Computational Complexity. Lecture 9

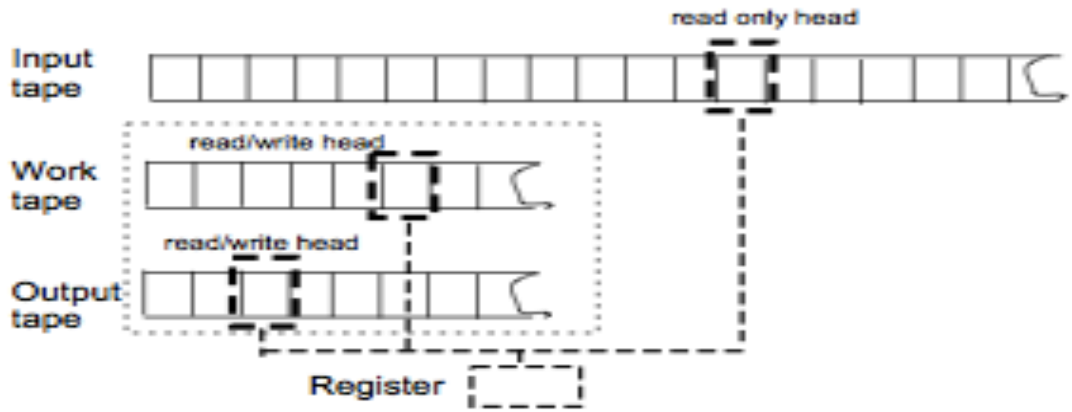
Space complexity.

Alexandra Kolla

Today

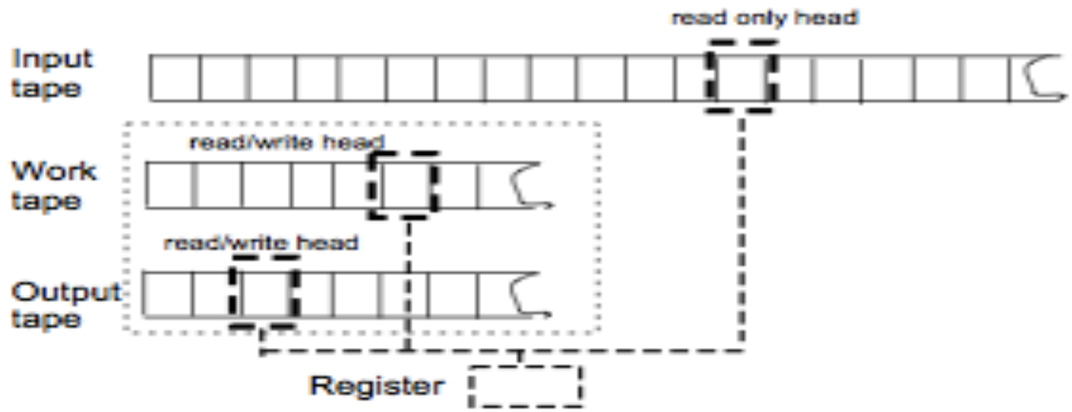
- Space Complexity, L,NL
- Configuration Graphs
- Log- Space Reductions
- NL Completeness, STCONN
- Savitch's theorem
- SL

Turing machines, briefly



- (3-tape) Turing machine M described by tuple (Γ,Q,δ), where
  - Γ is "alphabet" . Contains start and blank symbol, 0,1,among others (constant size).
  - Q is set of states, including designated starting state and halt state (constant size).
  - Transition function $\delta : Q \times \Gamma^3 \rightarrow Q \times \Gamma^2 \times \{L, S, R\}^3$ describing the rules M uses to move.

