Advanced Data Structures Jan-Apr 2012

Lecture 12 March 2 2012

Lecturer: Abhishek Dang *Scribe: Anudhyan Boral and Arjun Arul*

1 Overview

In the last lecture we gave certain cell-probe lower bounds for the predecessor searching problem in static data structures, using reduction to communication games. But, these are known to be not tight. In this lecture we present results due to M.Patrascu and M.Thorup, who achieve tight lower bounds.

2 The Predecessor Problem

Given a set Y of n integers of l bits each, the goal is to efficiently find **predecessor**$(x) = max\{y \in Y | y \leq x\}$. For this purpose, we represent Y on a RAM with word length w using S words of space. Defining $a = lg\frac{S}{n} + lgw$, it has been shown that the optimal search time is, up to constant factors:

$$min \begin{cases} log_w n \\ lg\frac{l-lgn}{a} \\ \frac{lg\frac{l}{a}}{lg(\frac{a}{lgn}.lg\frac{l}{a})} \\ \frac{lg\frac{l}{a}}{lg(\frac{lg\frac{l}{a}}{lg\frac{lgn}{a}})} \end{cases}$$

We present certain important lemmas which are used in proving the above four lower bounds.

3 Cell-Probe Elimination Lemma

An abstract decision data structure problem is defined by a function $f : D \times Q \to \{0, 1\}$. An input from D is given at preprocessing time, and the data structure must store a representation of it in some bounded space. An input from Q is given at query time, and the function of the two inputs must be computed through cell probes. We restrict the preprocessing and query algorithms to be deterministic. In general, we consider a problem in conjunction with a distribution $\mathcal{D}$ over $D \times Q$. Note that the distribution need not (and, in our case, will not) be a product distribution. We care about the probability the query algorithm is successful under the distribution $\mathcal{D}$ (for a notion of success to be defined shortly).

We work in the cell-probe model, and let w be the number of bits in a cell. We assume the querys input consists of at most w bits, and that the space bound is at most 2^w . For the sake of an

inductive argument, we extend the cell-probe model by allowing the data structure to publish some bits at preprocessing time. These are bits depending on the data structures input, which the query algorithm can inspect at no charge. Closely related to this concept is our model for a query being successful. We allow the query algorithm not to return the correct answer, but only in the following very limited way. After inspecting the query and the published bits, the algorithm can declare that it cannot answer the query (we say it rejects the query). Otherwise, the algorithm can make cell probes, and at the end it must answer the query correctly. Thus, we require an a priori admission of any error. In contrast to models of silent error, it actually makes sense to talk about tiny (close to zero) probabilities of success, even for problems with boolean output.

For an arbitrary problem f and an integer $k \leq 2^w$, we define a direct-sum problem $\bigoplus^k f : D^k \times ([k] \times Q) \to \{0,1\}$ as follows. The data structure receives a vector of inputs $(d^1, \ldots, d^k)$. The representation depends arbitrarily on all of these inputs. The query is the index of a subproblem $i \in [k]$, and an element $q \in Q$. The output of $\bigoplus^k f$ is $f(q, d^i)$. We also define a distribution $\bigoplus^k \mathcal{D}$ for $\bigoplus^k f$, given a distribution $\mathcal{D}$ for f.

Given an arbitrary problem f and an integer $h \leq w$, we can define another problem $f^{(h)}$ as follows. The query is a vector $(q_1, \ldots, q_h)$. The data structure receives a regular input $d \in D$, and integer $r \in [h]$ and the prefix of the query $q_1, \ldots, q_{r1}$. The output of $f^{(h)}$ is $f(d, q_r)$. Note that we have shared information between the data structure and the querier (i.e. the prefix of the query), so $f^{(h)}$ is a partial function on the domain $D \times \cup_{i=0}^{t1} Q^i \times Q$. We also have an input distribution $\mathcal{D}^{(h)}$ for $f^{(h)}$, given an input distribution $\mathcal{D}$ for f .

Notation: We give the $f^{(h)}$ operator precedence over the direct sum operator.

The central cell-probe elimination lemma:

Lemma 1. *There exists a universal constant C, such that for any problem f , distribution D and positive integers h and k, the following holds. Assume there exists a solution to $\bigoplus^k f^{(h)}$ with success probability δ over $\bigoplus^k D^{(h)}$, which uses at most $k\sigma$ words of space, $\frac{1}{C}(\frac{\delta}{h})^3 k$ published bits and T cell probes. Then, there exists a solution to $\bigoplus^k f$ with success probability $\frac{\delta}{4h}$ over $\bigoplus^k D$, which uses the same space, $k\sqrt[h]{\sigma}.Cw^2$ published bits and $T-1$ cell probes.*

Definition 2. *$P(n,l)$ is the colored predecessor problem on n integers of l bits each. This is the decision version of predecessor search, where elements are colored red or blue, and a queryjust returns the color of the predecessor.*

