5. Show that if every NP-hard language is also PSPACE-hard, then PSPACE = NP.

   Since NP-hard contains all of the NP-complete problems, every NP-complete language must also be PSPACE-hard. This means SAT must also be PSPACE-hard.

   Since SAT is NP-complete we know that $\forall$ langs A $\in$ NP, there is a polynomial time reduction A $\leq_p$ SAT.

   Since SAT $\in$ NP there is a machine that non-deterministically polynomial time TM M which recognizes SAT.

   Then we can construct a TM N

   N = ”On input w

   1. compute f(w) to polynomial nondeterministic mapping algorithm from A to B
   2. simulate f(w) on M, if f(w) is satisfiable, accept; else reject.”

   Since f is polynomial time computable, |f(w)| is polynomially bounded in |x| and therefore N runs in |x|, and since we have shown that w $\in$ A iff f(w) $\in$ SAT we have shown this is a valid reduction and that A $\in$ NP and thus PSPACE $\subseteq$