Turing machines, briefly



- (3-tape) NON-DETERMINISTIC Turing machine M described by tuple (Γ,Q,δ_0,δ_1), where
  - Γ is "alphabet" . Contains start and blank symbol, 0,1,among others (constant size).
  - Q is set of states, including designated starting state and halt state (constant size).
  - Two transition functions $\delta_0, \delta_1 : Q \times \Gamma^3 \rightarrow Q \times \Gamma^2 \times \{L, S, R\}^3$. At every step, TM makes non-deterministic choice which one to

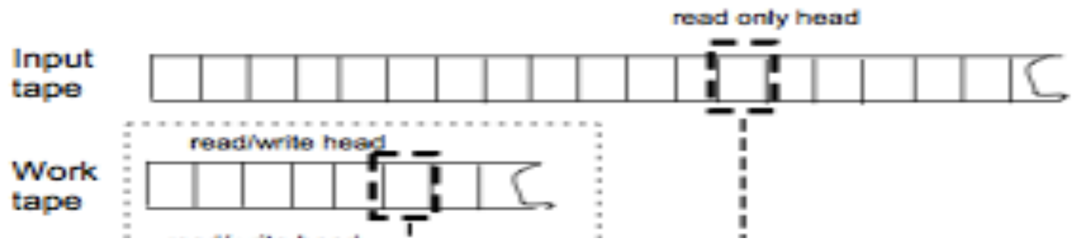
Space bounded turing machines

- Space-bounded turing machines used to study memory requirements of computational tasks.

- **Definition**. Let $s: \mathbb{N} \rightarrow \mathbb{N}$ and $L \subseteq \{0,1\}^*$. We say that L∈ SPACE(s(n)) if there is a constant c and a TM M deciding L s.t. at most c·s(n) locations on M's work tapes (excluding the input tape) are ever visited by M's head during its computation on every input of length n.
- We will assume a single work tape and no output tape for simplicity.

- Similarly for NSPACE(s(n)), TM can only use c·s(n) nonblank tape locations, regardless of its nondeterministic choices

Space bounded turing machines



- Read-only "input" tape.
- Read/write "work" or "memory" tape.
- We say that machine on input x, uses space s if it only uses the first s(|x|) cells of the work tape.
- Makes sense to consider TM that use less memory than length of input, need at least log n

Space complexity

- DTIME(s(n)) ⊆SPACE(s(n)) clearly.
- SPACE(s(n)) could run for as long as $2^{\Omega(s(n))}$ steps, can reuse space (i.e. count from 1 to $2^{s(n)-1}$ by maintaining counter of size s(n)).
- Next theorem shows this is tight, and it is the only relationship we know between the power of space-bounded and time-bounded computation.

Space vs. time complexity

Theorem 1. If a machine always halts, and uses s(.) space, with s(n)≥log n, then it runs in time $2^{O(s(n))}$.

Configuration graphs

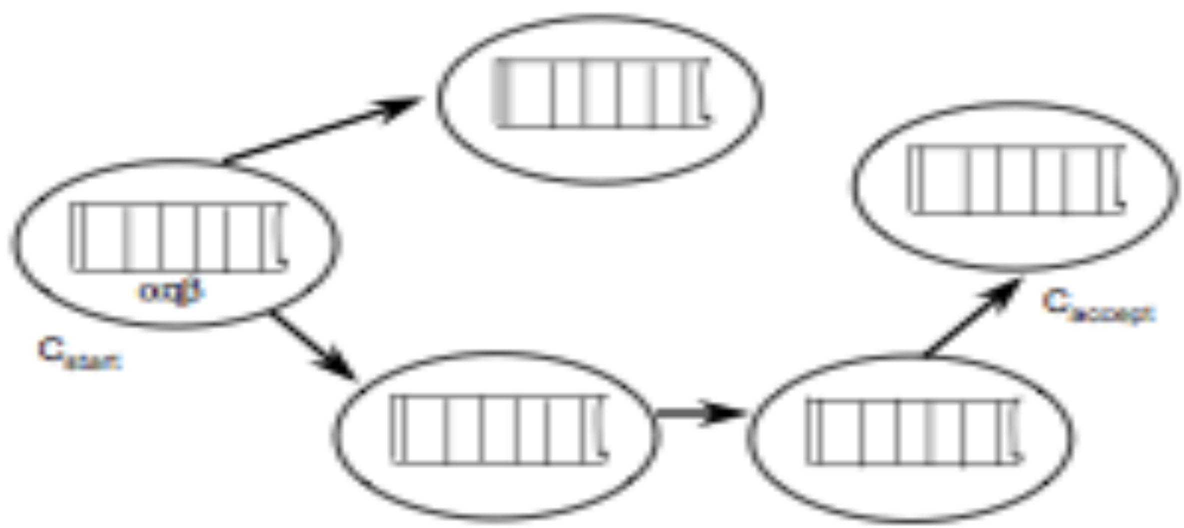
- Configuration of a TM M consists of contents of all non-blank entries of M's work tape, along with its state and head position on input tape, at a particular point in its execution.
- For every space s(n), TM M and input x, the configuration graph of M on input x, denoted $G_{M,x}$ is a directed graph whose nodes correspond to all possible configurations of M(x).

