

CS401 - Problem Set 6

1. The $\leq_p$ symbol tells us something about the *time* required to convert one type of problem to another. We used this symbol in our definition of **PSPACE-HARD**. It seems like it might make more sense to worry about the space used by our conversion procedure, rather than time. In this problem, you'll investigate what would happen if we were to use a space-based reduction definition.

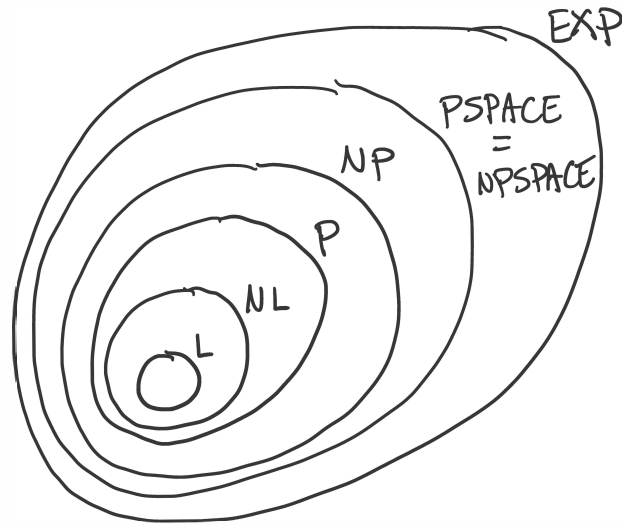
Suppose we instead defined $L' \in \mathbf{PSPACE-HARD}$ if for every $L \in \mathbf{PSPACE}$, $L \leq_{ps} L'$, where $\leq_{ps}$ means there exists a function f computable in polynomial *space* such that $f(x) \in L'$ iff $x \in L$. Consider the following: let $L \in \mathbf{PSPACE}$, and let L^* be any language except for the empty set, or the set of all binary strings. Then there is a string w such that $w \in L^*$ and a string y such that $y \notin L^*$. Consider the function $f_L : L \rightarrow L^*$:

$$f_L(x) = \begin{cases} w & \text{if } x \in L \\ y & \text{if } x \notin L. \end{cases} \quad (1)$$

- (a) Explain why f_L is computable in polynomial space.
 - (b) Explain why this shows it is not a good idea to use polynomial space reductions to define the concept of **PSPACE-HARD**.
 - (c) More generally, why is it good to require the conversion be done using less powerful resources than the resources of the class for which you are defining a HARD version? For example, we used polynomial time conversion to define **NP-HARD**, and where polynomial time is less powerful than **NP**. Then we used polynomial time conversion to define **PSPACE-HARD**, where again polynomial time is less powerful than **PSPACE**.
2. As in class, let $\varphi_i(C_1, C_2) = 1$ (true) if there is a path of length at most 2^i from configuration C_1 to configuration C_2 in the configuration graph of a TM M' , and 0 otherwise. (I've dropped the subscript M' for conciseness.) Explain why

$$\begin{aligned} \varphi_i(C, C') &\equiv \exists C^* : \varphi_{i-1}(C, C^*) \wedge \varphi_{i-1}(C^*, C') \\ &\equiv \exists C^* \forall D, D' ((D = C) \wedge (D' = C^*)) \vee ((D = C^*) \wedge (D' = C')) \rightarrow \varphi_{i-1}(D, D'). \end{aligned} \quad (2)$$

3. *Briefly* explain all of the inclusions in the following diagram:



(There are really only a couple of key ideas that can be used to explain all of the relationships. You don't need to explain why **PSPACE=NPSPACE**.)

4. **Challenge:** For more reduction practice, prove the following problem is NP-Complete. Consider a set of n cards that fit into a box. The cards can be put in facing up or flipped over the long side, as shown in the figure below. There are two columns of m possible holes on each card. The question is whether there is an way to insert the cards so that you can't see the bottom of the box (i.e. every hole in 1 card is blocked by a non-hole in a another card).

