Exercises on nominal sets

There are 3-point exercises, 4-point exercises and 5-point exercises. The scoring system is designed so that to get grade $n \in \{3, 4, 5\}$, it is sufficient to solve correctly three n-point exercises. The scoring system is: to get grade $n \in \{3, 4, 5\}$, you need at least $3n$ points, *and* at least $\frac{n}{2}$ points for some n-point exercise.

1. **3 points.** Consider nondeterministic register automata (slide 7). Show that one can decide if an automaton accepts words with arbitrarily large numbers of distinct data values. For instance, the answer should be yes for the automaton which accepts words where all even-numbered positions have the same data value (because the odd-numbered positions can have distinct data values).

2. **3 points.** Show the following pumping lemma. Let $\mathcal{A}$ be nondeterministic register automaton. One can compute a constant M such that if the automaton accepts a word of length at least M, then it also accepts words of arbitrarily large lengths.

3. **3 points.** Find a two-way one-register automaton (slide 13), which recognizes the language "all data values are different".

4. **3 points.** Let A be a finite alphabet. We use formulas of first-order logic to express properties of a data word

   $$(a_1, d_1) \cdots (a_n, d_n) \in (A \times \mathbb{D})^*.$$

   Variables range over positions $\{1, \ldots, n\}$, and the predicates are:

   - $x < y$: position y is after position x;
   - $x \sim y$: positions x and y carry the same data value;
   - $a(x)$: position x carries label $a \in A$.

   For instance, the formula

   $$\forall x \; a(x) \;\Rightarrow\; \exists y \; (b(y) \wedge y > x \wedge x \sim y).$$

   says every position with label a is followed by a position with label b and the same data value. Prove that satisfiability (the question if a given formula formula is true in at least one data word) is undecidable. Hint: use the coding from slide 11.

5. **4 points.** From the logic from the previous example, remove the order predicate $x < y$ (equality of positions $x = y$ is still allowed). Prove that satisfiability becomes decidable.

6. **5 points.** Consider a language $L \subseteq \mathbb{D}^*$, which is closed under arbitrary bijections of $\mathbb{D}$, and also closed under adding letters. Prove that L is recognized by a nondeterministic register automaton.

7. **4 points.** Slide 66 defines a nominal version of deterministic finite automata. Nominal versions can be defined for everything, including pushdown automata and context free grammars, as follows. A *nominal pushdown automaton* is defined like a normal pushdown automaton, except that:

   - The states, as well as the stack and input alphabets are only required to be orbit-finite, and not necessarily finite.
   - The remaining components, namely the initial and accepting subsets, and the transition relation, are required to be equivariant.

   Similarly, a *nominal context-free grammar* is defined like a normal context-free grammar, except that:

   - The nonterminals N and the input alphabet A are only required to be orbit-finite, and not necessarily finite. The initial nonterminal is required to be equivariant, while the set of rules $\Delta \subseteq N \times (N \cup A)^*$ is required to be equivariant and orbit-finite.

   Consider the symmetry where G contains all permutations of the data values, which is the symmetry studied in slides 27-59. Prove that languages recognized by nominal pushdown automata (nondeterministic, accepting under the empty stack) are the same as languages generated by nominal context-free grammars.

8. **5 points.** Consider the symmetry where the data values are rational numbers $\mathbb{Q}$, and the group is the set of translations

   $$G = \{x \mapsto (x + q) : q \in \mathbb{Q}\}.$$

   Prove that there are uncountably many non-isomorphic single-orbit nominal sets. Isomorphism here means equivariant bijection.

9. **4 points.** Consider the symmetry where G contains all permutations of the data values. Prove that for every orbit-finite nominal sets X and Y, for some $n \in \mathbb{N}$ there is an equivariant partial surjective function

   $$f : X^n \to Y$$

10. **5 points.** Consider formulas of first-order logic which talk about a directed graph. The sentences are allowed to use the edge predicate $E(x, y)$ and the equality predicate $x = y$, and quantify over nodes. For example, the sentence

    $$\forall x \exists y E(x, y)$$

says that every node has an outgoing edge. An graph is said to have *decidable first-order theory* if there is an algorithm, which inputs a sentence and says if it is true in the graph. Prove that if a class $\mathcal{G}$ of finite graphs has decidable membership and admits a universal graph (slide 73), then the universal structure has a decidable first-order theory.

11. **5 points.** Consider formulas of *monadic second-order logic*, which are like formulas of first-order logic, but which can also quantify over sets of nodes. Find a decidable class of graphs, which admits a universal graph, but such that the universal graph has an undecidable monadic second-order theory.

12. **3 points.** Consider a class $\mathcal{G}$ of finite graphs which admits a universal graph U (slide 73). Prove that U embeds into the Rado graph (slide 79).

13. **3 points.** Prove that the class of planar graphs does not have amalgamation (slide 75).

14. **4 points.** A *nominal Turing machine* is defined like a normal Turing machine, except that:

    - The states, as well as the work and input alphabets are only required to be orbit-finite, and not necessarily finite.
    - The remaining components, namely the initial and accepting subsets, and the transition relation, are required to be equivariant.

    Prove that for every nondeterministic nominal Turing machine there is a deterministic nominal Turing machine that accepts the same words.

15. **5 points.** We define the running time of a nominal Turing machine the same way as for a normal Turing machine: it is the number of computation steps. This leads to the notion of a nominal Turing machine that runs in polynomial time, with respect to the number of letters in the input. This leads to a nominal version of the P=NP question. Prove that in the symmetry where G contains all bijections, the nominal version of P=NP has the same answer as the standard version.