Configuration graphs

- $G_{M,x}$ has directed edge from config. C to config C′ if C′ can be reached from C in one step, according to M's transition function.
- If M deterministic, then graph has out-degree one.
- If M non-deterministic, then graph has out-degree two.
- Can assume w.l.o.g. only one accept configuration C_{accept}, on which M halts and outputs 1.

Configuration graphs

- M accepts input x iff there is directed path in $G_{M,x}$ from C_{start} to C_{accept}



Configuration graphs

- **Lemma**. Every vertex in $G_{M,x}$ can be described by using c·s(n) bits and, in partricular, $G_{M,x}$ has at most $2^{cs(n)}$ nodes.

Space vs. time complexity, II

Theorem 2. If DTM or NDTM halts, then

DTIME(s(n)) ⊆SPACE(s(n)) ⊆NSPACE(s(n)) ⊆DTIME($2^{O(s(n))}$)

Some space complexity classes

- PSPACE= $\cup_{c>0}$SPACE(n^c)
- NPSPACE= $\cup_{c>0}$NSPACE(n^c)
- L=SPACE(log n)
- NL= NSPACE(log n)

- Is NL the space analog of NP? (NL= set of decision problems with solutions that can be verified in log space?)

- **Corollary**. NL$\subseteq$P

Reductions in NL

- Would like to introduce notion of completeness in NL, analogous to the completeness we know for NP.
- For meaningful such notion, we cannot use poly-time reductions (otherwise every NL problem having at least a YES and a NO instance would be complete).
- Need weaker reductions.

