ON THE IMPROBABILITY OF REACHING BYZANTINE AGREEMENTS

(Preliminary Version)

Ronald L. Graham
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, New Jersey 07974

Andrew C. Yao*
Department of Computer Science
Princeton University
Princeton, New Jersey 08544

Abstract. It is well known that for the Byzantine Generals Problem, no deterministic protocol can exist for an n-processor system if the number t of faulty processors is allowed to be as large as $n/3$. In this paper we investigate the maximum achievable agreement probability $\beta_{n,t}$ in a model in which the faulty processors can be as devious and powerful as possible. We also discuss a restricted model with $\beta'_{n,t}$ denoting the corresponding maximum achievable probability. We will prove that: (i) for $n=3$, $t=1$ (the first nontrivial case), $\beta_{3,1}=(\sqrt{5}-1)/2$ (the reciprocal of the golden ratio); (ii) for all ϵ with $0<\epsilon<1$, if $\frac{t}{n} > 1 - \frac{\log(1-\epsilon)^{1/2}}{\log(1-(1-\epsilon)^{1/2})}$ then $\beta'_{n,t} < \epsilon$.

1. Introduction

The design of protocols for reaching agreements in the presence of faulty processors is an important issue in distributed computing. One classic problem in this area is the *Byzantine Generals Problem* ([PSL] [LSP]), where a system of n processors, in which as many as t of them may be faulty, wish to agree on a binary value v held by a certain distinguished processor, called the *commander*. A *protocol* is an algorithm which specifies the rules governing the behavior of the nonfaulty processors. In accordance with these rules the nonfaulty processors send and receive messages from each other in synchronized rounds before finally deciding on their values; the faulty processors may behave in an arbitrary manner. At the end of an execution of the protocol, we say that a *Byzantine agreement* has been achieved when: (1) all the nonfaulty processors agree on the same value, and (2) if the commander is nonfaulty, then all the nonfaulty processors in fact agree on the value v held by the commander.

A fundamental issue is to understand under what circumstances protocols exist for reaching Byzantine agreement, and when they do, whether efficient ones exist.

This problem has been studied extensively in the literature. Pease, Shostak, and Lamport [PSL] (see also [LSP]) showed that there exists a protocol which guarantees agreement if and only if $t<n/3$. Dolev and Strong [DS] then showed that when $t<n/3$, there are protocols which use at most a polynomial number (in n) of messages. Another efficient protocol for this problem was given in Dolev, Fischer, Fowler, Lynch, and Strong [DFFLS]). Concerning the number of *rounds* needed, Fischer and Lynch [FL] showed that no protocol exists that always terminates in fewer than $t+1$ rounds, which is best possible since the protocol given in [PSL] achieves this bound. There are many other interesting variants of the Byzantine Generals Problem. For example, the protocol may work asynchronously (see [ABDKPR], [Be], [Bra1] [FLP]). There are also numerous types of problems involving consensus seeking, and we refer the interested reader to the literature (e.g., see [BLS], [DLS], [KKL], and their references).

When guaranteed agreement is not possible, what is the best one can achieve? One direction is to relax the requirements for agreement, e.g., to allow *approximate agreement* (Dolev, Lynch, Pinter, Stark, and Weihl [DLPSW]), *inexact agreement* (Mahaney and Schneider [MS]), or *persuasive agreement* (Ben-Or [Be], Chor, Merritt and Shmoys [CMS], Dwork, Shmoys and Stockmeyer [DSS]). Randomized

* This research was supported in part by the National Science Foundation under Grant CCR-8813283.

algorithms have also been considered in Rabin [R] for use in the Byzantine Generals Problem, (also, see Ben-Or [Be], Bracha [Br2], Chor and Coan [CC]).

In this paper, we shall investigate the following basic problem. Retaining the original notion of agreement, what is the best *probability* of agreement one can achieve using randomized algorithms?

This problem was considered in Karlin and Yao [KY]. There it was shown that for a certain model (called the *simultaneous model*, described below), no protocol can achieve a probability of agreement greater than 2/3 when $t \geq n/3$. On the other hand, in this model the bound of 2/3 can be achieved when $n=3$, $t=1$.

In the present paper, we will study a different but related model, which we call the *sequential model*, described in Section 2. Let us denote by $\beta_{n,t}$ the least upper bound on the probability of agreement achievable in this model by any probabilistic protocol. Our first result is the following.

Theorem 1. $\beta_{3,1} = (\sqrt{5}-1)/2$.

Since $(\sqrt{5}-1)/2 < 2/3$, Theorem 1 shows that in the sequential model, in which faulty processors can wait to see the transmitted messages of nonfaulty processors in the current round before having to decide on their own messages, strictly increases the power of the adversary over that of the simultaneous model of [KY].

In Section 5 we discuss the situation when n is large. It turns out, perhaps not surprisingly, that when t can be large, say on the order of n itself, then the corresponding probability of success for any protocol must tend to zero, as n becomes large.

2. The Model

We begin by formalizing the notion of a probabilistic protocol for reaching Byzantine agreement. We employ a modified version of the formalism given by Broder and Dolev [BD].

Processors will be denoted by $G_0, G_1, \ldots, G_{n-1}$, where without loss of generality we let G_0 be the commander and $v \in \{0,1\}$ its initial value. Each processor G_i is assumed to have access to a family of probability distributions $f_{r,i}$ on the set $\{1,2,3,\ldots\}$, $r=1,2,3,\ldots$ These serve as G_i's source of randomness: at round r, G_i chooses a "random" integer $N_{r,i}$ according to the distribution $f_{r,i}$ (informally, we can think of this as G_i's "coin flip"). The sequence $(N_{1,i}, N_{2,i}, \ldots, N_{r,i})$ is denoted by $\overline{N_{r,i}}$.

In a k-round agreement protocol $\mathbb{A}$, the behavior of G_i during round r, $1 \leq r \leq k$, can be described by a *transition function* $\delta_i = \delta_i(r, M_{r-1,i}, \overline{N}_{r,i})$ where $M_{r-1,i}$ is the set of messages received by G_i up through round $r-1$. The value of $\delta_i(r, M_{r-1,i}, \overline{N}_{r,i})$ is a vector $\overline{M}_{r,i} = (m_{r,i}^0, m_{r,i}^1, \ldots, m_{r,i}^{n-1})$, where $m_{r,i}^j$ is the message (any finite binary string) G_i sends to G_j during round r. Also, we define a *binary decision function* $\mu_i(M_{k,i}, \overline{N}_{k,i})$, which is the probability that the value v_i that G_i finally decides upon is 0 after the k^{th} round, when all messages have been received $(M_{k,i})$ and all random choices have been made $(\overline{N}_{k,i})$. The complete specification of a protocol $\mathbb{A}$ consists of a transition function δ_i and a decision function μ_i for each processor i, $0 \leq i < n$. Let $\mathcal{A}_n$ denote the set of all such protocols.

