［1］ In this problem, you are asked to answer the questions below regarding set theory and first-order predicate logic. Below, A = {0, 1, 2} and B = {1, 2, 3, 4}.

a. Give numeric values of |A × B| and |B × A|.

|A × B| = 12 elements (13 counting Ø)

|B × A| = 12 elements (13 counting Ø)

b. Give numeric values of |A → B| and |B → A|.

|A → B| = 4^3 = 64 elements (65 counting Ø)

|B → A| = 3^4 = 81 elements (82 counting Ø)

c. Let |C| = n for n ∈ N. Give a numeric expression of |P(C)|, containing n.

We could use combinatorial numbers to calculate the numeric expression of |P(C)|.

$$|P(C)| = 1 + \sum_{i=1}^{n} \binom{n}{i}$$

d. Is $\models_{\rho_{int}} \forall x.\exists y.x < y$? Also, briefly explain the reason for your answer.

True.

For every ‘x’ there is a ‘y’ which is bigger. We could even define ‘y’ as <y = x+1>.

e. Is $\models_{\rho_{int}} \exists y.\forall x.x < y$? Also, briefly explain the reason for your answer.

False

There is not any ‘y’ which is greater than every other ‘x’. We would need a number which is greater than any other. Such a thing does not exist in natural numbers.

［2］ Let c be the following IMP command.

Z := X; (Y := 0; while Z > 0 do (Y := Y + X; Z := Z − 1))

Show the derivation of 〈c, {X 1→ 1, Y 1→ 0, Z 7→ 0}〉 → {X 7→ 1, Y 7→ 1, Z 7→ 0}. Also, describe concisely in plain English the final state obtained by evaluating c from an arbitrary initial state

$$\frac{\dfrac{\langle X,\sigma\rangle \to 1}{\langle Z := X,\sigma\rangle \to \sigma[1/Z]} \quad \dfrac{A}{\langle Y := 0; while\ Z > 0\ do\ (Y := Y + X; Z := Z - 1),\sigma[1/Z]\rangle \to \sigma[1/Y][0/Z]}}{\langle Z := X; \big(\, Y := 0; while\ Z > 0\ do\ (Y := Y + X; Z := Z - 1)\big), \{X \to 1, Y \to 0, Z \to 0\ \}\rangle \to \{X \to 1, Y \to 1, Z \to 0\}}$$

$$\text{A} = \frac{\dfrac{\langle 0,\sigma[1/Z]\rangle \to 0}{\langle Y := 0,\sigma[1/Z]\rangle \to \sigma[0/Y][1/Z]} \quad \dfrac{B}{\langle while\ Z > 0\ do\ (Y := Y + X; Z := Z - 1),\sigma[0/Y][1/Z]\rangle \to \sigma[1/Y][0/Z]}}{\langle Y := 0; while\ Z > 0\ do\ (Y := Y + X; Z := Z - 1),\sigma[1/Z]\rangle \to \sigma[1/Y][0/Z]}$$

$$\text{B} = \frac{\dfrac{\langle Z,\sigma[0/Y][1/Z]\rangle \to 1 \quad 1 > 0}{\langle Z > 0,\sigma[0/Y][1/Z]\rangle \to true} \quad \dfrac{C}{\langle Y := Y + X; Z = Z - 1,\sigma[0/Y][1/Z]\rangle \to \sigma[1/Y][0/Z]} \quad \dfrac{D}{\langle while\ Z > 0\ do\ (Y := Y + X; Z := Z - 1),\sigma[1/Y][0/Z]\rangle \to \sigma[1/Y][0/Z]}}{\langle while\ Z > 0\ do\ (Y := Y + X; Z := Z - 1),\sigma[0/Y][1/Z]\rangle \to \sigma[1/Y][0/Z]}$$

