6.897: Advanced Data Structures Spring 2005

Lecture 11 — March 10, 2005

Prof. Erik Demaine *Scribe: Yoyo Zhou*

1 Overview

In this lecture, we discuss lower bounds on the cell-probe complexity of the static predecessor problem. The parameters of interest are w, the size of the machine word, and also the size of input integers, and n, the set size. Observe that a requirement on space is essential for static lower bound, since with space 2^w one can precompute all answers, and respond in constant time.

In this lecture, we concentrat on bounds that are purely in w or n, and we assume the space is $n^{O(1)}$. In particular, we use a technique called round elimination to show that for all w there exists an n such that the problem requires $\Omega(\frac{\lg w}{\lg\lg w})$ time, and for all n there exists a w requiring $\Omega(\sqrt{\frac{\lg n}{\lg\lg n}})$ time. In the next lecture, we will discuss the tradeoff between w, n, and space.

2 Lower Bound Results

We only consider the cell-probe model, and the static problem. We require space to be polynomial in n. Since a static data structure can be constructed through n insertions, this implies the same lower bound on query for the dynamic problem, which holds even if updates take $n^{O(1)}$ time.

The first superconstant bound, proved by Ajtai [Ajt88], was that for all w, there exists n such that query time is $\Omega(\sqrt{\lg w})$. Later, another bound was proved by Miltersen [Mil94], who rephrased the same proof ideas in terms of communication complexity: for all n, there exists w such that query time is $\Omega(\sqrt[3]{\lg n})$. Subsequently, the proof idea was distilled even further into the technique of round elimination, which was used to reprove the same bounds in a very concise way [MNSW95].

Beame and Fich [BF99] proved two stronger bounds: for all w, there exists n such that query time is $\Omega(\lg w/\lg\lg w)$, and for all n, there exists w such that query time is $\Omega(\sqrt{\lg n/\lg\lg n})$. They also gave a data structure achieving $O(\min\{\frac{\lg w}{\lg\lg w}, \sqrt{\frac{\lg n}{\lg\lg n}})\}$, which shows that these bounds are optimal if we insist on pure bound in n or w. These same bounds were also proved earlier and independently by Xiao [Xia92].

Beame and Fich's proof extends that of Ajtai, and is somewhat complicated. Sen [Sen03] later gave a stronger version of the round elimination lemma, which gives a cleaner proof of the same bounds. In this lecture, we will see how to use round elimination to prove the predecessor lower bound, and we will sketch a proof of the round elimination lemma.

The results we have discussed are actually for an easier problem: *colored predecessor*. Each element is colored red or blue; a query on an element x returns the color of x's predecessor. Because we can solve colored predecessor using predecessor, a lower bound for colored predecessor will yield a lower bound for predecessor. Having a lower bound for the simpler problem is useful in reductions to other problems.

3 Communication Complexity

We consider the problem in the communication complexity model. Let Alice represent the query algorithm and Bob represent memory. Alice has an input x, the query, and Bob has an input y, the contents of the data structure. Alice and Bob are only permitted to communicate by sending messages to each other of size at most a and b respectively; we will assume $a = O(\lg n)$, so it is possible to address the entire data structure (using our polynomial space assumption), and $b = w$, so a word of memory is returned. The goal is to compute some function $f(x, y)$; in our case, the function is the color of the predecessor. The parameter of interest is the number of messages sent between Alice and Bob. This is at most twice the number of probes needed in the cell-probe model. Note, however, that the communication model is much stronger, since it allows both parties to perform arbitrary computation (the memory can "think").

