EPFL Formal Verification Course Exam, October 2022

The aim of this file is to give a solution to the exam questions and a justification of why it is correct. It is not necessarily representative of what was expected from the students.

1 Weakest Precondition (3pt)

Let $r \subseteq S \times S$ and $Q \subseteq S$. Give an expression defining weakest precondition $\mathsf{wp}(r, Q)$ using operations of inverse of a relation, $(\_^{-1})$, set difference $(\setminus)$, and image of a relation under a set, $(\_[\_])$. Prove that your expression is correct by expanding the definitions of wp as well as of relational and set operations.

Solution:

We can classify the points of S into three disjoint sets:

- Q_1 = the points whose image is empty
- Q_2 = the points such that a non empty proper subset of their image is contained in Q
- Q_3 = the points whose image is entirely contained in Q

Intuitevely $\mathsf{wp}(r, Q)$ is the union between Q_1 and Q_3, i.e $S \setminus Q_2$. We know that the preimage of the whole set, S, is $Q_2 \cup Q_3$. We also know that Q_3 is a subset of the preimage of Q. Finally, since a non empty subset of the image of every point in Q_2 is in $S \setminus Q$, we deduce that $Q_2 = r^{-1}[S \setminus Q]$. Therefore $\mathsf{wp}(r, Q) = S \setminus r^{-1}[S \setminus Q]$. Let's see whether this holds formally:

$$\begin{aligned}
S \setminus r^{-1}[S \setminus Q] &= \{s : s \notin r^{-1}[S \setminus Q]\} \\
&= \{s : \neg(\exists s'.(s, s') \in r \land s' \in S \setminus Q)\} \\
&= \{s : \neg(\exists s'.(s, s') \in r \land s' \notin Q)\} \\
&= \{s : \forall s'.(s, s') \notin r \lor s' \in Q\} \\
&= \{s : \forall s'.(s, s') \in r \to s' \in Q\} \\
&= \mathsf{wp}(r, Q)
\end{aligned}$$

2 Iterating a Relation (7pt)

Let $M = (S, I, r, A)$ be a transition system and $\bar{r} = \{(s, s') \mid (s, a, s') \in r\}$, as usual. Let $\Delta = \{(x, x) \mid x \in S\}$.

Let $\bar{r}^k$ denote the usual composition of relation $\bar{r}$ with itself k times.

Define the sequence of relations r_n, for all non-negative integers n, as follows:

- $r_0 = \Delta \cup \bar{r}$
- $r_{n+1} = r_n \circ r_n$

A) (2pt) Prove that $r_n \subseteq r_{n+1}$ for every n.

Solution:

By induction:

Base case:

$$\begin{aligned} r_0 \circ r_0 &= (\Delta \cup \bar{r}) \circ (\Delta \cup \bar{r}) \\ &= (\Delta \circ \Delta) \cup (\Delta \circ \bar{r}) \cup (\bar{r} \circ \Delta) \cup (\bar{r} \circ \bar{r}) \\ &= \Delta \cup \bar{r} \cup \bar{r}^2 \end{aligned}$$

In the previous steps we used the fact that composition distributes over union (cf Exercise Set 2 Part II), and that Δ is a neutral element wrt composition (since $\Delta = \bar{r}^0$).

$$r_0 = \Delta \circ \bar{r} \subseteq \Delta \cup \bar{r} \cup \bar{r}^2 = r_0 \circ r_0 = r_1$$

Induction:

$$r_{n+1} = r_n \circ r_n \subseteq r_{n+1} \circ r_{n+1} = r_{n+2}$$

By IH and the fact that if $r_1 \subseteq r_1'$ and $r_2 \subseteq r_2'$ then $r_1 \circ r_2 \subseteq r_1' \circ r_2'$

B) (3pt) Prove that for every n and every k where $0 \leq k \leq 2^n$ we have $\bar{r}^k \subseteq r_n$.

Solution:

By induction:

Base case:

$$\bar{r}^0 = \Delta \subseteq \Delta \cup \bar{r} = r_0 \quad \bar{r}^1 = \bar{r} \subseteq \Delta \cup \bar{r} = r_0$$

Induction:

If $0 \leq k \leq 2^n$: $\bar{r}^k \subseteq r_n \subseteq r_{n+1}$ by IH and A)

If $2^n \leq k \leq 2^{n+1}$: $\bar{r}^k = r^{2^n} \circ r^{k-2^n} \subseteq r_n \circ r_n = r_{n+1}$ by IH and since $0 \leq k - 2^n \leq 2^n$

C) (2pt) Suppose that S is finite. Find a bound B as a function of $|S|$ such that

$$\mathsf{Reach}(M) \subseteq r_B[I]$$

Aim to find as small bound as possible.