$$\text{C} = \frac{\dfrac{\langle Y,\sigma[0/Y][1/Z]\rangle \to 0 \quad \langle ,\sigma[0/Y][1/Z]\rangle \to 1 \quad 1 = 0 + 1}{\langle Y := Y + X,\sigma[0/Y][1/Z]\rangle \to \sigma[1/Y][1/Z]} \quad \dfrac{\langle Z,\sigma[1/Y][1/Z]\rangle \to 1 \quad \langle 1,\sigma[1/Y][1/Z]\rangle \to 1 \quad 0 = 1 - 1}{\langle Z := Z - 1,\sigma[1/Y][1/Z]\rangle \to \sigma[1/Y][0/Z]}}{\langle Y := Y + X; Z = Z - 1,\sigma[0/Y][1/Z]\rangle \to \sigma[1/Y][0/Z]}$$

$$\text{D} = \frac{\dfrac{\langle Z,\sigma[1/Y][0/Z]\rangle \to 0 \quad \langle 0,\sigma[1/Y][0/Z]\rangle \to 0 \quad 0 \not> 0}{\langle Z > 0,\sigma[1/Y][0/Z]\rangle \to false}}{\langle while\ Z > 0\ do\ (Y := Y + X; Z := Z - 1),\sigma[1/Y][0/Z]\rangle \to \sigma[1/Y][0/Z]}$$

As we can see, the first two commands assign values to the Z and Y variables. Therefore, any initial state that they have will not affect the final state. The only variable that will change the flow of control is the variable X. This variable will keep its value at the end of the program. This program will execute the loop 'X' times. In each loop, it will add the variable 'X' to the variable 'Y'.

With this information we can conclude that for an arbitrary initial state the final state will be:

Initial state: $\sigma = \{X \to n, Y \to n', Z \to n''\}$

Final state: $\sigma' = \{X \to n, Y \to n * n\ , Z \to 0\}$

$$\dfrac{\dfrac{\langle X,\sigma\rangle \to 1}{\langle Z := X,\sigma\rangle \to \sigma[1/Z]} \quad \dfrac{\dfrac{\langle 0,\sigma[1/Z]\rangle \to 0}{\langle Y := 0,\sigma[1/Z]\rangle \to \sigma[0/Y][1/Z]} \quad \dfrac{\dfrac{\dfrac{\langle Z,\sigma[0/Y][1/Z]\rangle \to 1 \quad 1 > 0}{\langle Z > 0,\sigma[0/Y][1/Z]\rangle \to true} \quad \dfrac{\dfrac{\langle Y,\sigma[0/Y][1/Z]\rangle \to 0 \quad \langle ,\sigma[0/Y][1/Z]\rangle \to 1 \quad 1 = 0 + 1}{\langle Y := Y + X,\sigma[0/Y][1/Z]\rangle \to \sigma[1/Y][1/Z]} \quad \dfrac{\langle Z,\sigma[1/Y][1/Z]\rangle \to 1 \quad \langle 1,\sigma[1/Y][1/Z]\rangle \to 1 \quad 0 = 1 - 1}{\langle Z := Z - 1,\sigma[1/Y][1/Z]\rangle \to \sigma[1/Y][0/Z]}}{\langle Y := Y + X; Z = Z - 1,\sigma[0/Y][1/Z]\rangle \to \sigma[1/Y][0/Z]} \quad \dfrac{\dfrac{\langle Z,\sigma[1/Y][0/Z]\rangle \to 0 \quad \langle 0,\sigma[1/Y][0/Z]\rangle \to 0 \quad 0 \not> 0}{\langle Z > 0,\sigma[1/Y][0/Z]\rangle \to false}}{\langle while\ Z > 0\ do\ (Y := Y + X; Z := Z - 1),\sigma[1/Y][0/Z]\rangle \to \sigma[1/Y][0/Z]}}{\langle while\ Z > 0\ do\ (Y := Y + X; Z := Z - 1),\sigma[0/Y][1/Z]\rangle \to \sigma[1/Y][0/Z]}}{\langle Y := 0; while\ Z > 0\ do\ (Y := Y + X; Z := Z - 1),\sigma[1/Z]\rangle \to \sigma[1/Y][0/Z]}}{\langle Z := X; \big(Y := 0; while\ Z > 0\ do\ (Y := Y + X; Z := Z - 1)\big),\{X \to 1, Y \to 0, Z \to 0\}\rangle \to \{X \to 1, Y \to 1, Z \to 0\}}$$

