6. Lower bounds

Objekttyp: **Chapter**

Zeitschrift: **L'Enseignement Mathématique**

Band (Jahr): **27 (1981)**

Heft 1-2: **L'ENSEIGNEMENT MATHÉMATIQUE**

PDF erstellt am: **29.05.2024**

6. Lower bounds

Definition. A *marked binary number* is a word over the alphabet $\{0, \mathit{0}, 1, \mathit{1}\}$ described by the regular expression $(0 \cup 1)^* \, \mathit{0}\, \mathit{1}^* \cup \mathit{1}^*$. The *value* of a marked binary number is given by the homomorphism h with $h(0) = h(\mathit{0}) = 0$ and $h(1) = h(\mathit{1}) = 1$, i.e. by disregarding the type of the digits.

Note: The digits in italics are those which will change their value when the marked binary number is increased by one.

Marked binary numbers allow the following local tests:

1. A word over the alphabet $\{0, \mathit{0}, 1, \mathit{1}\}$ is a marked binary number, iff the last digit is in italics and only the following adjacent pairs of digits occur:

   a) 00, 01, 0*0*, 10, 11, 1*0* (0, 1 or *0* behind 0 or 1), and

   b) *01*, *11* (*1* behind *0* or *1*).

2. For two right adjusted marked binary numbers x and y with y below x holds:

   value $(x) + 1 =$ value (y) iff only the following vertically adjacent pairs of digits occur:

   a) 0 or *0* below 0 or *1* and

   b) 1 or *1* below *0* or 1.

Theorem (Lower bound). *If a language L is accepted by a linear space bounded alternating Turing machine M, with at most q successors for each universal configuration, then L is polynomial time transformable to the set of satisfiable formulas of the monadic $\forall\exists^q$ class via length order $n \log n$.*

Proof. We can assume that M is a one-tape alternating Turing machine accepting L in space $n + 1$ and time $2^m - 1$ for an $m = O(n)$. We describe the case $q = 2$. To each input w of M, we define (using function symbols f_L and f_R) the functional form $F(w)$ of a formula $F'(w)$ of the monadic $\forall\exists^q$ class, such that:

Claim A : $w \in L$ iff $F(w)$ is satisfiable.

Before we define the formula $F(w)$, we show how to construct a structure α from an accepting computation tree, such that α will turn out to be a model of $F(w)$.

If w is accepted by M, then there is an accepting computation tree CT with the properties:

— Every node of the tree with depth less than $2^m - 1$ has exactly two sons, and every node with depth $2^m - 1$ is a leaf. I.e. it is a complete binary tree.

— If the same configuration appears in several nodes, then the corresponding successor configurations are the same.

Therefore, there are functions succ_L and succ_R, which define the instantaneous descriptions of the successor configurations in the tree. Furthermore, we can choose succ_L and succ_R in such a way that they have the following property:

For every pair consisting of a state and a scanned symbol, we consider the possible moves of M to be an ordered set.

If ID is a universal instantaneous description, then succ_L (ID) is the first and succ_R (ID) is the second successor of ID.

If ID is existential, then succ_L (ID) and succ_R (ID) are arbitrary successors of ID (typically succ_L (ID) $=$ succ_R (ID)).

If ID is accepting, then succ_L (ID) $=$ succ_R (ID) $=$ ID.

Given functions succ_L and succ_R and an accepting computation tree CT of depth $2^m - 1$ with the above properties, we define now the structure α, such that:

Claim B : α is a model of $F(W)$.

1. The universe $|\alpha|$ is the set $\{(t, \mathrm{ID}) \mid t$ is an integer with $0 \leqslant t \leqslant 2^m - 1$ and ID is the instantaneous description of a configuration occuring in a branch of the computation tree CT of M with input w at time $t\}$.
2. f_L (resp. f_R) is interpreted by a function mapping (t, ID) for $t < 2^m - 1$ to $(t+1, \mathrm{succ}_L(\mathrm{ID}))$ (resp. $(t+1, \mathrm{succ}_R(\mathrm{ID}))$) and $(2^m - 1, \mathrm{ID})$ to (0, start ID for input w). succ_L (ID) (succ_R (ID)) is defined to be the instantaneous description of the left (right) successor configuration of ID.
3. In (t, ID) the monadic predicates are interpreted as follows:

$$\text{Let } t = \sum_{i=0}^{m-1} b_i 2^i \text{ with } b_i \in \{0, 1\},$$

and let ID $= a_0 \dots a_{k-1} (a_k, q) a_{k+1} \dots a_n$ with $a_j \in \Sigma$ (alphabet) and $q \in Q$ (states).

