Lecture 13

1 Alternating Complexity, COncluded

We complete the proof of

Theorem: $PrTIME(t) \subseteq ATIME(t)$

and present the major result of Hopcroft, Paul and Valiant (1977) showing that

$$DTIME(t) \subseteq DSPACE(t/logt).$$

This was sharpened by Tompa and Dymond, by replacing DSPACE by ATIME. The key technical idea is a space-efficient pebbling game.

2 OUTLINE

Suppose a deterministic TM M accepts in time t. Let $B = B(t)$ be the blocking factor to be fixed.

- Define $t_i = iB$ ("sampling times") for $i = 0, 1, \ldots, t/B$.
- Define tape blocks, comprising B consecutive cells in a work tape.
- For any block b, let $Nbhd(b)$ comprise b and its two adjacent blocks.
- Define dag $G = (V, E)$ where $V = \{0, ..., t/B\}$. Each node $i \in V$ has "labels" comprising (assumed) positions of each tape head and the state at time t_i.
- Define: Block b is "(formally) visited" in time t_j.
- Edge set E depends on the labels: $(i, j) \in E$ iff "the information at time t_i can affect the computation at time t_j."
- The in-degree of G is $\leq 5k$ if there are k worktapes.
- "Expansion" of block i: the contents of the blocks visited in i at time t_i. "Contraction" is just the reverse – discard the expanded information. Expansion is correct if it is REALLY true contents at time t_i.
- OBSERVE: if we have CORRECT expansions of all predecessors of i then we can CORRECTLY expand node i.
- Pebbling Game on dags: G has a goal node v_0. A "pebbling" P is the placement of pebbles on the nodes, at most one per node. So a node is either pebbled or it is not. Relative to P: a node "can be pebbled" if all its predecessors are pebbled. A node can be unpebbled if it is pebbled. A play is a sequence of pebbling or unpebbling steps, with the last step being the pebbling of v_0, and starting out with no pebbled nodes. The "pebble cost" of the play is the maximum number of pebbled nodes at any time.

- THEOREM: every bounded degree dags can be pebbled witn $O(n/\log n)$ pebbles.
- ALTERNATING VERSION of pebbling game: A pebbling tree T for $G = (V, E)$ is one where each vertex $u \in T$ is labelled with a triple

  $$[i, X, Y]$$

  where $i \in V$ (challenged node), $X \subseteq V$ (pebbling set) and $Y \subseteq V$ (unpebbling set). The "currently pebbled set" $C(u)$ is defined as X if u is root, else if u' is parent of u then

  $$C(u) = (C(u') - Y) \cup X.$$

  At each vertex u with label $[i, X, Y]$, we have:

  1. The challended node at root is the goal v_0.
  2. Either $i \in X$ or all predecessors of i are in $C(u)$.
  3. u has $|X|$ children, and X is the set of challenged nodes at these children.

  NOTE: the pebbling set at leaves must therefore be empty!
- Connection of this tree to alternating computation to verify the labels of a dag G.

  DEFINE: pebbling time at vertex u is $|X| + |Y|$. Pebbling time of tree is maximum sum of pebbling along any path. Pebbling space is the maximum of $|C(u)|$ over all u.
- Theorem: pebbling time is $O_d(m/logm)$ for dags with indegree d and m edges.

  Proof is by partitioning the graph into two roughly equal parts, and consider 3 cases.