Communication and Computation

Ulrich Tamm

I. Introduction

In computer science, often computational problems can be transformed into communication problems. The most illustrative example is probably the evaluation of a function via a Boolean formula. The correspondence to an equivalent communication game is as follows. A Boolean formula computing the function value $g(x_1,\ldots,x_k)$ can be represented by a binary tree. The leaves are labeled with variables $x_i \in \{0,1\}$ or their negations (the same variable may occur on several leaves). In each inner vertex the two incoming subtrees are combined either via an "and" ($\wedge$) gate or an "or" ($\vee$) gate, i. e., in these inner nodes the results g_1 and g_2 obtained so far, are processed according to $g_1 \wedge g_2$ or $g_1 \vee g_2$. The result $g(x_1,\ldots,x_k)$ will finally be found at the root of the tree. So a computation is carried out as a process starting at the leaves and ending at the root. The depth (minimal length of a path from a leaf to a root) of an optimal Boolean formula for the calculation of a function g is an important parameter in theoretical computer science. The equivalent communication game is as follows. Alice holds an input $(x_1,\ldots,x_k)$ with $g(x_1,\ldots,x_k)=1$ and Bob holds an input $(y_1,\ldots,y_k)$ with $g(y_1,\ldots,y_k)=0$. They exchange bits of information until they find a component $i \in \{1,\ldots,k\}$ in which $x_i = y_i$. The minimal number of communication bits exchanged by Alice and Bob maximized over all inputs is the communication complexity of this game. The communication protocol can be represented by a binary tree. In this tree the vertices are labeled with the person whose turn it is to send, further an edge to the left successor of a vertex corresponds to a bit 1 and an edge to the right successor correspond to a bit 0. A one–to–one correspondence to a Boolean formula now is obtained by assigning the $\vee$– gates to Alice and the $\wedge$–gates to Bob (so it is the respective person's turn to send in these vertices). Hence, we can use the same tree as for the Boolean formula with the difference that communication starts in the root and terminates in the leaves. For recent results on this model we refer to [4].

A similar communication model, based on a decision problem rather than a search problem, is introduced in the next section. Methods from data compression are used to derive lower bounds. Finally, the application to a computation problem will be discussed.

II. Communication Complexity

The communication complexity $C(f)$ of a function f is the number of bits that two persons have to exchange in order to evaluate $f(x,y)$, when initially Alice only knows $x \in \mathcal{X}$ and Bob only knows $y \in \mathcal{Y}$. Communication complexity, as introduced by Yao [15] for functions $f : \mathcal{X}\times\mathcal{Y} \longrightarrow \mathcal{Z}$ with finite $\mathcal{X},\mathcal{Y},\mathcal{Z}$, turned out to be an important topic in computer science, especially in the derivation of bounds for area-time tradeoff in VLSI-layout and for depth of circuits, decision trees and branching programs. We refer to the books by Kushilevitz and Nisan [11] or Hromkovic [6] for further details. Connections between communication complexity and information theory are discussed by Karp [7] and Körner and Orlitsky [10].

Communication proceeds according to a protocol, on which the communicators agree in advance. Alice and Bob exchange bits of information over a noiseless channel until they both know the result $f(x,y)$. The set of messages a person is allowed to send is required to be prefix-free in order to assure that the second person immediately recognizes the end of the message.

An upper bound on the communication complexity is always obtained via the following naive protocol. Alice sends all the bits of her input x to Bob using at most $\lceil\log|\mathcal{X}|\rceil$ bits. Bob then knows x and y and hence is able to calculate $f(x,y)$, which he returns to Alice using at most $\lceil\log|\mathcal{Z}|\rceil$ bits. The logarithm throughout this paper is always taken to the base 2. Hence (w. l. o. g. $|\mathcal{X}| \le |\mathcal{Y}|$)

$$C(f) \le \lceil\log|\mathcal{X}|\rceil + \lceil\log|\mathcal{Z}|\rceil \tag{1}$$

