MPhil Advanced Computer Science
Topics in Logic and Complexity

Lent 2010 Anuj Dawar

Exercise Sheet 2

1. Page 11 of Handout 7 contains an illustration of a construction to show that *acyclicity* of graphs is not definable in first-order logic. Write out a proof of this result.

   Prove that *acyclicity* is not definable in $\mathsf{Mon}.\Sigma^1_1$. Is it definable in $\mathsf{Mon}.\Pi^1_1$?

2. Prove (using Hanf's theorem or otherwise) that 3-colourbility of graphs is not definable in first-order logic.

   Graph 3-colourability (and, indeed, 2-colourability) are definable in $\mathsf{Mon}.\Sigma^1_1$. Can you show they are not definable in $\mathsf{Mon}.\Pi^1_1$? Are they definable in *universal second-order logic*?

3. Prove the lemma on page 14 of Handout 8. That is, show that any formula that is positive in the relation symbol R defines a monotone operator.

4. Prove that the formula of LFP given on page 8 of Handout 9 does, indeed, define the greatest fixed point of the operator defined by ϕ.

5. In Handout 9, we say how definitions by simultaneous induction can be replaced by a single application of the **lfp** operator. In this exercise, you are asked to show the same for *nested* applications of the **lfp** operator.

   Suppose $\phi(\mathbf{x}, \mathbf{y}, S, T)$ is a formula in which the relational variables S (of arity s) and T (of arity t) only appear positively, and $\mathbf{x}$ and $\mathbf{y}$ are tuples of variables of length s and t respectively. Show that (for any t-tuple of terms $\mathbf{t}$) the predicate expression

   $$[\mathbf{lfp}_{S,\mathbf{x}}([\mathbf{lfp}_{T,\mathbf{y}}\phi](\mathbf{t}))]$$

   is equivalent to an expression with just one application of **lfp**.

6. On page 18 of Handout 11, the correspondence between k-pebble games and the equivalence $\equiv^k$ is stated. This exercise asks you to prove the easy direction of that equivalence. That is, show that if Duplicator has a winning strategy in the k-pebble game for q moves starting from position $(\mathbb{A}, \mathbf{a})$ and $(\mathbb{B}, \mathbf{b})$, then $(\mathbb{A}, \mathbf{a}) \equiv^k_q (\mathbb{B}, \mathbf{b})$.

7. We say that a graph $G = (V, E)$ contains a *perfect matching* if there is a set $M \subseteq E$ of edges such that for every vertex $v \in V$ there is *exactly one* edge in M that includes v. Prove that the property of having a perfect matching is not definable in LFP (or PFP).

8. Restricted to ordered structures, every query definable in second-order logic (SO) is also definable in PFP. We know this because PFP can express all queries on ordered structures that are in PSPACE while queries

expressible in SO are in the polynomial hierarchy. But, we can prove the inclusion SO $\subseteq$ PFP on ordered structures directly, without bringing in complexity theory. So, suppose $\phi(R)$ is a formula of first-order logic, with a second-order variable R. Prove that there is a formula of PFP that is equivalent (on ordered structures) to the formula $\exists R\phi$. Use this to prove that any formula of SO can be translated to an equivalent (on ordered structures) formula of PFP.

9. Pages 8–9 of Handout 12 describes a method for defining a pre-order that orders the k-variable types. Write down the formula ψ of IFP that defines this order.

10. We showed (page 11 of Handout 10 and page 11 of Handout 11) that on the class of structures in the empty vocabulary, LFP and PFP are no more expressive than first-order logic. The aim of this exercise is to generalise that result.

    (a) Let $\mathcal{E}$ be the class of pure sets, i.e. structures in an empty vocabulary. Prove that for every k, there is a p such that for all $\mathbb{A} \in \mathcal{E}$, $I_k(\mathbb{A})$ has at most p elements.

    (b) Prove that if $\mathcal{C}$ is any class of structures such that for every k, there is a p such that for all $\mathbb{A} \in \mathcal{C}$, $I_k(\mathbb{A})$ has at most p elements, then any formula of PFP is equivalent, on $\mathcal{C}$, to a formula of first-order logic.

    (c) Let $\mathcal{F}$ be the class of *equivalence relations*. That is, it consists of all structures $\mathbb{A} = (A, R)$ where R is a binary relation on A that is reflexive, symmetric and transitive. Prove that for every k, there is a p such that for all $\mathbb{A} \in \mathcal{F}$, $I_k(\mathbb{A})$ has at most p elements.

11. Prove the characterisation on page 18 of Handout 12. That is, show that a query Q is definable in LFP if, and only if, there is a k such that Q is invariant under $\equiv^k$ and the query Q' which takes input $I_k(\mathbb{A})$ and returns $\{[\mathbf{a}] \mid \mathbf{a} \in Q(\mathbb{A})\}$ is computable in polynomial time.

12. A second-order Horn sentence (SO-Horn sentence, for short) is one of the form

$$Q_1 R_1 \ldots Q_p R_p (\forall \mathbf{x} \bigwedge_i C_i)$$

    where, each Q_i is either $\exists$ or $\forall$, each R_i is a relational variable and each C_i is a *Horn* clause, which is defined for our purposes as a disjunction of atomic and negated atomic formulae such that it contains at most one positive occurrence of any relational variable (i.e. any of $R_1, \ldots, R_p$. A sentence is said to be ESO-Horn if it is as above, and all Q_i are $\exists$.

    (a) Show that any ESO-Horn sentence in a relational signature defines a class of structures decidable in polynomial time.

    (b) Show that, if K is an isomorphism-closed class of structures in a relational signature including $<$, such that each structure in K interprets

$<$ as a linear order and

$$\{[\mathcal{A}]_< \mid \mathcal{A} \in K\}$$

is decidable in polynomial time, then there is a ESO-Horn sentence that defines K.

(c) Show that any SO-Horn sentence is equivalent to a ESO-Horn sentence.