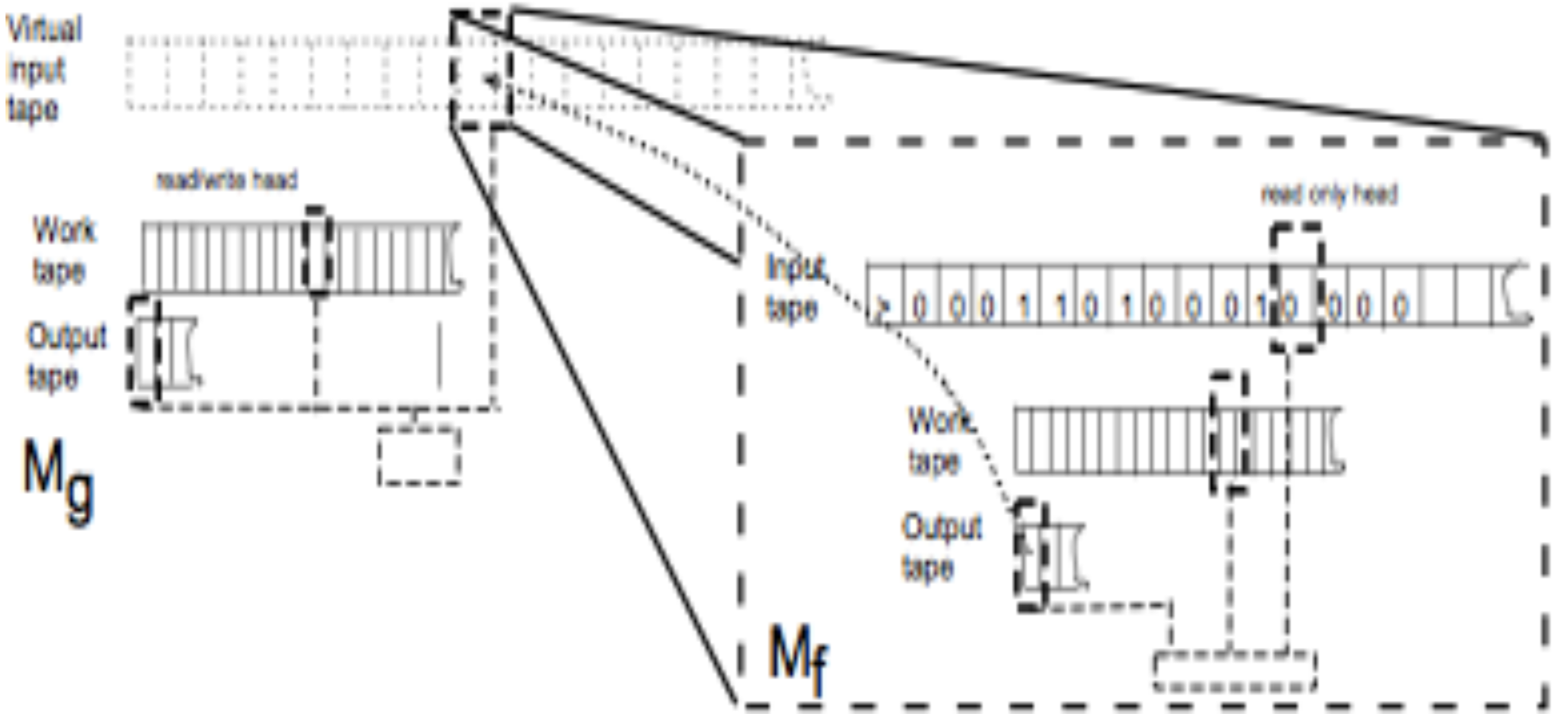
Reductions in NL

- **Definition** (log-space reductions). Let A and B be decision problems. We say that A is log space reducible to B, $A \leq_{log} B$, if there is a function f computable in log space such that x∈ A iff f(x) ∈ B and B ∈ L.

Reductions in NL

- **Theorem**. If B ∈ L and $A \leq_{log} B$, then A ∈ L

Reductions in NL



Reductions in NL

- **Theorem**. If $A \leq_{log} B$, B$\leq_{log} C$, then $A \leq_{log} C$.

NL Completeness

- **Definition**. A is NL-hard if for all B∈ NL, B$\leq_{log}$ A. A is NL-complete is A ∈ NL and A is NL-hard.

- STCONN (s,t-connectivity). Given in input a directed graph G(V,E) and two vertices s,t ∈ V, we want to determine if there is a directed path from s to t.

NL Completeness

- **Theorem**. STCONN is NL-complete.

Savitch's theorem

- What kind of tradeoffs are there between memory and time?
- E.g STCONN can be solved deterministically in linear time and linear space, using depth-first search.
- Can searching be done deterministically in less than linear space?

Savitch's theorem

- **Theorem**. If A is a problem that can be solved non-deterministically in space s(n)≥logn, then in can be solved deterministically in space $O(s^2(n))$.
- **Corollary.** STCONN can be solved deterministically in $O(log^2n)$ space.

Savitch's theorem

Corollary. STCONN can be solved deterministically in $O(log^2 n)$ space.

- Exponentially better space than deapth-first search, no longer poly time.
- Time required by Savitch's algorithm is super-poly.
- No known algorithm simultaneously achieves poly time and polylog space.

ST-UCONN and symmetric non-deterministic machines

- Undirected s,t, connectivity ST-UCONN: we are given undirected graph and the question is if there is path from s to t.
- Not known to be complete for NL, probably not, but complete for class SL (symmetric, non-deterministic TM with O(log n) space).
- Non-deterministic TM is symmetric if whenever transition s-s’ possible, so is s’-s.
- Same proof of completeness, since transition graph now is undirected.

An incomplete picture of what we know

- L $\subseteq$SL $\subseteq$ NL $\subseteq$ P $\subseteq$ NP $\subseteq$ PSPACE $\subseteq$ EXP
- We (should) know that P $\subsetneq$EXP and we will see L $\subsetneq$ PSPACE so some inclusions not strict. Maybe all?
- Reingold '04 showed in a breakthrough result that L=SL.