3.1 The Predecessor Lower Bound

We will prove an $\Omega(\min\{\lg_a w, \lg_b n\})$ lower bound on the number of messages needed in the communication game. From this, we can derive the Beame and Fich bounds. We have $a = \Theta(\lg n)$, i.e. the memory used is $n^{O(1)}$. Also, $b = w$. The lower bound is worst (smallest) when

$$\lg_a w = \lg_b n \Rightarrow \frac{\lg w}{\lg \lg n} = \frac{\lg n}{\lg w} \Rightarrow \lg^2 w = \lg n \lg \lg n$$

In terms of n, we find $\lg w = \sqrt{\lg n \lg \lg n}$, so the bound becomes $\lg_a w = \sqrt{\frac{\lg n}{\lg \lg n}}$. In terms of w, we find $\lg \lg w = \Theta(\lg \lg n)$, so the bound is $\lg_b n = \frac{\lg w}{\lg \lg w}$.

4 Round Elimination

Round elimination can be applied to an abstract communication game (not neccessarily related to the predecessor problem). It gives some conditions under which the first round of communication can be eliminated. To do this, we consider the "k-fold" of an arbitrary function f:

Definition 1. *Let $f^{(k)}$ be a variation on f, in which Alice has the k inputs $x_1, \ldots, x_k$, and Bob has inputs: y, $i \in 1, \ldots k$, and $x_1, \ldots, x_{i-1}$ (note that this overlaps with Alice's inputs). The goal is to compute $f(x_i, y)$.*

Now assume Alice must send the first message. Observe that she must send this message even though she doesn't know i yet. Intuitively, if $a \ll k$, with high probability she is unlikely to send anything useful about x_i, which is the only part of her input that matters. Thus, we can treat the communication protocol as starting from the second message, eliminating the first.

Lemma 2 (round elimination lemma). *Assume there is a protocol for $f^{(k)}$ where Alice speaks first that uses t messages and has error probability δ. Then there is a protocol for f where Bob speaks first that uses $t - 1$ messages and has error probability $\delta + O(\sqrt{a/k})$.*

One can give a good intuition for the result of this lemma. If i is chosen uniformly at random (which is the worst case), in Alice's first message the expected number of bits "about x_i" that are sent is $\frac{a}{k}$. Bob can guess these bits at random; the probability of guessing all bits correctly is $2^{-a/k}$, so the probability of failure is $1 - 2^{-a/k}$. The protocol should make an error either when it did originally (with probability δ), or when Bob guesses these bits incorrectly. Because we are interested in small $\frac{a}{k}$, we have $1 - 2^{-a/k} \approx a/k$. Thus, by eliminating Alice's message, the error probability should increase by about $\frac{a}{k}$. In the real world, this intuition is not entirely correct, and we can only bound the increase in the error by $\sqrt{a/k}$, but this is often enough for applications.

When proving a lower bound, the usual strategy is to use round elimination repeatedly. We first eliminate Alice's first message. Then, we interchange Alice and Bob in the lemma, and we eliminate Bob's first message. We continue doing this, until we are left with no message. If at this point we have a protocol with probability of correctness bounded away from $\frac{1}{2}$, we usually have a contradiction. This is because nontrivial functions of both intputs cannot be computed without any communication, so the best possible error probability is $\frac{1}{2}$ (random guessing). Observe that each time we eliminate a message, we are changing the problem (we had a protocol for $f^{(k)}$, and we obtain a protocol for f). This means that inputs are getting smaller, and it limits the number of times we can eliminate a round.

5 Proof of Predecessor Bound

Let t be the number of cell probes (equivalently, the number of rounds of communication) made by the predecessor algorithm. Our goal is to perform t round eliminations, leaving 0 messages. As we perform more eliminations, we are reducing n and w to some n' and w'. We want to increase the probability of error by at most $\frac{1}{3t}$ each time, so that at the end, we still have a nontrivial success probability (at least $\frac{2}{3}$). If, say, half the elements are red and half are blue, the color of the predecessor cannot be decided with probability better that $\frac{1}{2}$ (random guessing) given no communication at all. So we reach a contradiction. Our lower bound is the number of times we can do the round elimination.