［3］ In the class, we have discussed that there is no direct syntax for equality of arithmetic expressions, that is, "$a_0 = a_1$". Show derived Bexp evaluation rules for $a_0 = a_1$. Then, argue that the derived evaluation rules are correct by showing that $\langle a_0 = a_1, \sigma_i \rangle \rightarrow$ true (via the evaluation relation extended with your derived rules) if and only if $\langle a_0 \leq a_1 \wedge a_1 \leq a_0, \sigma_i \rangle \rightarrow$ true, and likewise for the case false is returned.

Derived rules $a_0 = a_1$

$$\frac{\langle a_0, \sigma\rangle \rightarrow n_0 \quad \langle a_1, \sigma\rangle \rightarrow n_1 \quad n_0 = n_1}{\langle a_0 = a_1, \sigma\rangle \rightarrow b}$$

Proof:

$$\langle a_0 = a_1, \sigma\rangle \rightarrow b \iff \langle a_0 \leq a_1 \wedge a_1 \leq a_0, \sigma\rangle \rightarrow b$$

Let's analyze the different cases.

-Assume $a_0 = a_1$

$$\langle a_0 = a_1, \sigma\rangle \rightarrow true \iff \langle a_0 \leq a_1 \wedge a_1 \leq a_0, \sigma\rangle \rightarrow true$$

$$\frac{\langle a_0, \sigma\rangle \rightarrow n_0 \quad \langle a_1, \sigma\rangle \rightarrow n_1 \quad n_0 = n_1}{\langle a_0 = a_1, \sigma\rangle \rightarrow true}$$

$$\frac{\dfrac{\langle a_0, \sigma\rangle \rightarrow n_0 \quad \langle a_1, \sigma\rangle \rightarrow n_1 \quad n_0 \leq n_1}{\langle a_0 \leq a_1, \sigma\rangle \rightarrow true} \quad \dfrac{\langle a_1, \sigma\rangle \rightarrow n_1 \quad \langle a_0, \sigma\rangle \rightarrow n_0 \quad n_1 \leq n_0}{\langle a_1 \leq a_0, \sigma\rangle \rightarrow true} \quad true \wedge true}{\langle a_0 \leq a_1 \wedge a_1 \leq a_0, \sigma\rangle \rightarrow true}$$

Assume $a_0 < a_1$

$$\langle a_0 = a_1, \sigma\rangle \rightarrow false \iff \langle a_0 \leq a_1 \wedge a_1 \leq a_0, \sigma\rangle \rightarrow false$$

$$\frac{\langle a_0, \sigma\rangle \rightarrow n_0 \quad \langle a_1, \sigma\rangle \rightarrow n_1 \quad n_0 = n_1}{\langle a_0 = a_1, \sigma\rangle \rightarrow false}$$

$$\frac{\dfrac{\langle a_0, \sigma\rangle \rightarrow n_0 \quad \langle a_1, \sigma\rangle \rightarrow n_1 \quad n_0 \leq n_1}{\langle a_0 \leq a_1, \sigma\rangle \rightarrow true} \quad \dfrac{\langle a_1, \sigma\rangle \rightarrow n_1 \quad \langle a_0, \sigma\rangle \rightarrow n_0 \quad n_1 \leq n_0}{\langle a_1 \leq a_0, \sigma\rangle \rightarrow false} \quad true \wedge false}{\langle a_0 \leq a_1 \wedge a_1 \leq a_0, \sigma\rangle \rightarrow false}$$

