Decomposition Complexity

. . . , Alexander Shen, . . .

Abstract

We consider a problem of decomposition of a ternary function into a composition of binary ones from the viewpoint of communication complexity and algorithmic information theory as well as some applications to cellular automata.

1 Introduction

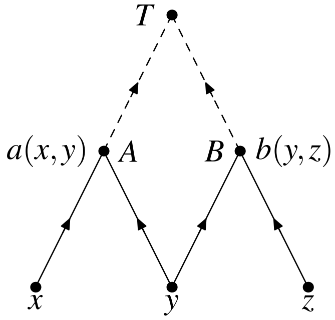
The 13th Hilbert problem asks whether all functions can be represented as compositions of binary functions. This question can be understood in different ways. Kolmogorov and Arnold (see [3]) gave positive answer for continuous functions proving that any continuous function of real arguments can be represented as a composition of continuous unary functions and addition (a binary function). On the other hand, for differentiable functions negative answer was obtained by Vituschkin. Later Kolmogorov interpreted this result in terms of information theory (see [2]): the decomposition is impossible since we have "much more" ternary functions than compositions of binary ones. (Below we present some discrete version of this argument.)

Let us start with a simple decomposition problem. An input (say, a binary string) is divided into three parts x, y and z. We want to represent $T(x, y, z)$ as a composition of three binary functions:

$$T(x, y, z) = t(a(x, y), b(y, z)).$$

In other words, we want to compute $T(x, y, z)$ under the following restrictions:

$T(x,y,z) = t(a(x,y), b(y,z))$



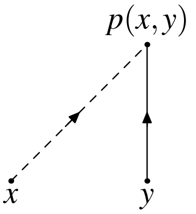
node A gets x and y and computes some function $a(x, y)$; node B gets y and z and computes some function $b(y, z)$; finally, the output node T gets $a(x, y)$ and $b(y, z)$ and should compute $T(x, y, z)$.

The two upper channels have limited capacity; the question is how much capacity is needed to make such a decomposition possible. If a- and b-channels are wide enough, we may transmit all the available information, i.e., let $a(x, y) = \langle x, y\rangle$ and $b(y, z) = \langle y, z\rangle$. Even better, we can split y in an arbitrary proportion and send one part with x and the other one with z.

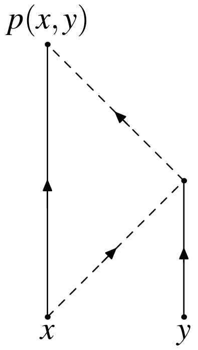
Is it possible to use less capacity? The answer evidently depends on the function T. If, say, $T(x, y, z)$ is xor of all bits in x, y and z, one bit for a- and b-values is enough. However, for other functions T it is not the case, as we see below.

In the sequel we prove different lower bounds for the necessary capacity of two upper channels in different settings; then we consider related questions in the framework of multisource algorithmic information theory [5]).

Before going into details, let us note that the definition of communication complexity can be reformulated in similar terms: one-round complexity corresponds to the network



(dotted line indicates channels of limited capacity) while two-rounds complexity corresponds to the network



etc. Another related setting that appears in communication complexity theory: three inputs x, y, z are distributed between three participants; one knows x and y, the other knows y and z, the third one knows x and z; all three participants send their messages to the fourth one who should compute $T(x, y, z)$ based on their messages (see [4]).

One can naturally define communication complexity for other networks (we select some channels and count the bits that go through these channels).

2 Communication complexity

Let $T = T(x, y, z)$ be a function defined on $\mathbb{B}^p \times \mathbb{B}^q \times \mathbb{B}^r$ (here $\mathbb{B}^k$ is the set of k-bit strings) whose values belong to some set M. We say that *decomposition complexity* of T does not exceed n if there exist $u+v \leqslant n$ and functions $a\colon \mathbb{B}^p \times \mathbb{B}^q \to \mathbb{B}^u$, $b\colon \mathbb{B}^q \times \mathbb{B}^r \to \mathbb{B}^v$ and $t\colon \mathbb{B}^u \times \mathbb{B}^v \to M$ such that

$$T(x, y, z) = t(a(x, y), b(y, z))$$

for all $x \in \mathbb{B}^p$, $y \in \mathbb{B}^q$, $z \in \mathbb{B}^r$. (As in communication complexity, we take into account the total number of bits transmitted via both restricted links. More detailed analysis could consider u and v separately.)

