MPhil Advanced Computer Science
Topics in Logic and Complexity

Lent 2010 Anuj Dawar

Solution Hints for Exercise Sheet 1

1. A configuration of M can be written in space $O(\log|x|)$. The machine R works by enumerating all pairs of configurations of M (on input x) on its work tape and for each such pair (c_1, c_2) determining whether $c_1 \rightarrow_M c_2$ (to do this, we only need to look at the transition table of M, which can be hard coded into R with no additional space requirements).

   This can be used to prove that $\mathsf{Reachability}$ is complete for NL under *logspace* reductions. Indeed, take any language $A \in \mathsf{NL}$ and let M be a nondeterministic machine that recognises A in logarithmic space. The machine R we constructed then defines a reduction from A to the problem of $\mathsf{Reachability}$ since x is accepted by M if, and only if, an accepting configuration is reachable from the starting configuration of (M, x) in the graph produced by R on input x.

   Since $\mathsf{Reachability}$ is complete for NL under *logspace* reductions, it follows that if it is in L, then $\mathsf{L} = \mathsf{NL}$.

2. If $L \in \mathsf{co\text{-}NP}$, then $\bar{L}$—the complement of L—is in NP. So, there is a a nondeterministic Turing machine M' and a polynomial p such that M' halts in time $p(n)$ for all inputs of length n and $\bar{L}$ is exactly the set of strings x such that *some* computation of M' on input x end in an accepting state. Now, define M to be the machine obtained from M' by interchanging all accepting and rejecting states.

3. Suppose L is decided by a strong nondeterministic Turing machine M running in polynomial time. Then, consider the machine M' that is obtained from M by replacing every *maybe* state with a rejecting state. Then, M' is a nondeterministic Turing machine that accepts L in the usual sense. To see this, suppose $x \in L$. Then, at least one computation of M on input x ends in *accept*, so M' accepts x. On the other hand, if $x \notin L$, then all computations of M end in *maybe* or *reject* and so all computations of M' on input x are rejecting. Thus, x is not accepted. This proves that L is in NP.

   Now, consider the machine M'' obtained from M by replacing all *maybe* states by accepting states. Then, we can show that L is exactly the set of strings x for which *every* computation of M'' ends in an accepting state. Thus, by the previous exercise, L is in $\mathsf{co\text{-}NP}$.

4. For the game of Geography, see Theorem 19.3 of Papadimitriou.

   For the game of HEX, see the following paper

   > S. Even and R.E. Tarjan. "A Combinatorial Game which is Complete for Polynomial Space". Journal of the ACM **23**:710–719 (1976).

5. See Theorems 3.2.16, 3.2.17 and 3.2.18 of Grädel et al.

6. See Theorem 3.2.6 of Grädel et al.

7. See Handout 11, page 3.

8. Since the property is in P, and therefore in co-NP, it can certainly be defined in USO. To give an explicit definition, it is perhaps easier to define in ESO the property of having an *odd* number of elements. This can be done by writing a sentence that says: there is a binary relation B, which is a bijection from the universe to itself, which is symmetric (i.e. $B(x, y)$ implies $B(y, x)$) and such that there is exactly one element x such that $B(x, x)$. Then, the negation of this ESO sentence is a USO sentence that defines the structures with an even number of elements.

9. To define explicitly a USO sentence that defines the 2-colourable graphs, it is again easiest to write an ESO sentence that defines the graphs that are *not* 2-colourable. This sentence just asserts that there exists a cycle in the graph of odd size. Note that, if a graph contains a cycle of odd length, then it contains one with no *chords*. That is, there is a cycle $v_1, \ldots, v_n$ where if $i < j$ and $j \neq i + 1$ then there is no edge from v_i to v_j.

   So, to say in ESO that there is a cycle of odd length, it suffices to say that there is a set of vertices C such that every vertex in C has exactly two neighbours in C and that the number of elements in C is odd. For the last, use the construction from the previous exercise.

10. Clearly such sentences exist, by Fagin's theorem. However, writing down such sentences explicitly is not straightforward. For the USO sentence one can use the Kuratowski characterisation of planar graphs that a graph is planar if, and only if, one cannot find five vertices $v_1, \ldots, v_5$ with paths between every pair of them which are mutually disjoint and one cannot find two sets of three vertices u_1, u_2, u_3 and v_1, v_2, v_3 such that there is a path from each u_i to each v_j and these paths are mutually disjoint.

11. Σ_{n+1} can also be characterized as the class of languages L such that there is a *polynomially* balanced relation R, which is in Π^1_n and such that $L = \{x \mid \text{ for some } y, (x, y) \in R\}$ (prove this!). A similar characterisation can be obtained for Π^1_{n+1} in terms of Σ^1_n.

    Now, proceed by induction on n to show the result.

12. This is by an argument analogous to that for Exercise 6 above.