Then the $O(m)$ monadic predicate symbols $B_j, M_j, Z, L_{j'}, S_p, T_{\sigma j}$ with $j \in \{0, \dots, m-1\}$, $j' \in \{0, \dots, n\}$, $p \in Q$ and $\sigma \in \Sigma$ are interpreted as

B_j^α $((t, \text{ID}))$ is true iff $b_j = 1$
M_j^α $((t, \text{ID}))$ is true iff $b_j = 1$ for all $i < j$, i.e. b_j is marked
Z^α $((t, \text{ID}))$ is true iff $b_i = 0$ for all i
$L_{j'}^\alpha$ $((t, \text{ID}))$ is true iff $j' = k$
S_p^α $((t, \text{ID}))$ is true iff $p = q$
$T_{\sigma j'}^\alpha$ $((t, \text{ID}))$ is true iff $a_{j'} = \sigma$

We now define the formula $F(w)$ and add some remarks about the intended meaning of its subformulas. This makes it obvious that claim B holds. $F(w)$ is the formula

$$\forall_y \big[F_H(y) \wedge F_V(y, f_L(y)) \wedge F_V(y, f_R(y)) \wedge F_0(y) \wedge F_U(y) \wedge F_W(y) \wedge F_L(y, f_L(y)) \wedge F_R(y, f_R(y)) \wedge F_A(y)\big]$$

where

a) $F_H(y)$ *is* $\bigwedge_{0 \leq j \leq m-2} [M_{j+1}(y) \leftrightarrow (M_j(y) \wedge B_j(y))] \wedge M_0(y)$

The intended meaning is:
All binary numbers are correctly marked. (H stands for horizontal condition.)

b) $F_V(y, z)$ is $\bigwedge_{0 \leq j \leq m-1} [B_j(z) \leftrightarrow (M_j(y) \leftrightarrow \neg B_j(y))]$

The intended meaning is:
The level number below level number l is $l + 1$. (V stands for vertical condition.)

c) $F_0(y)$ is $[\bigwedge_{0 \leq j \leq m-1} \neg B_j(y)] \leftrightarrow Z(y)$

The intended meaning is:
The configuration at level 0 is distinguished by Z.

d) $F_U(y)$ is

$$\Big[\bigwedge_{0 \leq j \leq n} \bigwedge_{\substack{\sigma, \sigma' \in \Sigma \\ \sigma \neq \sigma'}} \neg (T_{\sigma j}(y) \wedge T_{\sigma' j}(y))\Big] \wedge \Big[\bigwedge_{\substack{q, q' \in Q \\ q \neq q'}} \neg (S_q(y) \wedge S_{q'}(y))\Big]$$

The intended meaning is:

For every configuration there is at most one symbol in every tape cell, and the Turing machine is in at most one state.

e) $F_W(y)$ is $Z(y) \rightarrow [\bigwedge_{0 \leqq j \leqq n} T_{\sigma_j j}(y) \wedge L_0(y) \wedge \bigwedge_{1 \leqq j \leqq n} \neg L_j(y) \wedge S_{q_0}(y)]$

where $\sigma_0 \sigma_1 \ldots \sigma_n$ is wb (the input w extended by a blank endmarker B), and q_0 is the start state of M. This is the only subformula of $F(w)$ depending not only on $n = |w|$, but also on w. Its intended meaning is:
The distinguished configuration at level 0 is the start configuration.

f) Exactly as for nondeterministic Turing machines, it is possible to check if ID_1 is a successor of ID_0 by writing ID_1 below ID_0 and checking all 6-tuples seen through a window of length 3 and height 2 which is pushed over the two words, and by checking that no head of the Turing machine walks in or out of the tape portion represented by the instantaneous descriptions. In this way, we check

— for universal ID_0, if the left son is labeled with $\mathrm{succ}_L(\mathrm{ID}_0)$ and the right son is labeled with $\mathrm{succ}_R(\mathrm{ID}_0)$;

— for existential ID_0, just if both sons are labeled with any successors;

— for accepting ID_0, if both sons are labeled with ID_0.

It is easy to construct a formula $P_j^L(y, z)$ $(P_j^R(y, z))$ expressing the window condition at the positions $j, j+1, j+2$ for the ID's in node y and in its left (right) son z.

$P_j^L(y, z)$ and $P_j^R(y, z)$ are built from the atomic formulas

	$S_p(y), S_p(z)$	for $p \in Q$
and	$L_{j'}(y), L_{j'}(z)$	for $j' = j, j+1, j+2$
and	$T_{\sigma j'}(y), T_{\sigma j'}(z)$	for $j' = j, j+1, j+2$ and $\sigma \in \Sigma$.

The length of $P_j^L(y, z)$ and $P_j^R(y, z)$ are bounded by a constant times the maximal length of the atomic formulas.