To continue, a *scenario* σ consists of a binary value $v = v(\sigma)$, a set $F = F(\sigma) \subseteq \{0,1,\ldots,n-1\}$ indexing the faulty processors G_j, $j \in F$, and a *generalized transition function* δ_F' which dictates the behavior of the faulty processors G_j. Mathematically, δ_F' takes as input $(r, M_{r,i}[i \notin F], \overline{N}_{r,j}[j \in F])$ and assigns outputs $\overline{m}_{r,j}$, $j \in F$, where $\overline{m}_{r,j} = (m_{r,j}^0, m_{r,j}^1, \ldots, m_{r,j}^{n-1})$ is the set of messages to be sent by G_j, $j \in F$, in round r. Note that we have incorporated the capability for the faulty processors to collude, to spy on all communication lines, and to wait for messages transmitted by nonfaulty processors in the current round to arrive before making decisions on their own messages.

Consider a k-round agreement protocol $\mathbb{A}$ and any scenario σ with $|F| \leq t$. Given n "random" sequences $\overline{N}_i$ of positive integers generated by the $f_{r,i}$, $0 \leq i < n$, an execution of protocol $\mathbb{A}$ is completely determined by interpreting the $\overline{N}_i$ as the "coin flips" used by G_i. We call such an execution a *run* of $\mathbb{A}$, and we say it is *successful* if the following two conditions hold at the end after k rounds:

1. *Consistency*: $v_i = v_{i'}$ for all $i, i' \notin F$.
2. *Validity*: If $0 \notin F$ (i.e., the commander is not faulty) then $v_i = v$ for all $i \notin F$.

Let $\eta(\mathbb{A}, \sigma)$ denote the probability that a random run of $\mathbb{A}$ using σ will be successful, and define

$$\eta_t(\mathbb{A}) := \inf\{\eta(\mathbb{A}, \sigma) \mid |F(\sigma)| \leq t\}.$$

Thus, $\eta_t(\mathbb{A})$ is the smallest probability of agreement that the faulty processors can enforce on the system. Finally, define

$$\beta_{n,t} := \sup_{\mathbb{A}} \eta_t(\mathbb{A}).$$

We will call this the *sequential model*.

By way of contrast, the (simultaneous) model considered in [KY] only allows faulty processors to use information obtained in *previous* rounds (but not the current round) in deciding what to transmit during the current round. Thus, the corresponding allowable δ_F' in this case are more restricted than in the sequential model.

In the next two sections we will give a somewhat detailed outline of the proof of Theorem 1.

3. A Lower Bound for $\beta_{3,1}$

We first show that

$$(1) \qquad \beta_{3,1} \geq (\sqrt{5}-1)/2 := \hat{\phi}.$$

Consider the following protocol $\mathbb{A}_0$:

Step 1. G_0 sends v to both G_1 and G_2;

Step 2. If G_1 receives the value a from G_0 then G_1 will decide probabilistically on the value v_1, where $v_1 = a$ with probability $\hat{\phi}$ and $v_1 = 1-a$ with probability $1-\hat{\phi}$; G_1 then sends v_1 to G_2;

Step 3. Suppose G_2 receives b from G_0 in Step 1 and a' from G_1 in Step 2. If $b = a'$ then G_2 decides on $v_2 = b$; if $b \neq a'$ then G_2 will decide probabilistically on the value $v_2 = b'$, where $b' = b$ with probability $\hat{\phi}$ and $b' = 1-b$ with probability $1-\hat{\phi}$.

Claim. $\eta_1(\mathbb{A}_0) = \hat{\phi}$.

To compute $\eta_1(\mathbb{A}_0)$, we need to consider the various possibilities, namely all processors are nonfaulty, and exactly one processor is faulty. For example, suppose the commander G_0 is faulty, and in Step 1 sends the value 0 to G_1 and the value 1 to G_2. For the protocol to fail to reach agreement, we must have G_1 choosing $v_1 = 0$ in Step 2, and G_2 choosing $b' = 1$ in Step 3; this occurs with probability $\hat{\phi} \cdot \hat{\phi} = \hat{\phi}^2$. Thus, for this case, the probability of reaching agreement is $1 - \hat{\phi}^2 = \hat{\phi}$. In the same way one can prove that in each of the other cases, the probability for the protocol to reach agreement is at least $\hat{\phi}$. This shows that $\beta_{3,1} \geq \eta_1(\mathbb{A}_0) = \hat{\phi}$, as required.

4. An Upper Bound for $\beta_{3,1}$

We will require a number of definitions before being able to carry out a proof that

$$(2) \qquad \beta_{3,1} \leq \hat{\phi}.$$

Let $T = T_k$ be a complete binary tree consisting of a *root*, denoted by *root*(T) at level 1, and $k-1$ levels of other nodes. A node at level k is called a *leaf*. Any non-leaf u of T has two descendants, a *leftchild* u_0 and a *rightchild* u_1. The level of a node u is denoted by $\lambda(u)$. If $\lambda(u) = i < k$ then $\lambda(u_j) = i+1$, $j \in \{0, 1\}$. The notation $u\,u_j$ denotes the *edge* in T from u to u_j. We define $\lambda(u\,u_j)$, the level of $u\,u_j$ to be equal to $\lambda(u)$.

Definition. We call $T = T_k$ a *k-level cost tree* if T is a complete binary tree with k levels in which each edge e of T is assigned a *cost* $c(e)$, $0 \leq c(e) \leq 1$, where for each non-leaf node u, $c(u\,u_0) + c(u\,u_1) = 1$.

For a node u, let $P(u)$ denote the sequence of nodes *root*$(T) = w_1, w_2, \ldots, w_i = u$ from *root*(T) to u (thus, $\lambda(u) = i$). Also, let $P_e(u)$ denote the corresponding sequences of edges $e_1, e_2, \ldots, e_{i-1}$ from *root*(T) to u (thus, $e_j = w_j w_{j+1}$).

Definition. Let T be a k-level cost tree. A *control function* ρ for T is a mapping from the set I of internal (= non-leaf) nodes of T into $\{0, 1, *\}$. We say that a node u is *ρ-reachable*, and we write $\chi(\rho, u) = 1$, if the following holds: For all $w_j \in P(u)$, $w_j \neq u$, $\rho(w_j) = 0$ implies w_{j+1} is a leftchild of w_j, and $\rho(w_j) = 1$ implies w_{j+1} is a rightchild of w_j. Otherwise we write $\chi(\rho, u) = 0$.

We say that ρ is an *A-control-function* if $\rho(u) \in \{0, 1\}$ for all internal nodes u with $\lambda(u)$ odd. Similarly, we say that ρ is a *B-control-function* if $\rho(u) \in \{0, 1\}$ for all u with $\lambda(u)$ even. Let F_A and F_B denote the set of all A- and B-control-functions, respectively.

Definition. For a node u, define

$$\tau_A(u) := \prod_{\substack{e_j \in P_e(u) \\ \lambda(e_j)\ odd}} c(e_j)$$

$$\tau_B(u) := \prod_{\substack{e_j \in P_e(u) \\ \lambda(e_j)\ even}} c(e_j)$$

$$\tau_C(u) := \tau_A(u)\,\tau_B(u).$$

Definition. A leaf l of T is called an *A-leaf* if either $\lambda(l)$ is odd and *parent*(l) is a leftchild, or $\lambda(l)$ is even and l is a leftchild. A leaf l is called a *B-leaf* if either $\lambda(l)$ is odd and l is a rightchild, or $\lambda(l)$ is even and *parent(l)* is a rightchild. A leaf l is called a *C-leaf* if either l and *parent(l)* are both leftchildren, or l and *parent(l)* are both rightchildren.

We illustrate this definition in Fig. 1.