2.1 General upper and lower bounds

Since the logarithm of the image cardinality is an evident lower bound for decomposition complexity, it is natural to consider *predicates* T (so this lower bound is trivial).

Theorem 1 (Upper bound) *Complexity of any function does not exceed $n = p+q+r$; complexity of any predicate does not exceed $2^p + r$ as well as $2^r + p$.*

(Lower bound) *If p and r are not too small (at least $\log n + O(1)$), then there exists a predicate with decomposition complexity $n - O(1)$.*

The second statement shows that the upper bounds provided by the first one are rather tight.

Proof. (Upper bounds) For the first bound one can let, say, $a(x, y) = \langle x, y\rangle$ and $b(y, z) = z$. (One can also split y between a and b in an arbitrary proportion.)

For the second bound: for each x, y the predicate $t_{x,y}$

$$z \mapsto t_{x,y}(z) = t(x, y, z)$$

can be encoded by 2^r bits, so we let $a(x, y) = t_{x,y}$ and $b(z) = z$ and get decomposition complexity at most $2^r + p$. The bound $2^p + r$ is obtained in a symmetric way.

(Lower bound) We can use a standard counting argument. Assuming that a has u-bit values and b has v-bit values, we have

$(2^u)^{2^{p+q}}$ possible a's, $(2^v)^{2^{q+r}}$ possible b's and 2^{u+v} possible t's, i.e.,

$$2^{u2^{p+q}} \cdot 2^{v2^{q+r}} \cdot 2^{2^{u+v}}$$

possibilities (for fixed u, v). In total we get at most

$$m2^{u2^{p+q}} \cdot 2^{v2^{q+r}} \cdot 2^{2^m}$$

predicates of decomposition complexity less than m (the factor m appears since there are at most m decompositions of $m-1$ into $u+v$). Therefore, if all 2^{2^n} predicates $\mathbb{B}^p \times \mathbb{B}^q \times \mathbb{B}^r \to \mathbb{B}$ have decomposition complexity less than m, we have

$$m2^{u2^{p+q}} \cdot 2^{v2^{q+r}} \cdot 2^{2^m} \geqslant 2^{2^n}$$

or

$$\log m + u2^{p+q} + v2^{q+r} + 2^m \geqslant 2^n$$

At least one of the terms in the left-hand side should be $\Omega(2^n)$, therefore either $m \geqslant n - O(1)$, or $\log u \geqslant r - O(1)$, or $\log v \geqslant p - O(1)$. □

2.2 Bounds for explicit functions

As with circuit complexity, an interesting question is to provide a lower bound for an explicit function; it often is much harder than proving the existence results. The following statement provides a lower bound for a simple function.

Consider the predicate $T\colon \mathbb{B}^k \times \mathbb{B}^{2^{2k}} \times \mathbb{B}^k \to \mathbb{B}$ defined as follows:

$$T(x, y, z) = y(x, z)$$

where $y \in \mathbb{B}^{2^{2k}}$ is treated as a function $\mathbb{B}^k \times \mathbb{B}^k \to \mathbb{B}$.

Theorem 2 *The decomposition complexity of T is at least 2^k.*

(Note that this bound is weaker than in Theorem 1: it is close to the square root of the input size, not the input size as in the non-constructive bound.)

Proof. Assume that some decomposition of T is given:

$$T(x, y, z) = t(a(x, y), b(y, z))$$

where $a(x, y)$ and $b(y, z)$ have u and v bits respectively. Then every $y : \mathbb{B}^k \times \mathbb{B}^k \to \mathbb{B}$ determines two functions $a_y\colon \mathbb{B}^k \to \mathbb{B}^u$ and $b_y\colon \mathbb{B}^k \to \mathbb{B}^v$ obtained from a and b by fixing y. Knowing these two functions one should be able to reconstruct $T(x, y, z)$ for all x and z, since

$$T(x, y, z) = t(a_y(x), a_y(z)),$$

i.e., to reconstruct u. Therefore, the number of possible pairs $\langle a_y, b_y \rangle$, which is at most

$$2^{u2^k} \cdot 2^{v2^k},$$

is at least the number of all y's, i.e. $2^{2^{2k}}$. So we get

$$(u + v)2^k \geqslant 2^{2k},$$

or $u + v \geqslant 2^k$, therefore the decomposition complexity of T is at least 2^k. □

Remarks.