Lower bounds for $C(f)$ are obtained from the matrices $M_z(f) = (a_{xy})_{x,y}$ defined by $a_{xy} = \begin{cases} 1 & \text{if } f(x,y)=z \\ 0 & \text{if } f(x,y)\neq z \end{cases}$. In the sequel we need the following rank lower bound

$$C(f) \ge \lceil\log\sum_z \mathrm{rank} M_z(f)\rceil. \tag{2}$$

III. Vector–valued Functions and Data Compression

We are going to study the communication complexity of *vector-valued functions* f^n, which are defined on the direct sums $\mathcal{X}^n,\mathcal{Y}^n$ of the sets from the domain of some basic function $f : \mathcal{X}\times\mathcal{Y}\to\mathcal{Z}$. Elements of $\mathcal{X}^n$ and $\mathcal{Y}^n$ are denoted as x^n and y^n, respectively. Hence, e. g., $x^n = (x_1,\ldots,x_n)$ for some $x_1,\ldots,x_n \in \mathcal{X}$. With this notation

$$f^n(x^n,y^n) = \big(f(x_1,y_1),\ldots,f(x_n,y_n)\big)$$

For instance, let $si : \{0,1\}\times\{0,1\} \to \{0,1\}$ be the logical "and". If we interpret the vectors $x^n,y^n \in \{0,1\}^n$ as representations of two subsets of an n–elementary set ($x_i = 1$ exactly if the i–th element is contained in the subset represented by $x^n = (x_1,\ldots,x_n)$), then the vector–valued function $si^n(x^n,y^n)$ gives the intersection of these two sets.

U. Tamm is with the Department of Computer Science, University of Chemnitz, 09107 Chemnitz, Germany. E-mail: tamm@informatik.tu-chemnitz.de

THEOREM 1:

$$C(si^n) = \lceil \log 3 \rceil \tag{3}$$

Proof: For set-intersection the rank bound (2) combined with the techniques from [1], [13], [14] yields

$$C(si^n) \geq \lceil n \cdot \log(\text{rank}M_0(si) + \text{rank}M_1(si)) \rceil$$

$$= \lceil n \cdot \log(\text{rank} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} + \text{rank} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}) \rceil = \lceil n \cdot \log 3 \rceil$$

In order to obtain the same upper bound, we shall modify the naive protocol, which would require $2n$ bits of transmission. Again, in the first round Alice encodes her input $x^n \in \{0,1\}^n$, now using a code ϕ, and sends $\phi(x^n)$ to Bob. Bob then knows both values and hence is able to compute the result $si^n(x^n, y^n)$, which is returned to Alice. However, in knowledge of x^n the set of possible function values is reduced to the set $S(x^n) = \{y^n : y^n \subset x^n\}$. Hence, only $\lceil \log S(x^n) \rceil$ bits have to be reserved for the transmission of $si^n(x^n, y^n)$ such that Alice can assign longer messages to elements with few subsets. So, in contrast to the trivial protocol, the messages $\phi(x^n) : x^n \in \{0,1\}^n$ are now of variable length. Since the set $\{\phi(x^n) : x^n \in \{0,1\}^n\}$ must be a prefix code, Kraft's inequality yields a condition, from which the upper bound can be derived. Specifically, we require that to each x^n corresponds a message $\phi(x^n)$ of (variable) length $l(x^n)$ such that for all $x^n \in \{0,1\}^n$ the sum $l(x^n) + \lceil \log S(x^n) \rceil$ takes a fixed value, L say. Kraft's inequality states that a prefix code exists, if $\sum_{x^n} 2^{-l(x^n)} \leq 1$. This is equivalent to $\sum_{x^n} 2^{\lceil \log S(x^n) \rceil} \leq 2^L$. It can be shown that with the choice $L = \lceil \log 3^n \rceil$ Kraft's inequality holds.

IV. AMORTIZED COMMUNICATION COMPLEXITY

Direct sum methods in communication complexity as those used for the proof of Theorem 1 are useful tools in separating complexity classes [14]. Further applications are the comparison of lower bound techniques and the study of their power (how large can be the gap between the lower bound and the communication complexity). The intuition is that small gaps for the basic function f become large for the vector-valued function f^n.

Karchmer, Raz, and Wigderson [9] asked how much better simultaneous computations are compared to sequential (componentwise) evaluation of the function f^n for basic Boolean functions $f : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$. They conjectured that the *amortized communication complexity*

$$\overline{C}(f) = \frac{1}{n} \limsup_{n \to \infty} C(f^n)$$

is close to $C(f)$ – the communication complexity of the basic function f. This *direct sum conjecture* was further studied in [5], [8], and [12] and also randomized and nondeterministic protocols were considered.

