CPSC 421 - Homework 10

Tristan Rice, q7w9a, 25886145

1. Exercise 8.4 of Sip.

Show that PSPACE is closed under the operations union, complementation, and star.

Def:

$$PSPACE = \bigcup_k SPACE(n^k)$$

Union

For PSPACE to be closed under union, for any $x, y \in PSPACE, x \cup y \in PSPACE.$

For a language x to be in $PSPACE$ that means a Turing machine can decide it using a deterministic Turing machine using polynomial space.

We can construct a Turing machine that can recognize x or y by making a multitape Turing machine with both the tapes from x and y and having the tape symbols be $\Gamma = \Gamma_x \times \Gamma_y$ and having the cartesian product of states. The machine accepts if either x or y conditions are satisfied.

This machine space usage is bounded by the largest amount of space either x or y uses. Thus, that bound is in PSPACE.

Since a Turing machine can be constructed that decides the union of x, y in polynomial space, PSPACE is closed under union.

You can also show this by constructing a non-deterministic Turing machine that is the union of two deterministic Turing machines since NPSPACE = PSPACE.

Complementation

For PSPACE to be closed under complement, any language A in PSPACE must have it's complement also be in PSPACE.

We can construct a Turing machine B that decides the complement of A.

$$q_{accept,B} = q_{reject,A}$$
$$q_{reject,B} = q_{accept,A}$$

Everything else remains the same.

Since this decides the complement and uses no extra space for all languages in PSPACE, the complement must also be in PSPACE. Thus, PSPACE must be closed under complementation.

Star

For PSPACE to be closed under star, any language A in PSPACE must have $A* \in PSPACE$ as well.

We can construct a non-deterministic Turing machine to compute $A*$. Since NPSPACE = PSPACE, we can do this without any loss of generality.

For every possible splitting of the input w, we can run A on it. If there is a splitting such that all parts are accepted by A, accept, otherwise reject. While this is NP in time, it's still in PSPACE since each invocation of A is in PSPACE. Thus, PSPACE is closed under star.

2. Exercise 8.6 of Sip.

Show that any PSPACE-hard language is also NP-hard.

Def: A language B is PSPACE-complete if it satisfies two conditions:

1. B is in PSPACE, and
2. every A in PSPACE is polynomial time reducible to B.

If B merely satisfies condition 2, we say that it is PSPACE-hard.

Thus, a language B is PSPACE-hard if every A in PSPACE is polynomial reducible to B.

Savitch's theorem: For any function $f : \mathbb{N} \to \mathbb{R}^+$, where $f(n) \geq n, NSPACE(f(n)) \subseteq SPACE(f^2(n))$.

Using Savitch's theorem we know that PSPACE = NPSPACE.

Thus, a language B is PSPACE-hard if every A in NPSPACE is polynomial reducible to B.

It follows that $NP \subseteq NPSPACE$.

Thus, a language B is PSPACE-hard if every A in NP is polynomial reducible to B.

The definition of NP-hard: A language B is NP-hard if every A in NP can be reduced in polynomial time to B.

Thus, we've arrived at the same definition of PSPACE-hard and NP-hard. Thus, any PSPACE-hard language is also NP-hard.