1. In this way we get a lower bound $\Omega(\sqrt{n})$ (where n is the input size) for the case when x and z are of size $O(\log n)$. It is easy to see that we can add dummy bits to x and z and get the same bound $\Omega(\sqrt{n})$ for the case when x and z have bigger size. (On the contrary, we cannot significantly decrease the size of x and z according to Theorem 1.)

2. Note that if the predicate $t(x, y, z)$ is defined as $x = z$, we need to transmit x and z completely (pigeon-hole principle). So there is a trivial (and tight) linear bound if we let x and z be long (of $\Theta(n)$) size. It would be interesting to get a linear bound for an explicit function in the more interesting case where x and z are short compared to y (preferable even of logarithmic size as above)

Note that the function T defined above does not work (since there is an upper bound:

$a(x, y)$ can be xth row in matrix y). The function $T'\colon \mathbb{B}^k \times \mathbb{B}^{2^k} \times \mathbb{B}^k \times \mathbb{B}$ defined by $T'(x, y, z) = y(x \oplus z)$ also has a (non-trivial) sublinear upper bound, see [4]. (This upper bound is still much bigger than $\Omega(\sqrt{n})$ lower bound obtained by reduction to T.)

3. For communication complexity people consider also probabilistic and randomized versions. For decomposition complexity one may do the same: either consider random variables instead of binary function (with shared random bits or independent random bits) or look for a decomposition that is Hamming-close to a given function. The following natural question arise:

- what are the relations between these models?
- what upper and lower bounds can be proven for generic functions?
- what upper and lower bounds can be proven for explicit functions (in particular, for functions T and T' defined above)?

3 Applications to cellular automata

An (one-dimensional) cellular automata is a linear array of cells. Each of the cells can be in some state from a finite set S of states (the same for all cells). At each step all the cells update their state; new state is some fixed function of its old state and the states of its neighbors. All the updates are made synchronously.

Using a cellular automaton to compute a predicate, we assume that there are two special states 0 and 1 and a neutral state that is stable (the cell remains neutral if it is neutral together with its two neighbors). To compute $P(x)$ for a n-bit string x, we assemble n cells and put them into states that correspond to x; the rest of the (biinfinite) cell array is in a neutral state.

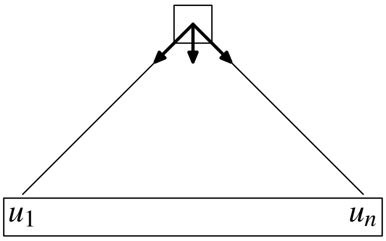
Then we start the computation; the answer should appear in some predefined cell (see below about the choice of this cell).

There is a natural non-uniform version of cellular automata: we assume that in each vertex of the time-space diagram an arbitrary function is used. Then the only restriction is caused by the limited capacity of links: we require that inputs/outputs of all functions (in all vertices) belong to some fixed set S.

In this non-uniform setting a predicate P on binary strings is considered as a family of Boolean functions P_n (where P_n is a restriction of P onto n-bit strings) and for each P_n we measure the minimal size of a set S needed to compute P_n in a non-uniform way described above. If this size is an unbounded function of n, we conclude that predicate P is not computable by a cellular automaton. (In compexity theory we use the same approach when we try to prove that some predicate is not in P since it needs superpolinomial circuits in a non-uniform setting.)

As usual, getting lower bounds for nonuniform model is difficult, but it turns out that the decomposition complexity can be used if the cellular automaton is obliged to produce the answer as soon as possible.

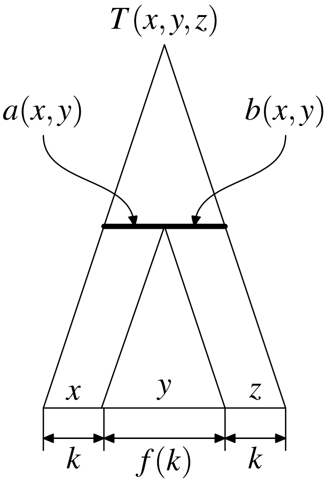
Let us fix the neighborhood and assume that each cell depends on itself and its two neighbors. Then the first occasion to use all n input bits happens around time $n/2$ in the middle of the string:



Now we assume that the output of a cellular automaton should be produced at this place (both in uniform and non-uniform model). (This is a very strong version of real-time computation by cellular automata; we could call it “as soon as possible”-computation .)