We denote the set of A-leaves, B-leaves and C-leaves of T by L_A, L_B and L_C, respectively.

Define:

$$c_A(T) := \min_{\rho \in F_A} \left\{ \sum_{l \in L_B} \chi(\rho, l) \tau_B(l) \right\}$$

$$c_B(T) := \min_{\rho \in F_B} \left\{ \sum_{l \in L_A} \chi(\rho, l) \tau_A(l) \right\}$$

$$c_C(T) := \sum_{l \in L_C} \tau_C(l) .$$

Finally, define the *cost* of T by

$$cost(T) := \min\{c_A(T), c_B(T), c_C(T)\}.$$

As an example, observe that for the 3-level cost tree T^* in Fig. 2, we have $cost(T^*) = \hat{\phi}$.

In fact, this cost tree is not unrelated to the protocol $\mathbb{A}_0$ described earlier which had $\eta_1(\mathbb{A}_0) = \hat{\phi}$. Our first lemma shows that T^* is extremal in this respect.

Lemma 1. *For any cost tree T,*

$$(3) \qquad cost(T) \le \hat{\phi} .$$

Proof: We proceed by induction on the number of levels of T. Assume that $T = T_3$ has 3 levels with edge costs as shown in Fig. 3 (where $\bar{x} = 1 - x$, etc.). Thus,

$$\begin{aligned} cost(T_3) &= \min\{\bar{y}, \bar{z}, x, xy + \bar{x}\bar{z}\} \\ &\le \sup_{0 \le x,y,z \le 1} \min\{\bar{y}, \bar{z}, x, xy + \bar{x}\bar{z}\} \\ &= \sup_{0 \le x,y \le 1} \min\{\bar{y}, x, xy + \bar{x}\} = \hat{\phi} \end{aligned}$$

which is achieved by taking $x = \hat{\phi} = \bar{y}$.

Now, assume that (3) holds for all k-level cost trees for some $k \ge 3$, and let T be an arbitrary $(k+1)$-level cost tree. First, observe that if T' is formed from T by setting $c(e) = 0$ for any edge e incident to a leaf $l \notin L_A \cup L_B \cup L_C$, then $cost(T) \le cost(T')$ (since such edges are not involved in the evaluation of the cost). We consider the bottom three levels of T'. There are two cases, depending on the parity of k. We illustrate the case of k even in Fig. 4(a). (The case of k odd is similar and is omitted.)

In Fig. 4(b) we show the "collapsed" tree $\hat{T}'$, where the indicated transformations are performed on all nodes u at level $k-2$. We next compute how the two fragments contribute to the costs of their respective trees (where we have to keep in mind the various options possible for the control-functions on these fragments). It follows from the definitions that these contributions are:

$$\begin{aligned} c_A(T') &= \cdots + \tau_B(u)(x\bar{r} + \bar{x}\bar{s}) + \cdots , \\ c_B(T') &= \cdots + \tau_A(u) \cdot \min(y,z) + \cdots , \\ c_C(T') &= \cdots + \tau_C(u)(x(yr + \bar{y}) + \bar{x}(zs + \bar{z})) + \cdots , \\ c_A(\hat{T}') &= \cdots + \tau_B(u)(x\bar{r} + \bar{x}\bar{s}) + \cdots , \\ c_B(\hat{T}') &= \cdots + \tau_A(u) \cdot \min(y,z) + \cdots , \\ c_C(\hat{T}') &= \cdots + \tau_C(u)(1 - x\bar{r} - \bar{x}\bar{s} + (x\bar{r} + \bar{x}\bar{s})(1 - \min(y,z))) \\ &\quad + \cdots . \end{aligned}$$

However, it is easily verified that

$$x(yr + \bar{y}) + \bar{x}(zs + \bar{z}) \le (1 - x\bar{r} - \bar{x}\bar{s} + (x\bar{r} + \bar{x}\bar{s})(1 - \min(y,z)))$$

so we have $c_A(T') = c_A(\hat{T}')$, $c_B(T') = c_B(\hat{T}')$ and $c_C(T') \le c_C(\hat{T}')$. This implies

$$cost(T) \le cost(T') \le cost(\hat{T}') \le \hat{\phi}$$

by the induction hypothesis, since $\hat{T}'$ only has k levels, and the lemma is proved. ■

Now, we proceed to the proof of (2). Let $\mathbb{A}$ be a $3t$-round agreement protocol with transition functions δ_i, $1 \le i \le 3t$. It is not hard to see that we do not lose any generality by assuming that during any particular round r, only one of the three processors actually transmits, and further, that each transmission consists of a single bit to each of the other two processors. Furthermore, we can assume that the last bits transmitted by G_1 and G_2 are, in fact, their respective decisions v_1 and v_2. Specifically, we assume that G_i transmits only during rounds $r \equiv i + 1 \pmod 3$, and that a single bit $\gamma_{r,j} \in \{0, 1\}$ is transmitted to G_j, $j \ne i$. Our overall plan will be to construct a $(2t+1)$-level cost tree $T = T(\mathbb{A})$ and scenarios σ_0, $\sigma_{1,\rho}$, $\rho \in F_A$, and $\sigma_{2,\rho'}$, $\rho' \in F_B$, so that $\eta(\mathbb{A}, \sigma_0) \le c_C(T)$, $\min_{\rho \in F_A} \eta(\mathbb{A}, \sigma_{1,\rho}) \le c_A(T)$ and $\min_{\rho' \in F_B} \eta(\mathbb{A}, \sigma_{2,\rho'}) \le c_B(T)$. By Lemma 1, this will be enough to prove (2).

Let $\xi_{\sigma,r,i,j}$, $1 \le r \le 3t$, denote the sequence of bits transmitted between G_i and G_j up through round r, and let $V^+_{\sigma,r,i}$ denote the random variable $(\xi_{\sigma,r,i,i+1}, \xi_{\sigma,r,i,i+2}, \bar{N}_{r,i})$ where index addition is performed modulo 3, and $\bar{N}_{r,i}$ represents G_i's random choices up through round r. (Thus, $\xi_{\sigma,r,i,j} = \xi_{\sigma,r,j,i}$).

Intuitively, $V^{+}_{\sigma,r,i}$ represents the *view* that G_i has seen up through round r. We will denote the truncated vector $(\xi_{\sigma,r,i,i+1}, \xi_{\sigma,r,i,i+2})$ by $V_{\sigma,r,i}$. Further, for ξ, $\xi' \in \{0,1\}^*$, define

$$P_{\sigma,r,i}(\xi,\xi') = Pr\{V_{\sigma,r,i} \in \{(\xi 0,\xi'), (\xi,\xi' 0)\} \mid V_{\sigma,r-1,i}=(\xi,\xi')\}.$$

We will be primarily interested in the case when G_i does not transmit during round r. In this case, $P_{\sigma,r,i}(\xi,\xi')$ is the probability that G_i will receive a 0 during round r, given that G_i's (truncated) view $V_{\sigma,r-1,i}$ at round $r-1$ is (ξ,ξ').