Solution:

Since there are $|S|$ states,

$$\mathsf{Reach}(M) \subseteq \bigcup_{k=0}^{|S|} \bar{r}^k[I] \subseteq r_{\lceil \log |S| \rceil}[I]$$

In fact, $\lceil \log |S| \rceil$ is the smallest n such that $\bar{r}^k[I] \subseteq r_n[I]$ for every k such that $0 \leq k \leq 2^n$

3 Propositional Logic with If (8pt)

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

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

A) (1pt) Express $x \wedge y$ as a pure ite expression;

Solution: $x \wedge y = \mathsf{ite(x,y,x)}$

B) (1pt) Express $x \vee y$ as a pure ite expression;

Solution: $x \vee y = \mathsf{ite(x,x,y)}$

C) (1pt) Let e denote a pure 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?

Solution:

One can show by structural induction that in this case that the truth value of any pure ite expression is 0.

Base case: $x_i \equiv 0$
Induction: Let $\mathsf{ite}(\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3)$ be a pure ite expression where $\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3$ are also pure ite subexpressions. By IH, $\mathsf{p}_1 \equiv \mathsf{p}_2 \equiv \mathsf{p}_3 \equiv \mathsf{0}$. Therefore, $\mathsf{ite}(\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3) \equiv \mathsf{ite}(\mathsf{0}, \mathsf{0}, \mathsf{0}) \equiv \mathsf{0}$

Similarly, when all variables have truth value one, a pure ite expression has also truth value 1. This means that 0 and 1 are not expressible as pure ite expressions.

D) (1pt) Given an example of a function $f : \{0,1\}^n \to \{0,1\}$ that is not expressible as a pure ite expression.

Solution:

$\neg x$ is an example of function that is not expressible as a pure ite expression as it does not fulfill the above property.

E) (4pt) List (in the separate sheet) the answer numbers next to all true completions of the statement:

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 ...

1. ...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 $f_0(x_1, \ldots, x_n) = 0$
2. ...all functions if $n \geq 3$; functions expressible using $\wedge, \vee$ in cases $n = 1, 2, 3$
3. ...the functions that could be expressed using an expression with only $\wedge$ and $\vee$
4. ...the functions f such that $f(0, \ldots, 0) = 0$
5. ...the functions f such that $f(0, \ldots, 0) = 0$ and $f(1, \ldots, 1) = 1$
6. ...the functions expressible using $\neg(x \wedge y)$ as a binary operation
7. ...the functions such that $f(x, \ldots, x) = x$ for every x
8. ...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.

Solution:

1. False: $\neg x$ is not a constant but it is not expressible as a pure ite expression.
2. False: $\neg(x_1 \wedge x_2 \wedge x_3)$ is not expressible as a pure ite expression as it does not fulfill the above property.
3. False: Let's consider a truth table as in the figure above. We prove that any 3 variables expression made with and and or connectors cannot contain at the same time a 1 in the second position (when $x = 0, y = 0, z = 1$) and a 0 in the sixth

position (when $x = 1, y = 0, z = 1$), whereas this is the case in $ite(x, y, z)$ The proof goes by structural induction:

Let P be a 3 variables propositional formula built from variables, and and or connectors.

Base case: The statement is true for any variable since their truth value in the second and sixth row is respectively $(0, 1); (0, 0)$ and (1, 1)

Induction:

If $P_1 \wedge P_2$, then in order to have $(1, 0)$ in the second and sixth position, the possible configurations for $P1$ and $P2$ are (1, 0) and (1, 0); (1, 0) and (1, 1); (1, 1) and (1, 0). In all these cases one of the subexpressions needs to have a 1 in the second position and a 0 in the sixth one, which is not possible by IH.

If $P_1 \vee P_2$, then in order to have $(1, 0)$ in the second and sixth position, the possible configurations for $P1$ and $P2$ are (1, 0) and (1, 0); (1, 0) and (0, 0); (0, 0) and (1, 0). In all these cases one of the subexpressions needs to have a 1 in the second position and a 0 in the sixth one, which is not possible by IH.