As Karchmer, Raz, and Wigderson [9] point out, a proof of their direct sum conjecture would be a decisive step towards a separation of the complexity classes NC^1 and NC^2 - a long outstanding open problem in computer science.

Recently, in [3] the notion of closeness in the above conjecture was defined more formally. Namely the direct sum conjecture in [3] was stated as

$$C(f^n) = n \cdot (C(f) - O(1))$$

Based on the results of the previous section, we want to discuss the application of information theoretic methods in order to analyze the direct sum conjecture. Observe that the amortized communication complexity is just the limit for $n \longrightarrow \infty$ of the communication complexity of the vector–valued function divided by the number of components n. Hence, with Theorem 1 the function si^n can be evaluated much faster considering all n components simultaneously than by componentwise communication of the results for the basic function si, which would cost $2n$ bits. So the amortized communication complexity of the function si is $\frac{1}{n} \lim_{n \to \infty} C(si^n) = \log 3$. Of course, the difference $C(si) - \overline{C}(si) = 2 - \log 3$ is too small in order to disprove the direct sum conjecture. However, the data compression techniques used in the proof of Theorem 1 might be applied to look for candidates f with a larger gap between communication complexity $C(f)$ and amortized complexity $\overline{C}(f)$.

REFERENCES

[1] R. Ahlswede and N. Cai, "On communication complexity of vector-valued functions", *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 2062–2067, 1994.

[2] R. Ahlswede, N. Cai, and U. Tamm, "Communication complexity in lattices", *Appl. Math. Letters* vol. 6, no. 6, pp. 53–58, 1993.

[3] A. Ambainis, H. Buhrmann, W. Gasarch, B. Kalyanasundaram, and L. Torenvliet, "The communication complexity of enumeration, elimination, and selection", *J. Comput. System Sci.* vol. 63, 148–185, 2001.

[4] J. Edmonds, R. Impagliazzo, S. Rudich, and J. Sgall, "Communication complexity towards lower bounds on circuit depth", *Computational Complexity*, vol. 10, pp. 210–246, 2001.

[5] T. Feder, E. Kushilevitz, M. Naor, and N. Nisan, "Amortized communication complexity", *SIAM J. Comp.*, vol. 24, no. 4, pp. 736–750, 1995.

[6] J. Hromkovic, *Communication Complexity and Parallel Computing*, Springer, 1997.

[7] R. M. Karp, "ISIT'98 Plenary Lecture Report: Variations on the theme of 'Twenty Questions' " , *IEEE Information Theory Society Newsletter*, vol. 49, no. 1, pp. 1–5 and 21–22, March 1999.

[8] M. Karchmer, E. Kushilevitz, and N. Nisan, "Fractional covers and communication complexity", *SIAM J. Disc. Math.*, vol. 8, no. 1, pp. 76–92, 1995.

[9] M. Karchmer, R. Raz, and A. Wigderson, "Super-logarithmic depth lower bounds via direct sum methods in communication complexity", *Comput. Complexity*, vol. 5, pp. 191–204, 1995.

[10] J. Körner and A. Orlitsky, "Zero-error information theory", *IEEE Trans. Inform. Theory*, Commemorative Issue, vol. 44, no. 6, pp. 2207–2229, 1998.

[11] E. Kushilevitz and N. Nisan, *Communication Complexity*, Cambridge University Press, 1997.

[12] M. Naor, A. Orlitsky, and P. Shor, "Three results on interactive communication", *IEEE Trans. Inform. Theory*, vol. 39, no. 5, pp. 1608–1615, 1993.

[13] U. Tamm, Communication complexity of sum–type functions invariant under translation, *Information and Computation*, vol. 116, pp. 162–173, 1995.

[14] U. Tamm, Communication complexity of functions on direct sums, *Numbers, Information and Complexity (Festschrift in Honour of Rudolf Ahlswede)*, I. Althöfer, N. Cai, G. Dueck, L. Khachatrian, M. Pinsker, A. Sárközy, I. Wegener, and Z. Zhang (eds.), pp. 589–602, Kluwer Academic Publishers, 2000.

[15] A. C. Yao, "Some complexity questions related to distributive computing", *Proceedings ACM Symposium on the Theory of Computing*, pp. 209–213, 1979.