5.1 Eliminating Alice-to-Bob

Alice's input has w' bits (initially, $w' = w$). Divide it into k equal-size chunks $x_1, \ldots, x_k$, where $k = \Theta(at^2)$. Each chunk is an integer of w'/k bits.

We can construct a tree with branching factor $2^{w'/k}$ on the w'-bit strings corresponding to the Alice's possible inputs, which are the elements of the data structure. The tree then has height k. This technique is reminiscent of van Emde Boas, in which we divided the query word into two parts, and in $O(1)$ steps decided that only one what interesting. For the lower bound, we need to divide into more parts, but the idea is the same.

Because we are proving a lower bound, we are free to choose the inputs to make the problem hard. Make the elements in the data structure have a shared prefix of length i with Alice's query and all differ in the i-th chunk. Thus, all elements branch off from the query in the same chunk (the hardest case for van Emde Boas). Alice and Bob know the structure of the inputs, so Bob only needs to know i, the value of the i-th chunk, and $x_1, \ldots, x_i$ (because all of Bob's values must start with this common prefix). Thus, when Alice's message is eliminated, w' is reduced to $w'/k = \Theta(w'/at^2)$. Using the lemma, the error probability increases by $O(1/t)$, which is exactly what we can afford per elimination.

5.2 Eliminating Bob-to-Alice

Now that Alice's message is eliminated, Bob is speaking first, so he doesn't know the query value. Bob's input is n' integers of w' bits. Divide the set into k equal chunks of n'/k integers each, where $k = \Theta(bt^2)$. Remember that fusion trees could recurse in a set of size $n/w^{1/5}$ after $O(1)$ cell probes. Here, we are proving that after one probe, you can only recurse into a set of size $n/w^{O(1)}$, which gives the same bound (because the branching factor is in the logarithm).

Again, we can construct a hard instance. We prefix the integers in each chunk by a value of $\lg k$ bits, giving a unique indentifier of each chunk (the i-th chunk starts with a prefix of i). Alice's query starts with some random $\lg k$ bits, which decides which chunk is interesting. If Bob speaks first, he cannot know which chunk is interesting, so using the lemma, we can eliminate Bob's message the error probability rises by $O(1/t)$.

The elimination reduces n' to $n'/k = \Theta(n'/bt^2)$ and w' to $w' - \lg k = w' - \Theta(\lg bt^2)$. At long as w' does not get too small, $w = \Omega(\lg(bt^2))$, this last term is negligible (say, it reduces w' by a factor of at most 2).

5.3 Stopping

Thus, each round elimination reduces n' to $\Theta(n'/bt^2)$ and w' to $\Theta(w'/at^2)$. Further, the probability of error at the end can be made to be at most $\frac{1}{3}$ by choosing appropriate constants.

We stop the elimination when $w' = O(\lg(bt^2))$ or $n' = 2$. If these stopping conditions are met, we have proven our lower bound: there were many rounds initially, so we could do enough eliminations to reduce n and w to these small values. Otherwise, we have a protocol which gives an answer with zero messages, and the error probability is at most $\frac{1}{3}$, which is impossible. So we must be in the first case (the stopping conditions are met).

Hence, we established a lower bound $t = \Omega(\min\{\lg_{at^2} w, \lg_{bt^2} n\})$. However, because $t = O(\lg n), a \geq \lg n$ and $t = O(\lg w), b = w$, the bases of the logarithms are between a and a^3 and between b and b^3 respectively. Thus, we found $t = \Omega(\min\{\lg_a w, \lg_b n\})$.