Our next goal will be to construct three special scenarios σ_A, σ_B, σ_C, and prove an important relationship connecting them. These scenarios will then be used to construct T and the final set of scenarios σ_0, $\sigma_{1,\rho}$, $\rho \in F_A$, and $\sigma_{2,\rho'}$, $\rho' \in F_B$. The scenarios σ_A, σ_B, σ_C are analogues of the three scenarios used in [PSL] to prove that no deterministic protocol exists for the case of one faulty processor among three. The basic idea is to force the two views of G_1 in σ_B and σ_C to be identical random variables, and to force the two views of G_2 in σ_A and σ_C to be identical random variables. In σ_A, G_1 is faulty and G_0 is trying to send the value 1; in σ_B, G_2 is faulty and G_0 is trying to send the value 0; and in σ_C, (a faulty) G_0 is trying to convince G_1 that 0 is the correct value, and is trying to convince G_2 that 1 is the correct value (see Fig. 5).

We now sequentially construct σ_A, σ_B, σ_C by specifying δ'_F in each case, i.e., describing how the faulty processors are to behave. Since G_0, G_1 and G_2 transmit in cyclic order (recall that G_i only transmits during rounds $r \equiv i+1 \pmod 3$), we see that for each round r, exactly one of the three scenarios has a faulty processor transmitting whose behavior must be specified (see Table 1). All other situations are already covered by the protocol $\mathbb{A}$.

Round	σ_A	σ_B	σ_C
1	–	–	G_0
2	G_1	–	–
3	–	G_2	–
4	–	–	G_0
5	G_1	–	–
6	–	G_2	–
7	–	–	G_0
⋮	⋮	⋮	⋮

When faulty processors transmit

Table 1.

Suppose for some j, $0 \le j < t$, the three scenarios are specified up through round $3j$. They are then extended through round $3j+3$ as follows:

$r = 3j+1$ $[\sigma_C]$:

(a) Suppose at this point $V_{\sigma_C,r-1,1} = (\xi,\xi')$. Then G_0 sends G_1 a 0 with probability $P_{\sigma_B,r,1}(\xi,\xi')$, (and, of course, a 1 with the complementary probability).

(b) Suppose at this point $V_{\sigma_C,r-1,2} = (\xi,\xi')$. Then G_0 sends G_2 a 0 with probability $P_{\sigma_A,r,2}(\xi,\xi')$.

$r = 3j+2$ $[\sigma_A]$:

(a) Suppose $V_{\sigma_A,r-1,2} = (\xi,\xi')$. Then G_1 sends G_2 a 0 with probability $P_{\sigma_C,r,2}(\xi,\xi')$.

(b) G_1 send G_0 a 1 (in fact this is irrelevant).

$r = 3j+3$ $[\sigma_B]$:

(a) Suppose $V_{\sigma_B,r-1,1} = (\xi,\xi')$. Then G_2 sends G_1 a 0 with probability $P_{\sigma_C,r,1}(\xi,\xi')$.

(b) G_2 sends G_0 a 1 (this is irrelevant).

Confusion Lemma. *For* $1 \le r \le 3t$,

$V^{+}_{\sigma_A,r,2}$ and $V^{+}_{\sigma_C,r,2}$ are identically distributed random variables, and

$V^{+}_{\sigma_B,r,1}$ and $V^{+}_{\sigma_C,r,1}$ are identically distributed random variables

(we denote this by writing $V^{+}_{\sigma_A,r,2} \sim V^{+}_{\sigma_C,r,2}$, etc.).

Proof: For $r=1$ this is clearly true. Suppose for some $r>1$ the lemma holds for $r-1$, i.e.,

(4) $$V^{+}_{\sigma_A,r-1,2} \sim V^{+}_{\sigma_C,r-1,2}$$

(5) $$V^{+}_{\sigma_B,r-1,1} \sim V^{+}_{\sigma_C,r-1,1}.$$

We show that (4) and (5) hold with $r-1$ replaced by r.

Case 1. $r = 3j+1$.

In σ_B, G_0 is a nonfaulty processor. Thus, if $V_{\sigma_B,r-1,1} = (\xi,\xi')$ then

(6)

$$V_{\sigma_B,r,1} = \begin{cases} (\xi,\xi' 0) & \text{with probability } P_{\sigma_B,r,1}(\xi,\xi'), \\ (\xi,\xi' 1) & \text{with probability } 1-P_{\sigma_B,r,1}(\xi,\xi'). \end{cases}$$

On the other hand, in σ_C, G_0 is faulty and our construction implies that if $V_{\sigma_C,r-1,1} = (\xi,\xi')$ then

(7)

$$V_{\sigma C,r,1} = \begin{cases} (\xi, \xi'0) & \text{with probability } P_{\sigma_B,r,1}(\xi, \xi'), \\ (\xi, \xi'1) & \text{with probability } 1-P_{\sigma_B,r,1}(\xi, \xi'). \end{cases}$$

Thus, by (5), (6) and (7), and the fact that G_1's probability of seeing a 0 as a function of $V^+_{\sigma_B,r-1,1}$ in fact depends only on its first two components, i.e., on $V_{\sigma_B,r-1,1}$, we have $V^+_{\sigma C,r,1} \sim V^+_{\sigma_B,r,1}$. Equation (4) for r can be proved in a similar way.

Case 2. $r = 3j + 2$.

Since G_1 is nonfaulty in σ_B and σ_C, and G_1 is transmitting, we have from (5), $V^+_{\sigma C,r,1} \sim V^+_{\sigma_B,r,1}$ (since $V^+_{\sigma C,r,1}$ is obtained from $V^+_{\sigma C,r-1,1}$ in the same way that $V^+_{\sigma_B,r,1}$ is obtained from $V^+_{\sigma_B,r-1,1}$).

It remains to prove (4) for r. If $V_{\sigma C,r-1,2} = (\xi, \xi')$ then since G_2 is nonfaulty in σ_C, we have

(8)

$$V_{\sigma C,r,2} = \begin{cases} (\xi, \xi'0) & \text{with probability } P_{\sigma C,r,2}(\xi, \xi'), \\ (\xi, \xi'1) & \text{with probability } 1-P_{\sigma C,r,2}(\xi, \xi'). \end{cases}$$

On the other hand, if $V_{\sigma_A,r-1,2} = (\xi, \xi')$ then by construction we have

(9)

$$V_{\sigma_A,r,2} = \begin{cases} (\xi, \xi'0) & \text{with probability } P_{\sigma C,r,2}(\xi, \xi'), \\ (\xi, \xi'1) & \text{with probability } 1-P_{\sigma C,r,2}(\xi, \xi'). \end{cases}$$

Thus, by (4), (8) and (9) we have (as in Case 1) $V^+_{\sigma C,r,2} \sim V^+_{\sigma_A,r,2}$. This proves Case 2. Case 3, $r = 3j + 3$, is similar to the previous cases and is omitted. ■

Our next step will be to construct $T = T(\mathbb{A})$, a special $(2t+1)$-level cost tree. The cost $c(e)$ assigned to each edge e will be chosen as follows. For each node u of T, let

$$\phi(u) = a_1 \dots a_d \in \{0, 1\}^d$$

denote the standard encoding of u, i.e., $a_i = 0$ if the i^{th} edge e_i in $P_e(u)$ is a leftchild, and $a_i = 1$ otherwise (where $\phi(root(T))$ is empty). Thus, $\lambda(u) = d + 1$. Let e_i denote $u\,u_i$, $i \in \{0, 1\}$, the two edges leaving u.

Case 1. $d = 2j$, $0 \le j < t$.