4. False: 0 is not expressible as a pure ite expression.
5. True: We first define $\varphi(x_i, ..., x_n) = \begin{cases} \mathsf{ite}(\mathsf{x_n}, \cdot, \cdot) & \text{if } i = n \\ \mathsf{ite}(\mathsf{x_i}, \varphi(\mathsf{x_{i+1}}, ..., \mathsf{x_n}), \varphi(\mathsf{x_{i+1}}, ..., \mathsf{x_n})) & \text{otherwise} \end{cases}$

   Where each placeholder $\cdot$ is a variable. This expression represents a tree where each branch is a row of the truth table of a n variables expression. In every branch except the leftmost and the rightmost one, some of the variables are 1 and others are 0. If we want the corresponding truth table row to have a value of 1 or 0, we just have to choose a variable with that truth value in the placeholder. Therefore, we can represent any function such that $f(0, ..., 0) = 0$ and $f(1, ..., 1) = 1$ with pure ite expressions.
6. False: As seen in Exercise Set 1, every function is expressible with NAND operations which is not the case for pure ite expressions.
7. True: This statement is equivalent to the 5th one.
8. False: $x \to y$ is such a function, but it not expressible as an ite expression.

4 Resolution with Congruence (6pt)

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:

$$x \sqsubseteq y \land y \sqsubseteq z \ \rightarrow \ x \sqsubseteq z \tag{TransLE}$$

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. You can choose to either use notation $\equiv, \sqsubseteq, \sqcup$ or E, L, f, but use one or the other consistently.

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 assoc, transE, ... (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.

Solution:

1. $\{\{x \sqsubseteq y\}, \{y \sqsubseteq z\}, \{\neg(x \sqsubseteq z)\}\}$ (from TransLE)
2. $\{\{\neg(x \sqsubseteq y), x \sqcup y \equiv y\}, \{x \sqsubseteq y, \neg(x \sqcup y \equiv y)\}\}$ (from DefLE)
3. $\{x \sqcup (y \sqcup z) \equiv (x \sqcup y) \sqcup z\}$ (from assoc)
4. $\{\neg(x \equiv y), \neg(y \equiv z), x \equiv z\}$ (from transE)
5. $\{\neg(x \equiv y), y \equiv x\}$ (from sym)
6. $\{\neg(x_1 \equiv x_2), x_1 \sqcup y \equiv x_2 \sqcup y\}$ (from congL)
7. $\{\neg(y_1 \equiv y_2), x \sqcup y_1 \equiv x \sqcup y_2\}$ (from congR)

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.

```
def sym(x: D, y: D) = {...}.ensuring( !(x ≡ y) || y ≡ x )
... // define the other axioms

def transitive(x: D, y: D, z: D) = {
  require(x ⊑ y)
  require(y ⊑ z)
  sym(y ⊔ z, z)
  congR(x, z, y ⊔ z)
  assoc(x, y, z)
  trans(x ⊔ z, x ⊔ (y ⊔ z), (x ⊔ y) ⊔ z)
  congL(x ⊔ y, y, z)
  transE(x ⊔ z, (x ⊔ y) ⊔ z, y ⊔ z)
  transE(x ⊔ z, y ⊔ z, z)
}.ensuring(x ⊑ z)
```

Solution:

8. $\{x \sqsubseteq y\}$ (by 1)

9. $\{x \sqcup y \equiv y\}$ (by 8 and 2 with $x := x;\ y := y$)

10. $\{y \sqsubseteq z\}$ (by 1)

11. $\{y \sqcup z \equiv z\}$ (by 10 and 2 with $x := x;\ y := y$)

12. $\{\neg(y \sqcup z \equiv z), z \equiv y \sqcup z\}$ (by 5 with $x := y \sqcup z;\ y := z$)

13. $\{z \equiv y \sqcup z\}$ (by 11 and 12)

14. $\{\neg(z \equiv y \sqcup z), x \sqcup z \equiv x \sqcup (y \sqcup z)\}$ (by 7 with $x := x;\ y_1 := z;\ y_2 := y \sqcup z$)

15. $\{x \sqcup z \equiv x \sqcup (y \sqcup z)\}$ (by 13 and 14)

16. $\{x \sqcup (y \sqcup z) \equiv (x \sqcup y) \sqcup z\}$ (by 3 with $x := x;\ y := y;\ z := z$)

