

Homework 1 (due Oct 5)
Borja Sotomayor

Exercise 1 We are asked to prove the following relations:

- (a) $\mathcal{L} \subseteq \mathcal{NL}$
- (b) $\mathcal{NL} \subseteq \mathcal{P}$
- (c) $\mathcal{P} \subseteq \mathcal{NP}$
- (d) $\mathcal{NP} \subseteq \mathcal{PSPACE}$
- (e) $\mathcal{PSPACE} \subseteq \mathcal{EXPTIME}$
- (f) $\mathcal{EXPTIME} \subseteq \mathcal{NEXPTIME}$

We claim that $\mathcal{DSPACE}(f(n)) \subseteq \mathcal{NSPACE}(f(n))$ and $\mathcal{DTIME}(f(n)) \subseteq \mathcal{NTIME}(f(n))$. Proof: Every DTM is a NTM where the number of ‘choices’ at each step is exactly 1 (see Theorem 7.4(a) in [1]). Relations (a), (c), and (f) follow immediately from this.

(b) [The following explanation is quite succinct. I realize that a more adequate proof would be something along the lines of Theorem 7.4(c) in [1], but I did not completely understand that proof]. We know that REACHABILITY is \mathcal{NL} -Complete (Theorem 16.2 in [1]) and there is a polynomial-time algorithm for REACHABILITY (BFS or DFS). Therefore, given an arbitrary language $L \in \mathcal{NL}$, we can reduce it in polynomial time to REACHABILITY, which can be solved itself in polynomial time. Therefore, $L \in \mathcal{P}$ and $\mathcal{NL} \subseteq \mathcal{P}$.

(d) For this relation, it is convenient to think of \mathcal{NP} as the ‘set of languages for which a witness can be verified in polynomial time’. Now, given an arbitrary language $L \in \mathcal{NP}$ we can compute it by iterating through every possible witness (which must necessarily have polynomial length). Although horribly inefficient in time, this computation only uses polynomial space, so $L \in \mathcal{PSPACE}$. See Theorem 7.4(b) in [1] for a more general proof ($\mathcal{NTIME}(f(n)) \subseteq \mathcal{DSPACE}(f(n))$).

(e) Languages in \mathcal{PSPACE} can be computed using polynomial space. Suppose that computing an arbitrary language $L \in \mathcal{PSPACE}$ requires a machine M with s bits (in the work tape/s) and q states. It follows that the running time cannot exceed $2^s \cdot q$ (which would mean going through all possible configurations of the machine). Therefore, the running time is, at most, exponential and $L \in \mathcal{EXPTIME}$.

Exercise 2 We are asked to prove that $\mathcal{P} \subseteq \mathcal{EXPTIME}$. First of all, we remember the definition of $\mathcal{EXPTIME}$:

$$\mathcal{EXPTIME} = \bigcup_{k \in \mathbb{N}} \mathcal{DTIME}(2^{n^k})$$

Claim: $\mathcal{P} \subseteq \mathcal{DTIME}(2^n)$. Proof: Any polynomial eventually becomes smaller than 2^n . Note that, by the above definition, $\mathcal{DTIME}(2^n) \subseteq \mathcal{EXPTIME}$ and we immediately have that $\mathcal{P} \subseteq \mathcal{EXPTIME}$.

However, we must prove that \mathcal{P} is a *proper* subset of $\mathcal{EXPTIME}$. To prove this we use the *deterministic time hierarchy theorem* [1, 4] which states that:

$$\mathcal{DTIME}(f(n)) \subset \mathcal{DTIME}(f(2n+1)^3)$$

By this theorem, $\mathcal{DTIME}(2^n) \subset \mathcal{DTIME}(2^{6n+3})$. In turn, $\mathcal{DTIME}(2^{6n+3}) \subseteq \mathcal{DTIME}(2^{n^2})$. And, by the above definition of $\mathcal{EXPTIME}$, we have that $\mathcal{DTIME}(2^{n^2}) \subseteq \mathcal{EXPTIME}$. So, we have that:

$$\mathcal{P} \subseteq \mathcal{DTIME}(2^n) \subset \mathcal{DTIME}(2^{6n+3}) \subseteq \mathcal{EXP}$$

So, $\mathcal{P} \subset \mathcal{EXP}$.