Define

$$c(e_0) = Pr\{\xi_{\sigma C,3j+2,1,2} = \phi(u)0 \mid \xi_{\sigma C,3j+1,1,2} = \phi(u)\},$$

(10)

$$c(e_1) = 1 - c(e_0).$$

Case 2. $d = 2j+1$, $0 \le j < t$.

Define

$$c(e_0) = Pr\{\xi_{\sigma C,3j+3,1,2} = \phi(u)0 \mid \xi_{\sigma C,3j+2,1,2} = \phi(u)\},$$

(11)

$$c(e_1) = 1 - c(e_0).$$

Cost Lemma 1. $\eta(\mathbb{A}, \sigma_C) = c_C(T)$.

Proof: It suffices to prove the following:

Claim. *Let* $\xi \in \{0, 1\}^{2j+i}$, $j \ge 0$, $i \in \{0, 1\}$, *and let* u *be the node of* T *with* $\phi(u) = \xi$. *Then*

$$Pr\{\xi_{\sigma C,3j+i+1,1,2} = \xi\} = \tau_C(u) = \prod_{e \in P_e(u)} c(e).$$

Proof of Claim: We prove the Claim by induction on $\lambda(u) = 2j+i+1$. Suppose $\lambda(u) \ge 1$ and we have proved the Claim for all smaller values of $\lambda(u)$. There are two cases, $i=0$ and $i=1$. We only treat the case $i=1$; the other case is similar and is omitted. Suppose $u = \text{leftchild}(u')$ (the case of rightchild is similar). By the induction hypothesis applied to u', we have

(12) $$Pr\{\xi_{\sigma C,3j+1,1,2} = \phi(u')\} = \tau_C(u').$$

By (10) we obtain

(13)

$$c(e_0) = Pr\{\xi_{\sigma C,3j+2,1,2} = \phi(u')0 \mid \xi_{\sigma C,3j+1,1,2} = \phi(u')\}.$$

Hence, by (12), (13) and the definition of τ_C, we have

$$Pr\{\xi_{\sigma C,3j+2,1,2} = \phi(u)\} = \tau_C(u')c(e_0) = \tau_C(u),$$

as required. As mentioned before, the arguments for the other cases are similar. Since the Claim is immediate for $u = root(T)$ then the Claim, and therefore Cost Lemma 1, are proved. ■

Our final step will be to construct the final scenarios σ_0, $\sigma_{1,\rho}$, $\rho \in F_A$, and $\sigma_{2,\rho'}$, $\rho' \in F_B$. To begin with, set $\sigma_0 = \sigma_C$. For $\sigma_{1,\rho} = (v_1, F_1, \delta'_{1,\rho})$, $\rho \in F_A$, we take $v_1 = 0$, $F_1 = \{G_1\}$, with $\delta'_{1,\rho}$ the strategy a *faulty* G_1 employs, specified by consulting the cost tree T as follows. Starting at $root(T)$, G_1 will traverse a path in T by the following process. Suppose G_1 is at a node u of T with *odd* level $\lambda(u) = 2j+1$, $j \ge 0$.

Case 1. $\rho(u) = 0$: G_1 looks at the random variable (x, y) where x and y are the final bits of

$V_{\sigma C,r,1} = (\xi_{\sigma C,r,1,2}, \xi_{\sigma C,r,1,0})$ and $r = 3j+1$. Let $s_{\alpha,\beta} := Pr\{x=\alpha, y=\beta\}$, $\alpha, \beta \in \{0,1\}$. G_1 will then generate $(0,y)$, $y \in \{0,1\}$ so that $Pr\{y=\beta\} = \frac{s_{0\beta}}{s_{00}+s_{01}}$, and send 0 to G_2 and y to G_0.

Case 2. $\rho(u) = 1$: The same as Case 1 except that G_1 generates $(1,y)$ so that $Pr\{y=\beta\} = \frac{s_{1\beta}}{s_{10}+s_{11}}$, and sends 1 to G_2 and y to G_0.

Now, G_1 takes either the *left* branch $u u_0$ (in Case 1) or the *right* branch $u u_1$ (in Case 2) and waits for G_2's next transmission γ, say at the node u' at level $2j+2$. If $\gamma = 0$ then G_1 takes the left branch $u' u_0'$; if $\gamma = 1$ then G_1 takes the right branch $u' u_1'$. G_1 is now at a node at level $2j+3$, and we repeat the process. This description specifies $\delta_{1,\rho}'$.

We remark here that it could happen that the conditional probabilities $\frac{s_{\alpha\beta}}{s_{\alpha 0}+s_{\alpha 1}}$ are not well defined when the denominators are zero. This technical difficulty can be circumvented by requiring all the various probabilities in $\mathbb{A}$ to be positive, and then letting the appropriate ones tend to zero. The continuity of the functions we are computing will then imply the desired results.

In a similar way, we construct $\sigma_{2,\rho'} = (v_2, F_2, \delta_{2,\rho'}')$, $\rho' \in F_B$. Starting at *root*(T), G_2 traverses a path in T. Whenever G_2 is at a node u with *even* level $\lambda(u) = 2j+2$, $j \geq 0$, G_2 will do the following.

Case 1. $\rho'(u) = 0$: G_2 looks at the random variable (x,y), where x and y are the final bits of $V_{\sigma C,r,2} = (\xi_{\sigma C,r,2,0}, \xi_{\sigma C,r,2,1})$ where $r = 3j+2$. Let $s_{\alpha\beta}' = Pr\{x=\alpha, y=\beta\}$, $\alpha, \beta \in \{0,1\}$. G_2 will then generate $(x,0)$, $x \in \{0,1\}$ so that $Pr\{x=\alpha\} = \frac{s_{\alpha 0}'}{s_{00}'+s_{10}'}$, and send 0 to G_1 and x to G_0.

Case 2. $\rho'(u) = 1$: The same as Case 1 except that G_2 generates $(x,1)$ so that $Pr\{x=\alpha\} = \frac{s_{\alpha 1}'}{s_{01}'+s_{11}'}$, and sends 1 to G_1 and x to G_0.

Cost Lemma 2.

$$\eta(\mathbb{A}, \sigma_{1,\rho}) = c_A(T, \rho),$$
$$\eta(\mathbb{A}, \sigma_{2,\rho'}) = c_B(T, \rho').$$

We prove below only the first equality; the second follows by similar arguments.

Proof: Let $V_{\sigma,r}^+ = (V_{\sigma,r,0}^+, V_{\sigma,r,1}^+, V_{\sigma,r,2}^+)$ denote the *global* view of scenario σ after round r. For $\rho \in F_A$, we let $\Delta_{1,2}(\rho)$ denote the set of $\xi \in \{0,1\}^*$ *consistent* with ρ, i.e., which could actually arise as the sequence of bits exchanged between G_1 and G_2 in scenario $\sigma_{1,\rho}$ when $\delta_{1,\rho}'$ is used.

