Lecture 3: Basic Results

1 Review

Last time, we introduced the Turing Model, and the Pointer Model. All complexity classes are defined using the Turing model, and yet, by our invariance theses, essentially the same canonical classes would be obtained if we use, say, the Pointer Model.

2 THE BASIC RESULTS

We highlight the basic ideas (which are mostly simple), by proving weaker versions of the theorems than are stated in the lecture notes. We also highlight the technical applications of these results.

2.1 (A) LINEAR REDUCTION OF COMPLEXITY

We show a progression of three results:

- RESULT A: $XSPACE(s) \subseteq XSPACE(s/2)$, $X = N, D$.

  Proof: Let M be a k-tape TM accepting in space $s(n)$. We construct a k-tape TM N to simulate M in space $s(n)/2$. If Σ is the tape symbols of M, then $\Sigma' \times \Sigma'$ is the tape symbols of N where $\Sigma' = \Sigma \cup \{\Box\}$. The actual simulation is now immediate.

- RESULT B: $DTIME(t) \subseteq DTIME(n + t/2)$

  Proof: This is similar to the previous simulation except for some additional details.

  If M is k-tape, then N is $(k+1)$-tape. The extra tape stores the "compressed input".

  First, instead of compressing 2 symbols into one, we compress c symbols into one. We determine c later.

  Second, before we start, we copy the input word into the extra tape but using compressed form (using n/c cells). This takes n steps. From now on, we use this compressed form as input for our simulation. To begin the simulation, we must move the tape head back to the start of the compressed input, using n/c steps.

  Finally, to carry out the simulation, we need to do a sequence of 8 moves: first we make 4 moves to determine the contents of the cells on each side of the currently scanned cells (call these cells the current neighborhood). Now we know what will happen when machine M first moves OUT of the current neighborhood, and where its head would be located. In another 4 moves, we update current neighborhood and move the head of N to the new composite cells.

Of course, if M halts before moving out of the current neighborhood, N can detect this and act accordingly.

Note that in order to move outside of the current neighborhood, M must have made at least c moves. Hence we have replaced at least c moves by 8 moves. The total time is therefore $n + (n/c) + 8t(n)/c$. This is at most $n + t(n)/2$ if $c = 18$.

- RESULT C: $NTIME(t) \subseteq NTIME(max\{n, t/2\})$.

  Note the surprising corollary that "nondeterministic realtime = nondeterministic linear time":

  $$NTIME(n+1) = NTIME(O(n))$$

  Prooff: This amounts to the previous proof in which we combine the "input compression" phase with the simulation phase. We can do this because we have nondeterminism on our side.

DISCUSSION: Technical importance of these results is that we can ignore the constant factors in complexity functions.

2.2 TAPE REDUCTION

We only briefly state the main results here:

- Can replace a XTM M accepting in time t by a 1-tape TM N acepting in time t^2.
- Can replace a DTM M accepting in time t by a 2-tape TM N acepting in time $O(t \log t)$.
- Can replace a NTM M accepting in time-space (t,s) by a 2-tape TM N acepting in time $O(t, s)$.

DISCUSSION: such result are important because we need the notion of "EFFICIENT UNIVERSAL MACHINE" to simulate all other machines. Such a universal machine must use a fixed number of tapes, which these theorems give us.

2.3 TIME SIMULATION

Again, we only state the key results here:

- $XSPACE - REV(s, r) \subseteq XTIME(O(n+s)r)$.
- $NSPACE(s) \subseteq DTIME(n.O(1)^s)$.
- $NTIME(s) \subseteq DTIME(O(1)^t)$.
- $DREVERSAL(r) \subseteq DTIME(n.O(1)^r)$.

These results uses a simple fact: counting of number of configurations. This does not seem very deep – yet this is the state of the art! RESEARCH PROBLEM: Is there an inherent reason why we cannot do any better?

2.4 SPACE SIMULATION

We will prove Savage's Theorem (generalized): suppose $t(n) \geq 2s(n) \geq \log n$. Then

$$\textit{N-TIME-SPACE}(t,s) \subseteq \textit{DSPACE}(s\log(t/s))$$

The proof uses two techniques:

Fix a NTM M and an input x of length n. Let the predicate $PATH_h(C,C',m)$ indicate if there is a path from C to C' in at most m steps using configurations that uses at most h space.

Let $h \geq \log n$.

CLAIM A: Can evaluate $PATH_h(C,C',m)$ in space $O(m+h)$.

CLAIM B: Can evaluate $PATH_h(C,C',m)$ tin space $O(h\log(m/h)$.

Note that CLAIM B leads us to the theorem immediately, if we con compute $h = s(n)$ and $m = t(n)$ using space at most $s\log(t/s)$. But since we do not assume $s(n)$ and $t(n)$ can be easily computed, we can do a doubly-nested loop to search for h and m.