The goal is to eliminate all cell probes, and then reach a contradiction. For this, we need the following impossibility result for a solution making zero cell probes:

Lemma 3. *For any $n \geq 1$ and $l \geq log_2(n+1)$, there exists a distribution $\mathcal{D}$ for $P(n,l)$ such that the following holds. For all $(\forall) 0 < \delta \leq 1$ and $k \geq 1$, there does not exist a solution to $\bigoplus^k P(n,l)$ with success probability δ over $\bigoplus^k \mathcal{D}$, which uses no cell probes and less than δk published bits.*

Using the above two Lemmas, we can prove the the optimal predecessor search time is:

$$min \begin{cases} lg\frac{l-lgn}{a} \\ \frac{lg\frac{l}{a}}{lg(\frac{lg\frac{l}{a}}{lg\frac{lgn}{a}})} \end{cases}$$

4 Communication Lower Bounds

We define an $[A; m_1, m_2, m_3, \dots]$-protocol to be a protocol in which Alice speaks first, sending m_1 bits, Bob then sends m_2 bits, Alice sends m_3 bits and so on. In a $[B; m_1, m_2, \dots]$-protocol, Bob begins by sending m_1 bits.

For a communication problem $f : A \times B \to \{0, 1\}$, define a new problem $f^{A,(k)}$ in which Alice receives $x_1, \dots, x_k \in A$, Bob receives $y \in B, i \in [k]$ and $x_1, \dots, x_{i1}$, and they wish to compute $f(x_i, y)$. We define $f^{B,(k)}$ symmetrically, with the roles of Alice and Bob reversed. Assume D is a distribution for f.

The first tool we use is round elimination, which has traditionally been motivated by predecessor lower bounds.

A known result is:

Lemma 4. *Suppose $f^{(k),A}$ has an $[A; m_1, m_2, \dots]$-protocol with error probability at most ϵ on $D^{A,(k)}$. Then f has a $[B; m_2, \dots]$-protocol with error probability at most $\epsilon + O(\sqrt{\frac{m_1}{k}})$ on D.*

Another known Lemma using message compression is as follows:

Lemma 5. *Suppose $f^{(k),A}$ has an $[A; m_1, m_2, \dots]$ protocol with error probability at most ϵ on $D^{A,(k)}$. Then for any $\delta > 0$, f has an $[A; O(\frac{1+(\frac{m_1}{k})}{\delta^2}), m_2, \dots]$-protocol with error probability at most $\epsilon + \delta$ on D.*

Since this lemma does not eliminate Alices message, but merely reduces it, it is used in conjunction with the message switching technique. If Alices first message has a bits, we can eliminate it if Bob sends his reply to all possible messages from Alice (thus increasing his message by a factor of 2^a), and then Alice includes her first message along with the second one (increasing the second message size additively by a):

Lemma 6. *Suppose f has an $[A; m_1, m_2, m_3, m_4, \dots]$-protocol. Then it also has a $[B; 2^{m_1} m_2, m_1 + m_3, m_4, \dots]$-protocol with the same error complexity.*

Message compression combined with message switching represent, in some sense, a generalization of the round elimination lemma, allowing us to trade a smaller k for a larger penalty in Bobs messages. However, the trade-off does yield round elimination as the end-point, because message compression cannot reduce Alices message below $\Omega(\delta^2)$ for any k. We combine these two lemmas to yield a smooth trade-off (with slightly worse error bounds), which is easier to work with:

Lemma 7. *Suppose $f^{(k)}$ has an $[A; m_1, m_2, m_3, m_4, \dots]$-protocol with error ϵ on $D^{A,(k)}$. Then for any $\delta > 0$, f has a $[B; 2^{O(\frac{m_1}{k\delta^4})} m_2, m_1 + m_3, m_4, \dots]$-protocol with error probability $\epsilon + \delta$ on D.*

Proof. If $\frac{m_1}{k} \leq \delta^2$, we can apply the round elimination lemma. Then, Alices first message is ommitted with an error increase of at most δ. None of the subsequent messages change.If $\frac{m_1}{k} \geq \delta^2$, we apply the message compression lemma, which reduces Alices first message to $O(\frac{1+\frac{m_1}{k}}{\delta^2})$ bits, while increasing the error by δ. Since $\frac{m_1}{k} \geq \delta^2$, the bound on Alices message is at most $O(\frac{m_1}{k\delta^4})$. Then, we can eliminate Alices first message by switching. Note our bound for the second message from Alice is loose, since it ignores the compression we have done. □

Using the above four Lemmas, we can prove the the optimal predecessor search time is:

$$min \begin{cases} log_w n \\ \dfrac{lg\frac{l}{a}}{lg(\frac{a}{lgn}.lg\frac{l}{a})} \end{cases}$$