Exercises 3

Exercise 1 (Skolemization & Herbrand Universes)**.** 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).

- $\forall x.\ E(x, x)$
- $\forall x, y, z.\ (P(x, y) \wedge P(y, z)) \rightarrow P(x, z)$
- $\forall x, y.\ P(x, y) \rightarrow \exists z. P(x, z) \wedge P(z, y)$
- $\forall x, y.\ E(x, y) \leftrightarrow \neg(P(x, y) \vee P(y, x))$
- $\forall x, y, z.\ (E(x, y) \wedge E(y, z)) \rightarrow E(x, z)$
- $\forall x, y, z.\ (E(x, y) \wedge P(x, z)) \rightarrow P(y, z)$
- $\forall x, y, z.\ (E(x, y) \wedge P(z, x)) \rightarrow P(z, y)$

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 $F_1 \wedge F_2$, $F_1 \vee F_2$ and $\neg F_1$.
2. Show there exists an algorithm to decide the satisfiability of such formulas.

Exercise 3. Consider the ternary propositional operation $\mathsf{ite}(\mathsf{x},\mathsf{y},\mathsf{z})$ (if x then y else z) defined by the following truth table:

x	y	z	$\mathsf{ite}(\mathsf{x},\mathsf{y},\mathsf{z})$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

We consider expressions build only from variables and ite, without constants 0 and 1. We call them pure ite expressions.

1. Express $x \wedge y$ and $x \vee y$ as a pure ite expression;
2. Let e denote an ite expression whose set of free variables is $V = \{x_1, \ldots, x_n\}$. Let v be an assignment assigning all variables in V to 0 (false). What is the truth value of e under v?
3. Give an example of a function $f : \{0,1\}^n \to \{0,1\}$ that is not expressible as a pure ite expression.
4. Which of the following completion of the statement are true?

   The functions $f : \{0,1\}^n \to \{0,1\}$ that can be expressed as pure ite expressions using variables $x_1, \ldots, x_n$ are precisely …

   (a) the functions that are not constant, i.e., not equal to either the function $f_1(x_1, \ldots, x_n) = 1$ nor to the function $f_0(x_1, \ldots, x_n) = 0$

   (b) all functions if $n \geq 3$; functions expressible using only $\wedge, \vee$ in cases $n = 1, 2, 3$

   (c) the functions that could be expressed using an expression with only $\wedge$ and $\vee$

   (d) the functions f such that $f(0, \ldots, 0) = 0$

   (e) the functions f such that $f(0, \ldots, 0) = 0$ and $f(1, \ldots, 1) = 1$

   (f) the functions expressible using only $\neg(x \wedge y)$ as a binary operation

   (g) the functions such that $f(x, \ldots, x) = x$ for every x

   (h) the functions such that $f(x, \ldots, x, f(x, \ldots, x)) = x$ for every x

   For those answers that you chose, explain briefly why they are correct.

Exercise 4 (Resolution with Congruence)**.** Consider a set D with a binary relation $\equiv$ on D and a binary function $\sqcup : D^2 \to D$ which satisfy the following properties:

$$\begin{array}{ll} \mathsf{assoc}: & x \sqcup (y \sqcup z) \equiv (x \sqcup y) \sqcup z \\ \mathsf{transE}: & (x \equiv y) \wedge (y \equiv z) \to (x \equiv z) \\ \mathsf{sym}: & (x \equiv y) \to (y \equiv x) \\ \mathsf{congL}: & (x_1 \equiv x_2) \to (x_1 \sqcup y \equiv x_2 \sqcup y) \\ \mathsf{congR}: & (y_1 \equiv y_2) \to (x \sqcup y_1 \equiv x \sqcup y_2) \end{array}$$

Define another binary relation $\sqsubseteq$ on D by:

$$x \sqsubseteq y \quad \leftrightarrow \quad x \sqcup y \equiv y \tag{DefLE}$$

We claim that it follows that $\sqsubseteq$ is a transitive relation, that is:

$$x \sqsubseteq y \wedge y \sqsubseteq z \;\to\; x \sqsubseteq z \tag{TransLE}$$

In other words, we wish to prove that the following holds in pure first-order logic, where all formulas are considered universally quantified:

$$\{\mathsf{assoc}, \mathsf{transE}, \mathsf{sym}, \mathsf{congL}, \mathsf{congR}, \mathsf{DefLE}\} \models \mathsf{TransLE} \tag{$*$}$$

Note that, when representing the problem in first order logic, $t_1 \equiv t_2$ is just a notation for $E(t_1, t_2)$ where E is some binary predicate symbol, $t_1 \sqsubseteq t_2$ is a notation for $L(t_1, t_2)$ where L is another binary predicate symbol, and $t_1 \sqcup t_2$ is a notation for $f(t_1, t_2)$ where f is some binary function symbol.