For $D = L$ and $D = R$,

$$F_D(y, z) \text{ is } Z(z) \vee [\bigwedge_{0 \leqq j \leqq n-2} P_j^D(y, z)$$
$$\wedge (L_0(z) \rightarrow (L_0(y) \vee L_1(y))) \wedge (L_n(z) \rightarrow (L_{n-1}(y) \vee L_n(y)))] .$$

g) $$F_A(y) \text{ is } [B_{m-1}(y) \wedge M_{m-1}(y)] \to \bigvee_{q \in Q_a} S_q(y)$$

Q_a is the set of accepting states of M. The intended meaning of $F_A(y)$ is:
At the deepest level $2^m - 1$, all branches of the computation tree accept.

Now the formula $F(w)$ is defined, and for $w \in L$ it should be clear that $F(w)$ is satisfiable and has the model α.

We still have to show the other direction of claim A. If $F(w)$ is satisfiable, then $w \in L$. Let α be a model of $F(w)$. In α the formula

$$\forall y [F_H(y) \wedge F_V(y, f_L(y)) \wedge F_V(y, f_R(y))]$$

is valid. Hence for all $b \in |\alpha|$ a level number $l(b)$ is defined by the interpretation of the predicate symbols B_j in α. The level numbers have the property

$$l(f_L^\alpha(b)) = l(f_R^\alpha(b)) = l(b) + 1 \bmod 2^m .$$

Therefore (as $|\alpha|$ is non-empty), there are elements of all levels mod 2^m, in particular, there is an element b_0 of level 0.
Because $\forall y [F_0(y) \wedge F_w(y)]$ is valid in α, the truth values of the predicates L_j^α, S_p^α and $T_{\sigma j}^\alpha$ in b_0 encode the start configuration of the alternating Turing machine M with input w.
Let $|\alpha|'$ be the subset of $|\alpha|$ which is accessible from b_0 by several applications of f_L^α and f_R^α. Then the validity of

$$\forall y [F_U(y) \wedge F_L(y, f_L(y)) \wedge F_R(y, f_R(y))]$$

in α ensures that the predicates L_j^α, S_p^α and $T_{\sigma j}^\alpha$ define for all $b \in |\alpha|'$ a unique instantaneous description ID (b) such that ID $(f_L(b))$ is a left successor of ID (b), and ID $(f_R(b))$ is a right successor of ID (b).
Finally, the validity of $\forall y\, F_A(y)$ guarantees that the computation tree is accepting.

It is easy to check that $F(w)$ contains only $O(n)$ atomic formulas, each of length $O(\log n)$. Therefore $|F(w)| = O(n \log n)$. It is also obvious that the formula $F'(w)$ and its functional form $F(w)$ can be computed from w in logarithmic space by a Turing machine. Note that most parts of $F(w)$ depend only on $n = |w|$. □

COROLLARY 1. *There is a $c > 1$ such that no deterministic Turing machine accepts the satisfiable formulas of the monadic $\forall\exists\exists$ class in time $O(c^{n/\log n})$.*

Proof. By standard diagonalization arguments, there is a language L in $DTIME\,(c_2^n)$ which is not in $DTIME\,(c_1^n)$ for $c_1 < c_2$ [19].

L is then in $ASPACE\,(n)$. Assume Corollary 1 is not true. Then by first transforming L according to the lower bound theorem to the monadic $\forall\exists\exists$ class, and then accepting this language fast, L could be accepted in deterministic time c_1^n. □

COROLLARY 2. *For every nondeterministic Turing machine M which accepts the satisfiable formulas of the monadic $\forall\exists$ class, there exists a constant c, such that M uses space $cn/\log n$ for infinitely many inputs.*

Proof. We use the hierarchy result for $NSPACE$ [35] and the fact that an alternating Turing machine with only one successor configuration for each universal configuration, is a nondeterministic Turing machine. □

CONCLUSIONS

Alternating Turing machines are a powerful tool in the few areas where applications have been found so far. They can make connections visible, which are not seen otherwise. It seems impossible to find the lower bound for the Ackermann case of the decision problem, without knowing alternating Turing machines. Even knowing the result, a direct description of the computation of a deterministic exponential time bounded Turing machine M by a $\exists^* \forall\exists^*$ formula, without obviously copying the simulation of M by an alternating Turing machine, seems impossible.

We are used to think that nondeterministic machines correspond to existential quantifiers (e.g. satisfiability in propositional calculus), and that alternating machines correspond to a sequence of alternating quantifiers (e.g. quantified boolean formulas, i.e. the theory of $\{0, 1\}$ with equality). This paper shows that this needs not always to be the case.

Examples

1. Not only the satisfiability problem of the $\exists^*$ class, but also of the $\forall^*$ class is *NP*-complete (not co-*NP*-complete).
2. Adding an existential quantifier to the $\forall$ prefix class, means moving from a time to a space complexity class.