Exercises on PSPACE and IP
CSCI 6114 Fall 2021

Joshua A. Grochow

November 11, 2021

A language L is in PSPACE if there is a deterministic Turing machine solving L that uses $\text{poly}(|x|)$ space.

1. Show that $\mathsf{PSPACE} \subseteq \mathsf{EXPTIME} = \mathsf{DTIME}(2^{\text{poly}(n)})$. *Hint:* Use the fact that the PSPACE machine must always halt. How many possible configurations does it have?

2. Show that $\mathsf{PSPACE}^{\mathsf{PSPACE}} = \mathsf{PSPACE}$. (Here note that we count the oracle tape in the space usage, so the oracle queries can only be polynomially long.)

3. (a) Show that $\mathsf{NP} \subseteq \mathsf{PSPACE}$.
   (b) Show that $\mathsf{BPP} \subseteq \mathsf{PSPACE}$.
   (c) Show that $\mathsf{PH} \subseteq \mathsf{PSPACE}$. Note that this implies that $\mathsf{AM} \subseteq \mathsf{PSPACE}$.
   (d) Show that $\mathsf{IP} \subseteq \mathsf{PSPACE}$.

4. Show that $\mathsf{IP}[2] = \mathsf{AM}$.

5. Given a Boolean formula φ in CNF form, our goal is to translate it into a polynomial f over the integers $\mathbb{Z}$ or the integer modulo a prime $\mathbb{Z}/p\mathbb{Z}$ such that

$$(\forall \vec{x} \in \{0,1\}^n) \qquad \varphi(\vec{x}) = f(x), \tag{1}$$

   where we think of 0 as false and 1 as true. We will build such an f inductively. First, a Boolean variable x_i turns into an algebraic variable x_i.

   (a) Suppose we have a polynomial f corresponding to a formula φ as above. What polynomial should correspond to the negation $\neg\varphi$? Show your construction satisfies (1) for $\neg\varphi$.
   (b) Suppose we have polynomials f, g corresponding to formulae φ, ψ. What polynomial should correspond to the conjunction $\varphi \wedge \psi$? Show your construction satisfies (1) for $\varphi \wedge \psi$.
   (c) Suppose we have polynomials f, g corresponding to formulae φ, ψ. What polynomial should correspond to the disjunction $\varphi \vee \psi$? Show your construction satisfies (1) for $\varphi \vee \psi$.
   (d) Why does PIT not let us solve UNSAT (thus putting NP into RP)? That is, it seems like we can use the above construction to build f, and then just test whether f is the identically zero. Where does this go wrong?

6. In this exercise our goal is to show that $\mathsf{coNP} \subseteq \mathsf{IP}$ (in fact we'll show that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{IP}$, which by Toda's Theorem already covers all of PH). We'll use the coNP-complete problem k-UNSAT: given a k-CNF, decide whether it is unsatisfiable. Using the construction in the previous exercise, let f_φ denote the polynomial (over the integers) associated to φ.

   (a) Show that the number of satisfying assignments to φ is

   $$n_\varphi = \sum_{\vec{x} \in \{0,1\}^n} f_\varphi(\vec{x}).$$

   (b) Suppose the prover claims that N is the number of satisfying assignments. The prover can send to the verifier the number N, as well as a partially evaluated version of the above function, namely,

   $$P_1(x_1) := \sum_{x_2, x_3, \ldots, x_n \in \{0,1\}} f_\varphi(x_1, x_2, x_3, \ldots, x_n).$$

   This is a univariate polynomial—note that all variables are summed over except that x_1 is left free. What is its degree?

   (c) Verifier then check thats $P_1(0) + P_1(1) = N$. Why is this the right thing to check?

   (d) Verifier then picks a random value r_1 to send to the prover. In the next round, the prover sends back

   $$P_2(x_2) := \sum_{x_3, x_4, \ldots, x_n \in \{0,1\}} f_\varphi(r_1, x_2, x_3, x_4, \ldots, x_n).$$

   Verifier will then check that $P_2(0) + P_2(1) = P_1(r_1)$. Why is this the right thing to check?

   (e) Verifier will then pick a random value r_2 to send to the prover. In the next round, the prover sends back

   $$P_3(x_3) := \sum_{x_4, x_5, \ldots, x_n \in \{0,1\}} f_\varphi(r_1, r_2, x_3, x_4, \ldots, x_n).$$

   And the process continues like this. If the prover gave the wrong value of n to begin with, what is the probability that the verifier accepts at the end of this procedure?

Resources

- Sipser §8.2 and 10.4.
- Moore & Mertens Sections 8.6 and 11.1–11.2
- *Gems of TCS* Chapter 21.
- Arora & Barak Chapters 4 and 8.
- Jonathan Katz's 2011 course, lectures 18–19 contain the proof that $\mathsf{IP} = \mathsf{PSPACE}$.