1 Short Review of classes

This entire class has culminated in us being able to identify which time and space complexity classes are included in others. There are 3 containments I will discuss here.

1.1 P, NP, PSPACE

We know that $P \subseteq NP$ and $P \subseteq coNP$, as well as $NP \subseteq PSPACE$ and $coNP \subseteq PSPACE$. However, we do not know if $P = NP$ or $NP = coNP$ or if $NP = PSPACE$ or even if $P = PSPACE$. None of these set containments have been proven to be strict, although most people think that all are strict. Interestingly enough there is the possibility that all are different and there exist languages that are both in NP and coNP, but not in P. Take a look at Fig. 1 for a graphical representation.

1.2 The Major Complexity Classes

As can be seen in Fig. 2, we have the following class containments:

$$L \subseteq NL = coNL \subseteq P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq \text{EXPTIME} \subseteq \text{EXPSPACE}$$

None of these containments are known to be strict. However, utilizing the Time Hierarchy Theorem, we can show that $P \subsetneq \text{EXPTIME}$.

Due to the Space Hierarchy Theorem, we know that $NL \subsetneq PSPACE$. The reason this last one is true is because $NL = NSPACE(\log n) \subsetneq SPACE(\log^2 n) \subsetneq PSPACE$, due to application of Savitch’s Theorem first and then the Space Hierarchy Theorem. Furthermore, $PSPACE \subsetneq EXPSPACE$ by simple application of the Space Hierarchy Theorem.

An interesting observation that arises now is that even though $P = PSPACE$ and $NL = P$ are open problems, because $NL \subsetneq PSPACE$, we know for sure that both cannot be true.
Figure 1: The P, NP, coNP and PSPACE Classes.
Figure 2: The Major Complexity Classes.
1.3 Probabilistic Complexity Classes

We know that $NP \subseteq IP$, because we can view the interaction between the Prover and the Verifier as the Prover giving the Verifier the certificate to check membership in the language which is in NP. Hence, the error rate of this IP algorithm is 0.

Furthermore, $BPP \subseteq IP$, because a $BPP$ algorithm works just like an IP System but ignores the Prover and gets the same error bounds.

No relationship between $NP$ and $BPP$ is known. As you proved in Problem 6 of Pset 6, if $NP \subset BPP$ holds, then $NP = RP$ will also hold.

2 Completeness

So far this semester, we have talked about NL-Completeness (PATH), NP-Completeness (SAT, 3SAT, etc.), PSPACE-Completeness (TQBF, GG) and EXPSPACE-Completeness ($EQ_{REX_1}$). Particular care is needed when dealing with NL-Completeness, because here unlike the rest of the completeness class, log-space reducibility is used and we also cannot compute the entire reduction output, store it, and then feed it into subsequent Turing Machines, because the reduction output might be polynomial in length. Hence, the "compute on the fly" or "compute as you go along" method is needed, where we may need to compute the reduction multiple times to avoid storing the entire reduction output (polynomial space) on the work tape of an NL-machine, where it does not fit.