6 Sketch of the Proof for the Round Elimination Lemma

6.1 Some Information Theory Basics

Definition 3. *$H(x)$, called the entropy of x, is the number of bits needed on average to represent a sample from a distribution of the random variable x. Formally,*

$$H(x) = \sum_{x_0} \Pr[x = x_0] \cdot \lg \frac{1}{\Pr[x = x_0]}$$

Definition 4. *$H(x \mid y)$ is the conditional entropy of x given y: the entropy of x, if y is known:*

$$H(x \mid y) = E_{y_0}[H(x|y = y_0)]$$

Definition 5. *$I(x : y)$ is the shared information between x and y:*

$$I(x : y) = H(x) + H(y) - H((x, y)) = H(x) - H(x \mid y)$$

$I(x : y \mid z)$ is defined in a manner similar to that of $H(x \mid y)$.

6.2 The Round Elimination Lemma

Call Alice's first message $m = m(x_1, \ldots, x_k)$. Next, we use a neat theorem from information theory to rewrite entropy as a sum:

$$a = |m| \geq H(m) = \sum_{i=1}^{k} I(x_i : m \mid x_1, \ldots, x_{i-1})$$

If i is distributed uniformly in $\{1, \ldots, k\}$, then $E_i[I(x : m \mid x_1, \ldots, x_k)] = \frac{H(m)}{k} \leq \frac{a}{k}$. This is why $\frac{a}{k}$ was an estimate for how many bits of information Bob could learn from the message about Alice's message. Note that we bounded $I(x_i : m \mid x_1, \ldots, x_{i-1})$, so even if Bob already knows $x_1, \ldots, x_{i-1}$ and receives m, he still learns at most $\frac{a}{k}$ bits about x_i.

The prove the lemma, we must build a protocol for f given the assumed protocol for $f^{(k)}$. We can build a protocol $f(x, y)$ as follows:

1. Fix $x_1, \ldots, x_{i-1}$ and i a priori (known to both players) at random.
2. Alice pretends $x_i = x$.
3. Run the $f^{(k)}$ protocol, starting at the second message, by assuming the first message is $m = m(x_1, \ldots, x_{i-1}, \tilde{x}_i, \ldots, \tilde{x}_k)$, where $\tilde{x}_j$ is a random variable drawn from the distribution of x_j. Now the first message does not depend on $x_i = x$ (even x_i is chosen randomly), so Bob can generate it by himself, without any initial message from Alice.
4. Now Alice has some actual x, which she must use as x_i, and almost certainly $\tilde{x}_i \neq x$. But we know that $I(x_i : m)$ is very small, so the message doesn't really depend on x_i in a crucial way. This means that a random message was probably good: Alice can now fix $x_{i+1}, \ldots, x_k$, so that $m(x_1, \ldots, x_{i-1}, \tilde{x}_i, \ldots, \tilde{x}_k) = m(x_1, \ldots, x_{i-1}, x, \ldots, x_k)$, for the desired $x_i = x$.

The last step is the crucial one, and it is what introduces an error probability of $O(\sqrt{a/k})$. This is proved based on the "Average Encoding Theorem" from information theory. There is also a more subtle problem that this theorem solves: not only must $x_{i+1}, \ldots, x_k$ exist, so that a Bob's random guess for a message is made valid, but their distributions are close the the original distributions, so the error probability δ does not increase too much.

References

[Ajt88] M. Ajtai: *A lower bound for finding predecessors in Yao's cell probe model*, Combinatorica 8(3): 235-247, 1988.

[BF99] P. Beame, F. Fich: *Optimal Bounds for the Predecessor Problem*, Symposium on the Theory of Computing 1999: 295-304.

[Mil94] P. Miltersen: *Lower bounds for union-split-find related problems on random access machines*, Symposium on the Theory of Computing 1994: 625-634.

[MNSW95] P. Miltersen, N. Nisan, S. Safra, A. Wigderson: *On data structures and asymmetric communication complexity*, Symposium on the Theory of Computing 1995: 103-111.

[Sen03] P. Sen: *Lower bounds for predecessor searching in the cell probe model*, IEEE Conference on Computational Complexity 2003, 73-83.

[Xia92] B. Xiao: *New bounds in cell probe model*, PhD thesis, University of California, San Diego, 1992.