NOTE: I did not use the hint provided, but see that it could be used to prove the deterministic time hierarchy theorem in the case when $f(n) = 2^n$ (see section 7.2 in [1] and also [4]).

Exercise 3 Not solved.

Exercise 4 We are asked to prove that $\mathcal{NP} \subset \mathcal{NEXP}$. We will do so using the *nondeterministic time hierarchy theorem* [4], which states that given a proper complexity function $g(n)$, and a function $f(n)$ such that $f(n+1) = o(g(n))$ then:

$$\mathcal{NTIME}(f(n)) \subset \mathcal{NTIME}(g(n))$$

Similarly to exercise 2, we claim that $\mathcal{NP} \subseteq \mathcal{NTIME}(2^n)$ (the proof is the same: Any polynomial eventually becomes smaller than 2^n). By applying the above theorem, we can see that $\mathcal{NTIME}(2^n) \subset \mathcal{NTIME}(2^{n^2})$ because $2^{n+1} = o(2^{n^2})$. And, since $\mathcal{NTIME}(2^{n^2}) \subseteq \mathcal{NEXP}$ we have that:

$$\mathcal{NP} \subseteq \mathcal{NTIME}(2^n) \subset \mathcal{DTIME}(2^{n^2}) \subseteq \mathcal{EXP}$$

So, $\mathcal{NP} \subset \mathcal{NEXP}$.

Exercise 5 We are asked to prove that $\mathcal{NL} \subset \mathcal{PSPACE}$. We will do so using Savitch's Theorem (seen in class), which states:

$$\mathcal{NSPACE}(s(n)) \subseteq \mathcal{DSPACE}(s^2(n))$$

And the *space hierarchy theorem* (see Corollary 1 in [5]) which states that, for any two functions $f_1(n)$ and $f_2(n)$, where $f_1(n) = o(f_2(n))$ and f_2 is a proper complexity function, then:

$$\mathcal{DSPACE}(f_1(n)) \subset \mathcal{DSPACE}(f_2(n))$$

Claim 1: $\mathcal{NL} \subseteq \mathcal{DSPACE}(\log^2 n)$. Proof: By Savitch's Theorem (consider that $\mathcal{NL} = \mathcal{NSPACE}(\log n)$).

Claim 2: $\mathcal{DSPACE}(\log^2 n) \subset \mathcal{PSPACE}$. Proof: Consider $f_1(n) = \log^2 n$ and $f_2(n) = n$. Since $\log^2 n = o(n)$, we can apply the Space Hierarchy Theorem and we have that $\mathcal{DSPACE}(\log^2 n) \subset \mathcal{DSPACE}(n)$. And, since $\mathcal{DSPACE}(n) \subseteq \mathcal{PSPACE}$, we have that $\mathcal{DSPACE}(\log^2 n) \subset \mathcal{PSPACE}$.

Combining both claims, we have that $\mathcal{NL} \subset \mathcal{PSPACE}$.

Exercise 6 Not solved, although I see that there is a proof (which I do not fully understand) in [3].

Exercise 7 Not solved.

References

- [1] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [2] J. E. Hopcroft, R. Motwani, J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 2nd edition, 2001.
- [3] R. Book. Comparing complexity classes, *Journal of Computer and System Sciences* 3(9):213-229, 1974.
- [4] *Time hierarchy theorem*. http://en.wikipedia.org/wiki/Time_hierarchy_theorem
- [5] *Space hierarchy theorem*. http://en.wikipedia.org/wiki/Space_hierarchy_theorem