Lecture 7

Lance Fortnow(notes for 01/23/08 scribed by Sanchit Misra)

January 25, 2008

1 Immerman-Szelepcsényi Theorem

NSPACE(s(n)) = Co.NSPACE(s(n))

For any language L $\in$ NSPACE(s(n)) and corresponding NTM M, we can construct a directed graph with node for every configuration of NTM M. If there is a path from $Conf_0$ to any accepting configuration, a non deterministic TM can guess it. But for Co.NSPACE(s(n)), we want to non-deterministically check that there is no path from $Conf_0$ to any accepting configuration. Now if we guess one path and it doesn't reach an accepting configuration, we can't claim that there is no such path.

Let m: # of nodes that are reachable from $Conf_0$. Suppose we know m. Then we can use the following algorithm to check that there is no path from $Conf_0$ to any accepting configuration.

- *Algorithm*

  Let $a = 0$
  For every conf not accepting
      Guess path from $conf_0$ to conf
      If found $a = a + 1$
  rof

  If $a = m$ then accept else reject.

- Claim: This algorithm accepts exactly when there is no path from initial configuration to any accepting configuration.

- $a \leq c^{s(n)} \Rightarrow \log(a) = O(s(n))$ and space for all configurations = O(s(n))
  Total space = O(s(n))
- Problem: We don't know m

To calculate m:

- Let m_i = # nodes reachable from $conf_0$ by a path of length $\leq$ i.
- $m_0 = 1$, m_{N-1} = m, where N is the number of nodes in the configuration graph.

Assume we know m_i, find m_{i+1}

- *Algorithm*

```
Let m_{i+1} = 0
For every conf /* see if conf_0 reaches conf in i+1 steps */
    a = 0
    for every conf'
        Guess a path from conf_0 to conf' of length <= i
        If found
            a = a+1
            If (conf' = conf or conf' -> conf in 1 step) then
                m_{i+1} = m_{i+1} + 1
                Goto next conf
    rof
    If a < m_i crash & reject
rof
```

Total space required = O(s(n))

2 Other Notes

Earlier proved result:
DTIME(t(n)) $\subseteq$ NTIME(t(n)) $\subseteq$ DSPACE(t(n)) $\subseteq$ NSPACE((t(n)) $\subseteq \bigcup_{c>0}$ DTIME($c^{t(n)}$)
$\Rightarrow$ NTIME(t(n)) $\subseteq \bigcup_{c>0}$ DTIME($c^{t(n)}$) $\subseteq \bigcup_{c>0}$ Co.NTIME($c^{t(n)}$)

2.1 More results to be proved in forthcoming classes

- If we have more space, we can do more things:
  DSPACE(n^2) $\subset$ DSPACE(n^3)

- In general, if two functions s_1(n) and s_2(n) ($\geq$ log(n)) are "space constructible" and s_1(n) = o(s_2(n)) (i.e. $\lim_{n\to\infty} \frac{s_1(n)}{s_2(n)}$) then
  DSPACE(s_1(n)) $\subset$ DSPACE(s_2(n))

  Def. *Space Constructible*: s(n) is space constructible if there is a TM M such that M(1^n) outputs $1^{s(n)}$ on output tape and uses O(s(n)) space.

  Every function that we normally use(natural functions) are space constructible. We don't normally encounter non-space constructible functions.

  Def. t(n) is *Time constructible* if there is a TM M with an output tape. M(1^n) outputs $1^{t(n)}$ in time O(t(n)).

- If t_1(n) and t_2(n) are time constructible and $\lim_{n\to\infty} \frac{t_1(n)log(t_1(n))}{t_2(n)} \to 0$ then
  DTIME(t_1(n)) $\subset$ DTIME(t_2(n))

- If $\lim_{n\to\infty} \frac{t_1(n+1)}{t_2(n)} \to 0$ then
  NTIME(t_1(n)) $\subset$ NTIME(t_2(n))