp.187-254, 1979.