17. $\{\neg(x \sqcup z \equiv x \sqcup (y \sqcup z)), \neg(x \sqcup (y \sqcup z) \equiv (x \sqcup y) \sqcup z), x \sqcup z \equiv (x \sqcup y) \sqcup z\}$ (by 4 with $x := x \sqcup z;\ y := x \sqcup (y \sqcup z);\ z := (x \sqcup y) \sqcup z$)

18. $\{\neg(x \sqcup (y \sqcup z) \equiv (x \sqcup y) \sqcup z), x \sqcup z \equiv (x \sqcup y) \sqcup z\}$ (by 15 and 17)

19. $\{x \sqcup z \equiv (x \sqcup y) \sqcup z\}$ (by 16 and 18)

20. $\{\neg(x \sqcup y \equiv y), (x \sqcup y) \sqcup z \equiv y \sqcup z\}$ (by 6 with $x_1 := x \sqcup y;\ x_2 := y;\ y := z$)

21. $\{(x \sqcup y) \sqcup z \equiv y \sqcup z\}$ (by 9 and 20)

22. $\{\neg(x \sqcup z \equiv (x \sqcup y) \sqcup z), \neg((x \sqcup y) \sqcup z \equiv y \sqcup z), x \sqcup z \equiv y \sqcup z\}$ (by 4 with $x := x \sqcup z;\ y := (x \sqcup y) \sqcup z;\ z := y \sqcup z$)

23. $\{\neg((x \sqcup y) \sqcup z \equiv y \sqcup z), x \sqcup z \equiv y \sqcup z\}$ (by 15 and 22)

24. $\{x \sqcup z \equiv y \sqcup z\}$ (by 21 and 23)

25. $\{\neg(x \sqcup z \equiv y \sqcup z), \neg(y \sqcup z \equiv z), x \sqcup z \equiv z\}$ (by 4 with $x := x \sqcup z;\ y := y \sqcup z;\ z := z$)

26. $\{\neg(y \sqcup z \equiv z), x \sqcup z \equiv z\}$ (by 25 and 24)

27. $\{x \sqcup z \equiv z\}$ (by 26 and 11)

28. $\{\neg x \sqsubseteq z\}$ (by 1)

29. $\{x \sqcup z \equiv z\}$ (by 28 and 2 with $x := x;\ y := z$)

30. $\{\}$ (by 27 and 29)

5 Approximating Relations (7pt)

Consider a guarded command language whose meanings are binary relations on the set states U.

Let $E(a_1, \ldots, a_n)$ denote an expression built from some atomic relations $a_1, \ldots, a_n$, as well as diagonal relations

$$\Delta_P = \{(x, x) \mid x \in P\}$$

for various sets $P \subseteq U$. The expression E is built from these relations using union (to model non-deterministic choice, relation composition (to represent sequential composition) and transitive closure (to represent loops).

Let us call a relation $s \subseteq U \times U$ an *effect* if it is reflexive (R) and transitive (T).

A) (2pt) Prove that if s is an effect and $a_i \subseteq s$ for all $1 \leq i \leq n$, then

$$E(a_1, \ldots, a_n) \subseteq s$$

Solution:

By structural induction:

Base case:

$a_i \subseteq s$ for all $1 \leq i \leq n$ and since s is reflexive, $\Delta_P \subseteq s$ for every $P \subseteq U$.

Induction:

If $E(a_1, ..., a_n) = E_1(a_1, ..., a_n) \cup E_2(a_1, ..., a_n)$

$$E(a_1, ..., a_n) = E_1(a_1, ..., a_n) \cup E_2(a_1, ..., a_n) \subseteq s \cup s = s \text{ (by IH)}$$

If $E(a_1, ..., a_n) = E_1(a_1, ..., a_n) \circ E_2(a_1, ..., a_n)$: since by IH $E_1(a_1, ..., a_n), E_2(a_1, ..., a_n) \subseteq s$, and s is transitive,

$$E(a_1, ..., a_n) = E_1(a_1, ..., a_n) \circ E_2(a_1, ..., a_n) \subseteq s$$

If $E(a_1, ..., a_n) = E_1(a_1, ..., a_n)^*$

$$E(a_1, ..., a_n) = E_1(a_1, ..., a_n)^* \subseteq s^* = s$$

Where the inclusion is due to IH and the last equality to the fact that s is both transitive and reflexive.

B) (2pt) Let $U = \mathbb{Z}^2$ denote pairs of integers, denoted by integer variables x, y. Let s be a specification relation given by the formula:

$$s = \{((x, y), (x', y')) \mid y \geq 0 \rightarrow (x' \leq x \wedge y' \geq 0)\}$$

Show that s is an effect.