Theorem 3 *Let $T_k\colon \mathbb{B}^k \times \mathbb{B}^{f(k)} \times \mathbb{B}^k \to \mathbb{B}$ be a family of predicates that is non-uniformly computable in this sense. Then the decomposition complexity of T_k is $O(k)$, and the constant in O-notation is the logarithm of the number of states.*

Proof: see the picture



(we use bigger units for time direction to make the picture more clear).

We consider the contents of the line of length $2k$ located k steps before the end of the computation. The left half is $a(x,y)$, the right half is $b(y,z)$ and the function t is computed by the upper part of the circuit. It is easy to see that $a(x,y)$ indeed depends only on x and y since information about z has not arrived yet; for the same reason $b(y,z)$ depends only on y and z. □

Corollary: the predicate T from Theorem 2 cannot be computed in this model.

Note that this predicate is computable by a cellular automaton in linear time (we combine the string x and z into a $2k$-binary string; then we move this string across the middle part of input subtracting one at each step and waiting until our counter decreases to zero; then we know where the output bit should be read. So we get the following result:

Theorem 4 *There exists a linear-time computable predicate that is not computable “as soon as possible” even in a non-uniform model.*

Remark. This result and the intuition behind the proof are not new (see the papers of V. Terrier [6]; see also [1]). However, the explicit use of decomposition compleixty helps to formalize the intuition behind the proof. It also allows us to show (in a similar way) that this predicate cannot be computed not only “as soon as possible”, but even after $o(\sqrt{n})$ steps after this moment (which seems to be an improvement).

Questions: There could be other ways to get lower bounds for non-uniform automata (=triangle circuits). Of course, there is a counting lower bound, but this does not give any explicit function. Are there some other tools?

4 Algorithmic Information Theory

Now we can consider the Kolmogorov complexity version of the same decomposition problem. Assume that we have four binary strings x, y, z, t such that $\mathrm{K}(t|x,y,z) \approx 0$. Here $\mathrm{K}(\alpha|\beta)$ stands for conditional complexity of α when β is known, i.e., for the minimal length of a program that transforms β to α. (Hence our requirement says that there is a short program that produces t given x, y, z.)

We are looking for strings a and b such that $\mathrm{K}(a|x,y) \approx 0$, $\mathrm{K}(b|y,z) \approx 0$, and $\mathrm{K}(t|a,b) \approx 0$. Such a and b always exist, since we may let $a = \langle x,y \rangle$ and $b = \langle y,z \rangle$ (again, y can be split between a and b). However, the situation change if we restrict the complexity of a and b. As we shall see, sometimes we need a and b of total complexity close to $\mathrm{K}(x) + \mathrm{K}(y) + \mathrm{K}(z)$ even if t has much smaller compmlexity. (Note the now we cannot restrict to one-bit strings t for evident reasons.)

Theorem that shows that in general it is not possible

TO BE WRITTEN:

Proof by a game-theoretic argument: opponent has less moves so we can still create a problem for him after every move

A stronger result requires that t is obtained from x, y, z by a simple *total* function. To prove this we need a probabilitic argument: random function is covered by a small family of binary functions with negligible probability

Reformulation: is there a function that has large decomposition complexity even if we allow multi-valued functions?

Question: it would be nice to get bounds for an explicit function $t(x, y, z)$.

Question: is the reformulation in terms of classical information theory possible? Is it related to the probabilistic decomposition complexity?

References

[1] C. Choffrut and K. Culik II, On Real-Time Cellular Automata and Trellis Automata, *Acta Informatica*, **21**, 393–407 (1984).

[2] Колмогоров А.Н., Тихомиров В.М., ε-энтропия и ε-ёмкость множеств в функциональных пространствах. *Успехи математических наук*, **14** (2), p. 3–86.

[3] Колмогоров А. Н., О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения. *Доклады Академии наук СССР*, **114**(5), 953–956 (1957)

[4] Eyal Kushilevitz, Noam Nisan, *Communication complexity*, Cambridge University Press, 1997.

[5] Shen A., Multisource information theory, *Theory and Applications of Models of Computation*, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 3959 (2006), p. 327-338.

[6] Véronique Terrier, Language not recognizable in real time by one-way cellular automata. *Theoretical Computer Science*, **156**(1–2), 281–287 (1996).