A Kraft–Type Inequality for d–Delay Binary Search Codes

R. Ahlswede and N. Cai

1 Introduction

Among the models of delayed search discussed in [1], [2], the simplest one can be formulated as the following two–player game. One player, say A, holds a secret number $m \in \mathcal{M} \triangleq \{1, 2, \ldots, M\}$ and another player, say Q, tries to learn the secret number by asking A at time i questions, like "Is $m \geq x_i$?", where x_i is a number chosen by Q. The rule is that at time $i + d$ A must answer Q's question at time i correctly and at time j Q can choose x_j according to all answers he has received. How many questions has Q at least to ask to get the secret number. Let

$$B_d(t) = \begin{cases} 1 & \text{if } t \leq 0 \\ B_d(t-1) + B_d(t-d-1) & \text{if } t > 0. \end{cases} \tag{1}$$

Then the main result of [1] is

Theorem AMS. (Ambainis–Bloch–Schweizer) *There exists a scheme for Q to win the game by asking t questions iff $M \leq B_d(t)$.*

We notice that the answers are determined by Q's scheme and the secret number, since A does not lie. So for a fixed scheme, for Q wining by asking t questions, each number $m \in \{1, 2, \ldots, M\} = \mathcal{M}$ gives a binary sequence of length at most t in such a way that the ith component of the sequence is zero iff the answer is "yes" if the secret number is m. Thus all successful schemes for Q define a subset in $\{0,1\}^* \triangleq \bigcup_{i=1}^{\infty} \{0,1\}$ and we shall call them d–delay binary search (d–DBS) codes. Then Theorem ABS can be restated: there exists a d–DBS code C whose codewords have at most length t iff

$$|C| \leq B_d(t). \tag{2}$$

For a given d–DBS code we denote by $\ell(c)$ the length of codeword c. Then $\{\ell(c) : c \in C\}$ must satisfy the Kraft inequality, because a d–DBS code has to be prefix free. However a prefix code is not necessarily a d–DBS code. The main result of the paper is a sharper Kraft–type inequality for d–DBS codes based on the work [1]. The inequality is stated and proved in the next section.

R. Ahlswede et al. (Eds.): Information Transfer and Combinatorics, LNCS 4123, pp. 704–706, 2006.

2 The Inequality

Main Inequality:
For all d-DBS codes C,

$$\sum_{c\in C} B_d^{-1}(\ell(c)) \leq 1. \tag{3}$$

Lemma 1. *Let C be a d–DBS code and let L be an integer such that $\ell(c) \leq L$ for all $c \in C$, then*

$$\sum_{c\in C} B_d\big(L-\ell(c)\big) \leq B_d(L). \tag{4}$$

Proof: Originally we got the idea to prove the lemma from [1], and the result follows from Theorem ABS, and the following extension of code C. Let $|C| = M$, let $\mathcal{S}$ be the scheme corresponding to C on $\{0, 1, \dots, M-1\}$, and let

$$M^* = \sum_{c\in C} B_d\big(L-\ell(c)\big).$$

It is sufficient for us to present a successful scheme for Q to win the game by L queries if the secret number is in $\{1, 2, \dots, M^*\}$. Let c_j be the codewords given by secret number j in scheme $\mathcal{S}$ and $\ell(c_j)$ be its length.

Then the scheme with L queries on $\{0, 1, \dots, M^*-1\}$ can be defined as follows.

1. Let $b_m = \sum_{j=0}^{m} B_d\big(L-\ell(c_j)\big)$ for $m = 0, 1, \dots, M-1$.
2. Q first simulates the scheme $\mathcal{S}$. That is, Q asks "$\geq b_m$?" whenever in $\mathcal{S}$ "$\geq m$?" is asked, until a $j \in \{0, 1, \dots, M-1\}$ is found by $\mathcal{S}$. In this case Q knows the "secret number" $m \in \{b_j, b_j+1, \dots, b_{j+1}-1\}$. This takes $\ell(c_j)$ queries.
3. Next Q uses a scheme with $\big(L-\ell(c_j)\big)$ questions achieving $B\big(L-\ell(c_j)\big) = |\{b_j, b_j+1, \dots, b_{j+1}-1\}|$ to find the "secret number" m. □

Lemma 2

$$B_d(\ell_1)B_d(\ell_2) \geq B_d(\ell_1+\ell_2). \tag{5}$$

Proof: We proceed by induction on $\min(\ell_1, \ell_2)$ and w.l.o.g. assume $\ell_1 \leq \ell_2$.

Case $\boldsymbol{\ell_1 \leq 0}$
LHS of (5) $= B_d(\ell_2) \geq B_d(\ell_2 - |\ell_1|) = B_d(\ell_1+\ell_2)$, where "$\geq$" holds because by (1) B_d is non–decreasing.

Case $\boldsymbol{\ell_1, \ell_2 > 0}$
Assume (4) holds for all $\min(\ell_1', \ell_2') < \ell_1 < \ell_2$.

$$
\begin{aligned}
\text{LHS of (5)} &= B_d(\ell_1)B_d(\ell_2) \\
&\overset{(i)}{=} \big(B_d(\ell_1-1)+B_d(\ell_1-d-1)\big)\big(B_d(\ell_2-1)+B_d(\ell_2-d-1)\big) \\
&= B_d(\ell_1-1)B_d(\ell_2-1)+B_d(\ell_1-1)B_d(\ell_2-d-1) \\
&\qquad + B_d(\ell_1-d-1)B_d(\ell_2-1)+B_d(\ell_1-d-1)B_d(\ell_2-d-1) \\
&\overset{(ii)}{\geq} B_d(\ell_1+\ell_2-2)+2B_d(\ell_1+\ell_2-d-2)+B_d(\ell_1+\ell_2-2d-2) \\
&= \Big[\big(B_d(\ell_1+\ell_2-1)-1\big)+B_d\big((\ell_1+\ell_2-1)-d-1\big)\Big] \\
&\qquad \Big[B_d\big((\ell_1+\ell_2-d-1)-1\big)+B_d\big((\ell_1+\ell_2-d-1)-d-1\big)\Big] \\
&\overset{(iii)}{\geq} B_d(\ell_1+\ell_2-1)+B_d(\ell_1+\ell_2-d-1) \\
&\overset{(iv)}{\geq} B_d(\ell_1+\ell_2),
\end{aligned}
$$

where (i) holds by (1), (ii) holds by the induction hypothesis, and (iii) holds, because by (1) we have for all t $B_d(t) \leq B_d(t-1)+B_d(t-d-1)$. □

Apply Lemma 2 to $\ell_1=\ell(c)$ and $\ell_2=L-\ell(c)$ for all $c \in C$, then we obtain

$$B_d\big(L-\ell(c)\big) \geq B_d^{-1}\big(\ell(c)\big)B_d(L). \tag{6}$$

Substituting (6) by (4) we get

$$\sum_{c\in C} B_d^{-1}\big(\ell(c)\big)B_d(L) \leq \sum_{c\in C} B_d\big(L-\ell(c)\big) \leq B_d(L)$$

i.e., (3).

References

1. A. Amboinis, S.A. Bloch, and D.L. Schweizer, Delayed binary search, or playing twenty questions with a procrastinator, Algorithmica, 32, 641-650, 2002.
2. F. Cicalese and V. Vaccaro, Coping with delays and time–outs in binary search proceduresm, Lectures Notes in Computer Science, Vol. 1969, Springer, 96–107, 2000.