Solution:

Reflexivity:

For arbitrary $(x, y) \in \mathbb{Z}^2$

$$y \geq 0 \rightarrow (x \leq x \wedge y \geq 0) \equiv y \geq 0 \rightarrow y \geq 0 \equiv T$$

Which implies that $((x, y)(x, y)) \in s$ and therefore that s is reflexive.

Transitivity:

Let$((x_1, y_1), (x_2, y_2))$ and $((x_2, y_2), (x_3, y_3)) \in s$. The two following formula holds:

$$y_1 \geq 0 \rightarrow (x_2 \leq x_1 \wedge y_2 \geq 0) \quad y_2 \geq 0 \rightarrow (x_3 \leq x_2 \wedge y_3 \geq 0)$$

Let's show if the formula $y_1 \geq 0 \rightarrow (x_3 \leq x_1 \wedge y_3 \geq 0)$ holds.

Assume $y_1 \leq 0$: by the first statement $x_2 \leq x_1$ and $y_2 \geq 0$ hold. Since $y_2 \geq 0$, by the second statement $x_3 \leq x_2$ and $y_3 \geq 0$ are also true. Therefore $x_3 \leq x_1$ and y_3.

Since s is reflexive and transitive, it is an effect.-

C) (3pt) For the effect s in the previous point, prove that $\rho(p) \subseteq s$ where p is the following program (the initial values of variables can be arbitrary):

```
while (y ≥ 0) {
  x = x − y;
  if (x % 2 == 0)
    y = y / 2
  else
    y = 3*y + 1
}
```

Notation $\rho(p)$ denotes the relation corresponding to the program p.

Solution:

$$\rho(\text{p}) = (\Delta_{y\geq 0} \circ \rho(\text{body}))^* \circ \Delta_{y<0}$$
$$\rho(\text{body}) = \rho(x = x - y) \circ (\Delta_{x\%2==0} \circ \rho(y = y/2) \cup \Delta_{x\%2!=0} \circ \rho(y = 3 * y + 1))$$

$\rho(x = x - y) \subseteq s$: if $y \geq 0$, $y' \geq 0$ (since $y = y'$) and $x' = x - y \leq x$.
$\rho(y = y/2) \subseteq s$: if $y \geq 0$, $y' = y/2 \geq 0$. Moreover since $x' = x$, $x' \leq x$.
$\rho(y = 3 * y + 1) \subseteq s$: if $y \geq 0$, $y' = 3 * y + 1 \geq 0$. Moreover since $x' = x$, $x' \leq x$.

Since all the atomic expressions of the programs are subset of s and since s is an effect, by the result of A) $\rho(\text{p}) \subseteq s$.

6 Logic of Partial Functions (9pt)

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 Problem 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 $p(x, y)$.

Solution: One of the possible answers is

$$\forall x.\forall y.\forall z.\ (E(x, z) \rightarrow E(p(x, y), p(z, y))) \land (E(y, z) \rightarrow E(p(x, y), p(x, z)))$$

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.

Solution: One of the possible answers is

$$\forall x.\forall y.\ E(p(b,y),b) \land E(p(x,b),b)$$

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

Solution:

The set of ground terms of this language is of the form

$$GT_{\mathcal{L}} = \{b, p(b,b), p(p(b,b),b)...\}$$

Formally $GT^0_{\mathcal{L}} = \emptyset \quad GT^{i+1}_{\mathcal{L}} = \{b\} \cup \{p(t_1,t_2) | t_1, t_2 \in GT^i_{\mathcal{L}}\} \quad GT_{\mathcal{L}} = \bigcup_{i=0}^{\infty} GT^i_{\mathcal{L}}$

It is countably 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.

Solution: One of the possible answers is
$\forall x.\forall y.\forall z.\forall w.\ P(x,y,z) \rightarrow$
$((E(x,w) \rightarrow P(w,y,z)) \land (E(y,w) \rightarrow P(x,w,z)) \land (E(z,w) \rightarrow P(x,y,w)))$

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.

Solution: One of the possible answers is

$$\forall x.\forall y.\forall z.\forall w.\ (P(x,y,z) \land P(x,y,w)) \rightarrow E(z,w)$$

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?

Solution:

Since there are no existential quantifiers, applying Skolemization does not introduce new function symbols in the language. Therefore, the set of ground terms in this new language is $\{a\}$

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.