Induction hypothesis: For $r \geq 1$ and any $\xi \in \Delta_{1,2}(\rho)$,

(14)
$$(V_{\sigma_A,r}^+ \mid \xi_{\sigma_A,r,1,2} = \xi) \sim (V_{\sigma_{1,\rho},r}^+ \mid \xi_{\sigma_{1,\rho},r,1,2} = \xi).$$

Suppose for some $r \geq 1$ that (14) holds for all values less than or equal to r. If $r = 3j+1$ then G_0 is transmitting in round r. Since G_0 is nonfaulty in both σ_A and $\sigma_{1,\rho}$ then (14) holds for $r+1$. If $r = 3j+3$ then G_2 is transmitting in round r, and since G_2 is nonfaulty in both σ_A and $\sigma_{1,\rho}$, then (14) also holds for $r+1$. If $r = 3j+2$ then G_1 is transmitting according to the rule specified by the description of $\sigma_{1,\rho}$. This rule implies

$$(V_{\sigma_A,r+1}^+ \mid \xi_{\sigma_A,r+1,1,2} = \xi\rho(u)) \sim (V_{\sigma_{1,\rho},r+1}^+ \mid \xi_{\sigma_{1,\rho},r+1,1,2} = \xi\rho(u))$$

where u is any node of T which could be reached after r rounds using ρ. Since $\xi\rho(u) \in \Delta_{1,2}(\rho)$ implies $\xi\overline{\rho(u)} \notin \Delta_{1,2}(\rho)$, we have proved (14) for $r+1$ in this case as well.

This completes the induction step, and since (14) holds trivially for $r=1$, then it holds for all r.

We now use (14) to prove the following: For each leaf l of T with $\chi(\rho, l) = 1$,

(15)
$$Pr\{\xi_{\sigma_{1,\rho},3t,1,2} = \phi(l)\} = \tau_A(l).$$

This will suffice to establish $\eta(A, \sigma_{1,\rho}) = c_A(T, \rho)$.

Let u be a node at level $2j+1$ along the path $P_e(l)$ from *root*(T) to l. We prove

(16)
$$Pr\{\xi_{\sigma_{1,\rho},3j,1,2} = \phi(u)\} = \tau_A(u).$$

(This will prove (15) by taking $u = l$.) We prove (16) by induction on $j \geq 0$. Trivially, (16) holds for $j=0$. Let $j > 0$ and assume that (16) holds for all values less than j. Let $u' = parent(u)$, $u'' = parent(u')$. Suppose $\phi(u) = \phi(u'')rs$, $r, s \in \{0,1\}$. We know by the induction hypothesis

(17)
$$Pr\{\xi_{\sigma_{1,\rho},3j-3,1,2} = \phi(u'')\} = \tau_A(u'').$$

By the definition of $\sigma_{1,\rho}$ we have

(18)
$$Pr\{\xi_{\sigma_{1,\rho},3j-2,1,2} = \phi(u')\} = Pr\{\xi_{\sigma_{1,\rho},3j-3,1,2} = \phi(u'')\}$$

and

(19)

$$Pr\{\xi_{\sigma_{1,\rho},3j,1,2} = \phi(u)\} = Pr\{\xi_{\sigma_{1,\rho},3j-1,1,2} = \phi(u)\}.$$

Also, by the definition of $\sigma_{1,\rho}$,

$$Pr\{\xi_{\sigma_{1,\rho},3j-1,1,2} = \phi(u)\} = Pr\{\xi_{\sigma_{1,\rho},3j-1,1,2} = \phi(u')s\}$$
$$= Pr\{\xi_{\sigma_{1,\rho},3j-1,1,2} = \phi(u')s \mid \xi_{\sigma_{1,\rho},3j-2,1,2} = \phi(u')\}$$
$$\times Pr\{\xi_{\sigma_{1,\rho},3j-2,1,2} = \phi(u')\}.$$

(20)

By (14) and the fact that G_2 is nonfaulty in both σ_A and $\sigma_{1,\rho}$, we have

$$Pr\{\xi_{\sigma_{1,\rho},3j-1,1,2} = \phi(u')s \mid \xi_{\sigma_{1,\rho},3j-2,1,2} = \phi(u')\}$$
$$= Pr\{\xi_{\sigma_A,3j-1,1,2} = \phi(u')s \mid \xi_{\sigma_A,3j-2,1,2} = \phi(u')\}$$
$$= c(u'u)$$

(21)

By (18)-(21),

(22)

$$Pr\{\xi_{\sigma_{1,\rho},3j,1,2} = \phi(u)\} = c(u'u) \cdot Pr\{\xi_{\sigma_{1,\rho},3j-3,1,2} = \phi(u'')\}.$$

It now follows from (17) and (22) that (16) holds for the value j. This completes the induction step, and Cost Lemma 2 is proved. ■

Combining the various preceding components, we have finally completed the proof of (2), the upper bound on $\beta_{3,1}$. This, together with (1) completes the proof of Theorem 1.

5. General n

As might be expected, the estimation of $\beta_{n,t}$ for general n and t is considerably more complex than the case of $\beta_{3,1}$. In the remainder of this extended abstract, we will restrict ourselves to a very special case of the model which we will call the *one-round broadcast sequential model*. We do this partly for ease of exposition, partly because of space limitations and partly because our results are less complete in this case. We also will be more informal in our description.

For this model, we will assume that the commander G_0 initially transmits the chosen value v to each of the other processors $G_1, \ldots, G_m$. If G_0 is faulty then arbitrary bits can be transmitted to each G_i. These processors then each broadcast in the order $G_1, G_2, \ldots, G_m$, a single bit to all other processors, say G_i broadcasts $\gamma_i \in \{0,1\}$, where of course γ_i depends on everything G_i has heard up to this point, and on G_i's "coin flip," as well. A faulty processor can choose its bit arbitrarily. (Note that we have substantially weakened the power of the faulty processors from the earlier sequential model.) When the last processor G_m completes its broadcast, a *decision function* μ maps the vector $\bar{\gamma} = (\gamma_1, \ldots, \gamma_m)$ into $\mu(\bar{\gamma}) = (\delta_1, \ldots, \delta_m)$, where δ_i is the value decided on by G_i. The specific set of rules by which these various choices are made constitute a protocol $\mathbb{A}$.

We let $\beta'_{n,t}$ denote the maximum achievable success probability $\eta_t(\mathbb{A})$ any such protocol $\mathbb{A}$ can be guaranteed of achieving over all possible scenarios, when we have n processors, at most t of which can be faulty.

Theorem 2. *For all* ϵ *with* $0 < \epsilon < 1$, *if*

$$\frac{t}{n} > 1 - \frac{\log(1-\epsilon)^{1/2}}{\log(1-(1-\epsilon)^{1/2})} \qquad (23)$$

then $\beta'_{n,t} < \epsilon$.

Sketch of Proof: Assume for some protocol $\mathbb{A}$ that (23) holds and $\eta_t(\mathbb{A}) \geq \epsilon$.

We assume for notational simplicity that $m = n - 1 = 2ks$ for integers k and s.

We first consider the scenario σ in which (a faulty) G_0 transmits a 0 to G_i, $1 \leq i \leq ks$, and a 1 to G_j, $ks+1 \leq j \leq 2ks$, and all the other processors, assumed to be nonfaulty, execute the given protocol $\mathbb{A}$. This induces an n-level cost tree T as follows: if $u = parent(u')$ then $c(u\,u') := Pr\{u' \text{ is reached in } \sigma \mid u \text{ is reached in } \sigma\}$ where a node u of T with $\phi(u) = u_1 \cdots u_i$ is identified with the initial set of transmissions $\gamma_1 \cdots \gamma_i$ in the obvious way. (Thus, the terminal vectors $\bar{\gamma}$ can be identified with leaves l of T, so that $\mu(l)$ is well-defined.) If u is a descendant of w in T we write $u \in D(w)$. For $u \in D(w)$, we denote by $\pi(u,w)$ the product $\prod_{e_i} c(e_i)$ where e_i runs over all edges in the path from w to u. We abbreviate $\pi(u, root(T))$ by $\pi(u)$, and for a subset X of nodes of T, we define $\pi(X) := \sum_{x \in X} \pi(x)$.

