Exercise Sheet 3

October 13, 2022

Exercise 1. The following formulas (on the signature $\{P, E\}$ with two predicate symbol and $ar(P) = ar(E) = 2$) form the theory of Unbounded Dense Total Orders. The set $\mathbb{R}$ of real numbers is an example of such an order where $P(x, y)$ denotes $x < y$ and $E(x, y)$ denotes $x = y$. For each of the axioms, apply *negation normal form*, *Skolemization* and *prenex normal form* (in this order).

- $\forall x.\ E(x, x)$
- $\forall x, y, z.\ (P(x, y) \land P(y, z)) \to P(x, z)$
- $\forall x, y.\ P(x, y) \to \exists z. P(x, z) \land P(z, y)$
- $\forall x, y.\ E(x, y) \leftrightarrow \neg(P(x, y) \lor P(y, x))$
- $\forall x, y, z.\ (E(x, y) \land E(y, z)) \to E(x, z)$
- $\forall x, y, z.\ (E(x, y) \land P(x, z)) \to P(y, z)$
- $\forall x, y, z.\ (E(x, y) \land P(z, x)) \to P(z, y)$

What does Herbrand's Theorem say about this set of axiom? Can you find an example?

Exercise 2. (Effectively Propositional Logic) Consider the class of formulas of first order logic built on a signature containing only constant symbols (arity 0 functions) and predicate symbols, and of the following form:

$$\forall x_1 ... \forall x_n.\ F(x_1, ..., x_n)$$

where F is quantifier-free.

1. Show that this set of formula is closed under conjunction, disjunction and negation for satisfiability, by which is meant that for arbitrary formulas F_1 and F_2, you can efficiently compute formulas in the above form that are equisatisfiable to $F_1 \land F_2$, $F_1 \lor F_2$ and $\neg F_1$.
2. Show there exists an algorithm to decide the satisfiability of such formulas.

Exercise 3. Recall the definition of weakest precondition:

$$\mathrm{wp}(r, Q) = \{s \mid \forall s'.\ (s, s') \in r \to s' \in Q\}$$

Prove or disprove the following properties:

- $\mathrm{wp}(r_1 \cup r_2, Q) = \mathrm{wp}(r_1, Q) \cup \mathrm{wp}(r_2, Q)$
- $\mathrm{wp}(r_1 \cup r_2, Q) = \mathrm{wp}(r_1, Q) \cap \mathrm{wp}(r_2, Q)$
- $\mathrm{wp}(r, Q_1 \cap Q_2) = \mathrm{wp}(r, Q_1) \cap \mathrm{wp}(r, Q_2)$
- $\mathrm{wp}(r, Q_1 \cup Q_2) = \mathrm{wp}(r, Q_1) \cup \mathrm{wp}(r, Q_2)$

For those that are wrong, do any of them hold if r is restricted to being functional, i.e. if r satisfies

$$\forall x, y_1, y_2.\ ((x, y_1) \in r \land (x, y_2) \in r) \to (y_1 = y_2)$$

Exercise 4. Consider the following program:

```
case class Container(var x: INT, var y: INT):
  def fun: Unit = {
    require(x > 0 && y > 0)
    if x > y then
      x = x+y
      y = x−y
      x = x−y
    else
      y = y−x
  }.ensuring(2*old(this).x + old(this).y > 2*x + y &&
               x > 0 && x > 0)
end Container
```

- Compute R(fun) formally, by expressing all intermediate formulas corresponding to subprograms.
- Write the formula expressing the correctness of the ensuring clause. Is the formula valid when `INT` denotes mathematical integers with their usual operations (`type INT = BigInt` in Scala)?
- Is the the verification condition formula valid for machine integers,

$$Z_{2^{32}} = \{-2^{31}, \ldots, -1, 0, 1 \ldots, 2^{31} - 1\}$$

  and where operations are intepreted as the usual machine arithmetic (`type INT = Int` in Scala)?