Our goal is to use resolution for first-order logic to derive a formal proof of $(*)$ by deriving a contradiction.

A) (2pt) To begin the proof, write down a numbered sequence of *clauses* that you will need in your proof, corresponding to $\mathsf{assoc}, \mathsf{transE}, \ldots$ (whatever the initial set of clauses should be to prove $(*)$ by contradiction).

   1. $\{x \sqcup (y \sqcup z) \equiv (x \sqcup y) \sqcup z\}$ (from assoc)
   2. $\{\neg(x \equiv y), \neg(y \equiv z), (x \equiv z)\}$ (from transE)
   3. …

   You may need more than one clause to encode some of the formulas.

B) (4pt) Continue to write proof steps where each step follows from previous ones. For each step indicate from which previous steps it follows and by which substitution of variables. The last step should be the empty clause $\emptyset$. To save you time in finding the resolution proof, we provide the following fragment of a Stainless file which we used to

check this fact; even if this is not a resolution proof, the invocations of functions in the proof provides hints about the substitutions that you may want to use in your proof.

```
1  def sym(x: D, y: D) = {...}.ensuring( !(x ≡ y) || y ≡ x )
2  ... // define the other axioms
3
4  def transitive(x: D, y: D, z: D) = {
5    require(x ⊑ y)
6    require(y ⊑ z)
7    sym(y ⊔ z, z)
8    congR(x, z, y ⊔ z)
9    assoc(x, y, z)
10   trans(x ⊔ z, x ⊔ (y ⊔ z), (x ⊔ y) ⊔ z)
11   congL(x ⊔ y, y, z)
12   transE(x ⊔ z, (x ⊔ y) ⊔ z, y ⊔ z)
13   transE(x ⊔ z, y ⊔ z, z)
14 }.ensuring(x ⊑ z)
```

Exercise 5 (Logic of Partial Functions)**.** We wish to use first-order logic for our verification task, which requires proving validity of formulas. We use a first-order language (signature) with a relational symbol E, which we would like to represent equality and satisfy the laws of reflexivity, symmetry, and transitivity, as well as congruence laws (analogous to congL and congR in Exercise 4): equal elements are related in the same way by all other relation symbols. We also need a way to represent a *partial* function $\bar{p}(x, y)$ of two arguments: the function can have at most one result, but it can be undefined for certain pairs of elements. Applying $\bar{p}$ when argument is undefined gives undefined result. We will analyze two possibilities for encoding such partial functions in first-order logic, with questions arising in each of them.

Encoding 1. We use a constant b to represent undefined element and a binary function symbol $p(x, y)$ to represent the function $\bar{p}$ (note that we use p to represent the function symbol, whereas $\bar{p}$ represents the function that interprets it). The language of formulas in this part has only symbols $\{E, p, b\}$.

A) (1pt) Write down, as universally quantified first-order logic formulas, the congruence properties of p with respect to E, namely: if in the expression $p(x, y)$ we change x to an E-related element or change y to an E-related element, the new result is E-related to $p(x, y)$.

B) (1pt) Write down, as a universally quantified first-order logic formula, the property that applying p when one of the arguments is undefined results in an undefined value.

C) (1pt) Describe Herbrand universe (the set of ground terms) in this language. Is it finite or infinite?

Encoding 2. We use a ternary relation symbol $P(x, y, z)$ to represent the fact $\bar{p}(x, y) = z$ when all values are defined. To represent that the function is not defined for a given x and y, relation interpreting P would simply not contain any (interpretation of) z such that $P(x, y, z)$ is true. Let a be a constant denoting an arbitrary element of the universe. The language of formulas in this part has only symbols $\{E, P, a\}$.

D) (1pt) Write down, as universally quantified first-order logic formulas, the congruence properties of P with respect to E, namely: if elements x, y, z are related by P, then elements E-related to them are also related by P, as expected by a relation with properties of equality.

E) (1pt) Write down as a universally quantified formula, the property that P should represent a functional relation modulo E: for given pair of elements x, y, the result of $\bar{p}$, if it exists, is unique up to E.

F) (1pt) Consider the result of applying Skolemization of the properties in D) and E). How many new Skolem functions are introduced? Describe the Herbrand universe (the set of ground terms) in this language. Is it finite or infinite?

G) (3pt) Is there a terminating algorithm for checking, given two formulas F_1, F_2 in prenex form with only universal quantifiers using symbols from $\{E, P, a\}$, whether $\mathsf{Ax} \cup \{F_1\} \models F_2$ holds, where Ax are the Skolemized versions of properties introduced in D) and E). If yes, sketch the algorithm. If no, argue why the problem is undecidable.