The basic idea we use for our estimates is the following. Suppose $G_{j+1}, G_{j+2}, \ldots, G_{j+k}$ have all been sent the same bit, say $\beta \in \{0,1\}$ by G_0, and suppose u is a node of T at level $j+1$. Let $D_k(u)$ denote the set of descendants u' of u which are at level $j+k+1$. We say u' is a *β-node* if for every leaf $l \in D(u')$, $\mu(l) = (a_1, a_2, \ldots, a_m)$ has $a_{j+1} = a_{j+2} = \cdots = a_{j+k} = \beta$. Let $B_k(u)$ the set of β-nodes in $D_k(u)$.

Claim. $\pi(B_k(u)) \geq \epsilon$.

This is easy to see since if not, then in the scenario in which G_0 is nonfaulty, $v = \beta$ and the only other nonfaulty processors are $G_{j+1}, \ldots, G_{j+k}$, the faulty processors can make sure that the transmitted γ_i, $1 \le i \le j$, lead to the node u from $root(T)$. In this scenario any successful path must reach a β-node $u' \in D_k(u)$ (otherwise the remaining faulty processors can lead us to a leaf l which has $\mu(l) = (a_1, a_2, \ldots, a_m)$ with $a_i \ne \beta$ for some $j+1 \le i \le j+k$). Thus, this probability, which is just $\pi(B_k(u))$, must be at least ϵ, by the hypothesis on $\mathbb{A}$.

We now partition the processors into $2s$ blocks of k each, by setting $\mathcal{B}_i = \{G_{ki+1}, \ldots, G_{ki+k}\}$, $0 \le i < 2s$. We first apply the Claim to $\mathcal{B}_1$, so that we get $\pi(B_k(root(T))) \ge \epsilon$. We then apply the Claim for $\mathcal{B}_2$ to each $u \in \overline{B_k(root(T))}$, and then for $\mathcal{B}_3$ to each $u' \in \overline{B_k(u)}$, etc. (where if L_l denotes the set of nodes at level l, $\bar{B}$ denotes the complementary set $L_l \setminus B$). It is easy to see that the set X of nodes in level $ks+1$ which are *not* descendents of any of the 0-nodes has $\pi(X) \le (1-\epsilon)^s$. Now we apply the same process to $\bar{X}$, this time throwing out 1-nodes. Again, we do this for s steps (now we have reached the leaves). The construction implies that the set L' of leaves l with $\mu(l)$ having some block of k 0's (in the first half) and some block of k 1's (in the second half) satisfies $\pi(L') \ge (1-(1-\epsilon)^s)^2$.

However, if we finally consider the scenario used to generate T, so that only G_0 is faulty and all other G_i are nonfaulty, then $\mathbb{A}$ succeeds only if some leaf in the complementary set $\overline{L'} \in L'$ is reached. Since

$$\pi(\overline{L'}) \le 1 - (1-(1-\epsilon)^s)^2 < \epsilon$$

by (23) then Theorem 2 follows. ■

We point out that with considerably more effort, these methods can be extended to apply to the general multi-round case, showing that $\beta'_{n,t} = o(1)$ whenever $t > \left(1/2 + \epsilon\right)n$ with similar results applying to $\beta_{n,t}$ as well. We plan to pursue these and related questions in a later paper.

6. Acknowledgements

The authors gratefully acknowledge the insightful comments made by Fan Chung and Anna Karlin during the course of this research.

7. References

[ABDKPR] H. Attiya, A. Bar-Noy, D. Dolev, D. Koller, D. Peleg, and R. Reischuk, "Achievable cases in an asynchronous environment," *Proc. 28th IEEE Symp. on Found. of Computer Science*, (1987), 337-346.

[Be] M. Ben-Or, "Another Advantage of Free Choice: Completely Asynchronous Agreement Protocols," *Proc. 2nd ACM Symp. on Principles of Distributed Computing* (1983), 27-30.

[BLS] M. Ben-Or, N. Linial, and M. Saks, "Collective coin flipping and other models of imperfect information, *Rutcor Research Report 44-87, RUTCOR, Rutgers University*, December, 1987.

[Bra1] G. Bracha, "An Asynchronous $\left\lfloor \frac{(n-1)}{3} \right\rfloor$-Resilient Consensus Protocol," *Proc. 3rd ACM Symp. on Principles of Distributed Computing* (1984), 154-162.

[Bra2] G. Bracha, "An $O(\log n)$ expected rounds randomized Byzantine Generals protocol," *Journal of ACM* 34 (1987), 910-920.

[BD] A. Z. Broder and D. Dolev, "Flipping Coins in Many Pockets (Byzantine Agreement on Uniformly Random Values)," *Proc. 25th IEEE Symp. on Found. of Computer Science* (1984), 157-170.

[CC] B. Chor and B. Coan, "A simply and efficient randomized Byzantine agreement algorithm," *Proc. 4th Symp. on Reliability in Distributed Software and Database Systems* (1984), 98-106.

[CMS] B. Chor, M. Merritt, and D. Shmoys, "Simple Constant-Time Consensus Protocols and Realistic Failure Models," *Proc. 4th ACM Symp. on Principles of Distributed Computing* (1985), 152-162.

[DFFLS] D. Dolev, M. Fischer, M. J. Fowler, N. Lynch, and R. Strong, "Efficient Byzantine agreement without authentication," *Information and Control* 52 (1982), 257-274.

[DLPSW] D. Dolev, N. Lynch, S. Pinter, E. Start, and W. Weihl, "Reaching approximate agreement in the presence of faults," *Journal of ACM* 33 (1986), 499-516.

[DS] D. Dolev and H. R. Strong, "Polynomial algorithms for multiple processor agreement." In *Proceedings* *of the 14th ACM Symp. on Theory of Computing* (1982), 401-407.

[DLS] C. Dwork, N. Lynch and L. Stockmeyer, "Consensus in the presence of partial Synchrony," *Journal of ACM* 35 (1988), 288-323.

[DSS] C. Dwork, D. Shmoys, and L. Stockmeyer, "Flipping coins persuasively in constant expected time," *Proc. 27 IEEE Symp. on Found. of Computer Science* (1986), 222-232.

[FL] M. Fischer, and N. Lynch, "A lower bound for the time to assure interactive consistence," *Inf. Proc. Lett 14*, 4 (1982), 183-186.

[FLM] M. Fischer, N. Lynch, and M. Merritt, "Easy impossibility proofs for distributed consensus problems," *Distr. Comput. 1*, 1 (1986).

[FLP] M. Fischer, N. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *Journal of ACM 32*, 2 (1985), 374-382.

[KKL] J. Kahn, G. Kalai, and N. Linial, "The Influence of Variables on Boolean Functions," *Proc. 29 IEEE Symp. on Found. of Computer Science* (1988), 68-80.

[KY] A. Karlin and A. Yao, "Probabilistic lower bounds for the Byzantine Generals Problem," unpublished.

[LSP] L. Lamport, R. Shostak, and J. Pease, "The Byzantine generals problems," *ACM Trans. Program Lang. Syst. 4*, 2 (1982), 382-401.

[MS] S. Mahaney and F. Schneider, "Inexact agreement: accuracy, precision, and graceful degradation," Preprint, May, 1985.

[PSL] M. Pease, R. Shostak, and L. Lamport, "Reaching Agreement in the Presence of Faults," *Journal of ACM 27* (1980), 228-234.

[R] M. O. Rabin, "Randomized Byzantine Generals," *Proc. 24th IEEE Symp. on Found. of Computer Science* (1983), 403-409.

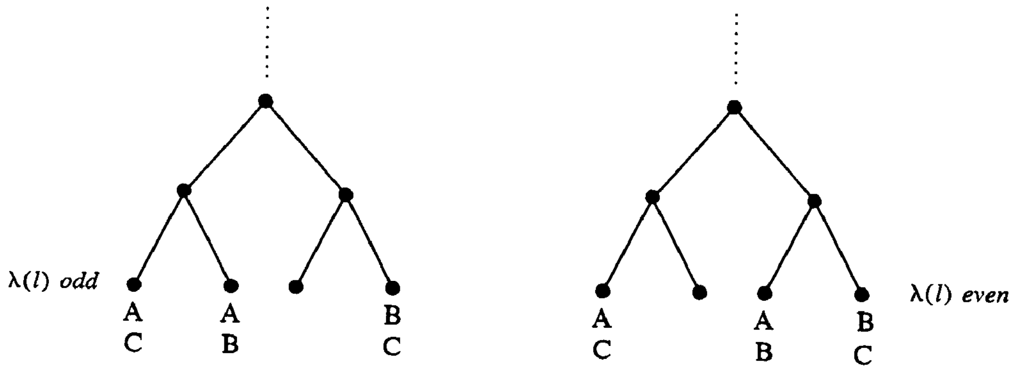


Figure 1

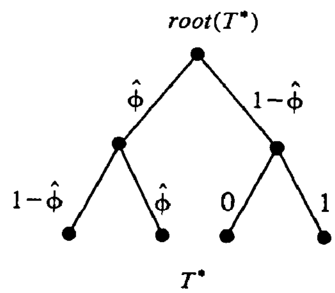


Figure 2

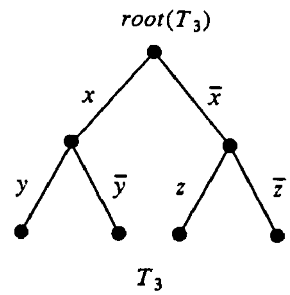


Figure 3

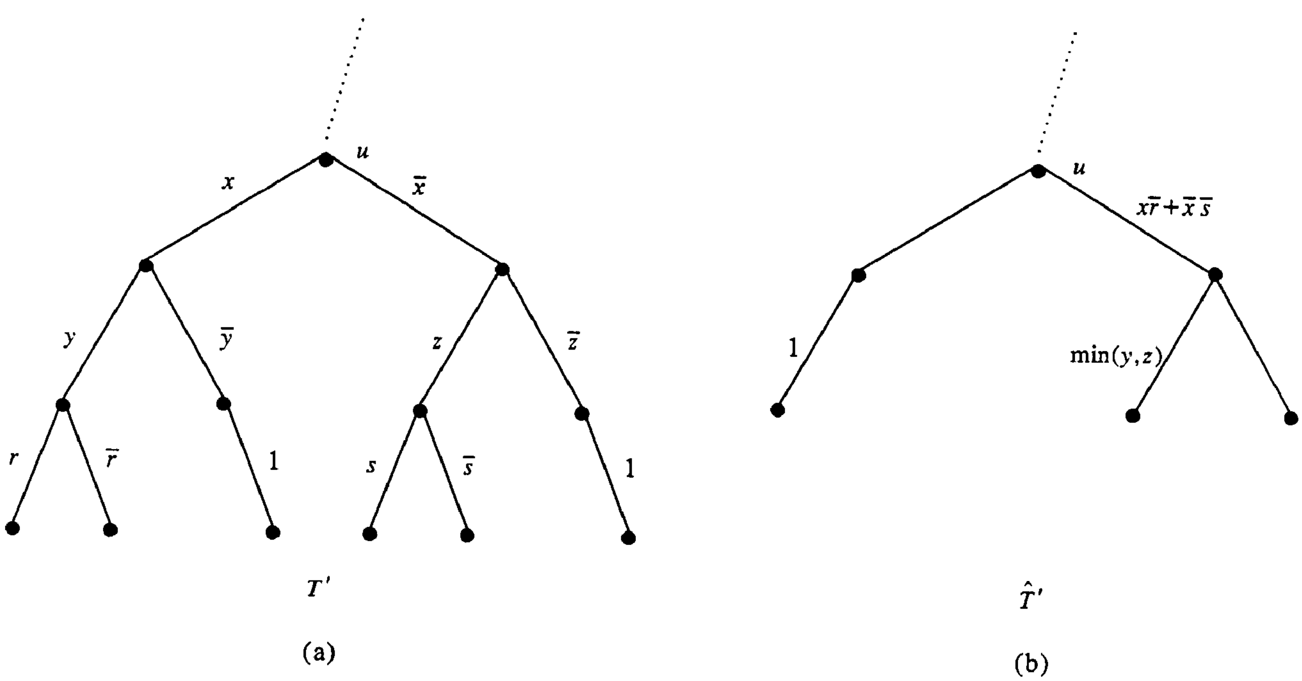
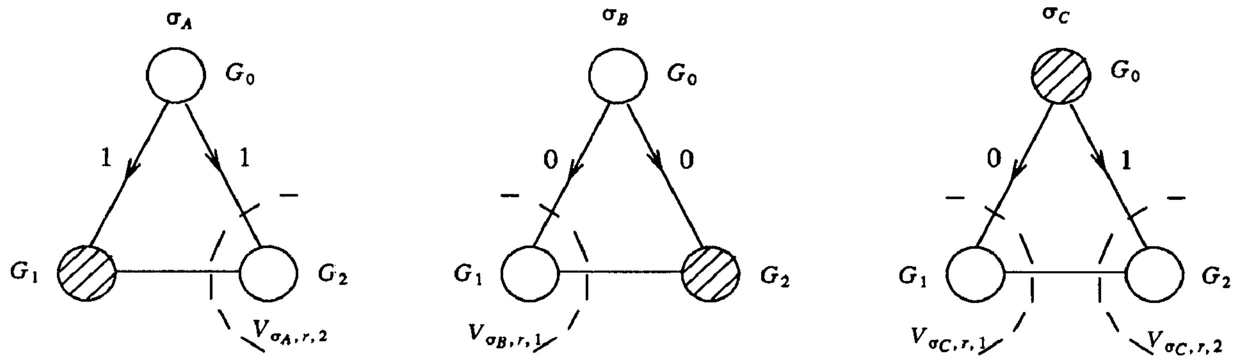


Figure 4



σ_A
G_0
1
1
G_1
G_2
$V_{\sigma_A,r,2}$
σ_B
G_0
0
0
G_1
G_2
$V_{\sigma_B,r,1}$
σ_C
G_0
0
1
G_1
G_2
$V_{\sigma_C,r,1}$
$V_{\sigma_C,r,2}$

Figure 5