-Assume $a_0 > a_1$

$$\langle a_0 = a_1, \sigma\rangle \rightarrow false \iff \langle a_0 \leq a_1 \wedge a_1 \leq a_0, \sigma\rangle \rightarrow false$$

$$\frac{\langle a_0, \sigma\rangle \rightarrow n_0 \quad \langle a_1, \sigma\rangle \rightarrow n_1 \quad n_0 = n_1}{\langle a_0 = a_1, \sigma\rangle \rightarrow false}$$

$$\frac{\dfrac{\langle a_0, \sigma\rangle \rightarrow n_0 \quad \langle a_1, \sigma\rangle \rightarrow n_1 \quad n_0 \leq n_1}{\langle a_0 \leq a_1, \sigma\rangle \rightarrow false} \quad \dfrac{\langle a_1, \sigma\rangle \rightarrow n_1 \quad \langle a_0, \sigma\rangle \rightarrow n_0 \quad n_1 \leq n_0}{\langle a_1 \leq a_0, \sigma\rangle \rightarrow true} \quad false \wedge true}{\langle a_0 \leq a_1 \wedge a_1 \leq a_0, \sigma\rangle \rightarrow false}$$

［4］ In this problem, you are asked to prove that the sequence operation of IMP commands is associative. Prove the following: (c_0; (c_1; c_2), σ_i)→ σ ′ if and only if ((c_0; c_1); c_2, σ_i) → σ ′ .

To prove that the sequence operation of Imp commands is associative we are going to elaborate the derivation tree of both operations. After that we would be able to compare the results of both.

$$\frac{\langle c_0, \sigma\rangle \rightarrow \sigma_1 \quad \dfrac{\langle c_1, \sigma_1\rangle \rightarrow \sigma_2 \quad \langle c_2, \sigma_2\rangle \rightarrow \sigma'}{\langle c_1;\ c_2,\ \sigma_1\rangle \rightarrow \sigma'}}{\langle c_0;\ (c_1;\ c_2), \sigma\rangle \rightarrow \sigma'}$$

$$\frac{\dfrac{\langle c_0, \sigma\rangle \rightarrow \sigma_1 \quad \langle c_1, \sigma_1\rangle \rightarrow \sigma_2}{\langle c_0;\ c_1,\ \sigma\rangle \rightarrow \sigma_2} \quad \langle c_2, \sigma_2\rangle \rightarrow \sigma'}{\langle c_0;\ (c_1;\ c_2), \sigma\rangle \rightarrow \sigma'}$$

As we can see in both trees, we end up with the same three leaves that are executed in the same order:

$$\langle c_0, \sigma\rangle \rightarrow \sigma_1$$

$$\langle c_1, \sigma_1\rangle \rightarrow \sigma_2$$

$$\langle c_2, \sigma_2\rangle \rightarrow \sigma'$$

So we can conclude (c0; (c1; c2), σ_i)→ σ ′ if and only if ((c0; c1); c2, σ_i) → σ ′.

［5］ In this problem, you are asked to prove that the evaluation relation of Bexp is deterministic. Prove the following by structural induction on the syntax of b: if $\langle b, \sigma_i \rangle \rightarrow b_0$ and $\langle b, \sigma_i \rangle \rightarrow b_1$ then $b_0 = b_1$.

Let $\phi(b) = \forall \sigma, b_1, b_2 . \langle b,\sigma \rangle \rightarrow b_1 \wedge \langle b,\sigma \rangle \rightarrow b_2 \Rightarrow b_1 = b_2$

Pick $b \in$ Bexp, σ, b_1, b_2 and $a \in$ Aexp arbitrarily. We show that $\langle b,\sigma \rangle \rightarrow b_1 \wedge \langle b,\sigma \rangle \rightarrow b_2 \Rightarrow b_1 = b_2$ by structural induction on b. Suppose that $\langle b,\sigma \rangle \rightarrow b_1 \wedge \langle b,\sigma \rangle \rightarrow b_2$. We will show that $b_1 = b_2$ by case analysis on b.

- Case b = true
  - $b_1 = b_2$ = true by the rule for evaluating booleans.
- Case b = false
  - $b_1 = b_2$ = false by the rule for evaluating booleans.
- Case b= $(a_0 \leq a_1)$

  By the rules to evaluate comparison, we have:
  - $\langle a_0,\sigma \rangle \rightarrow n_0$, $\langle a_1,\sigma \rangle \rightarrow n_1$ and $b_1 = (n_0 \leq n_1)$
  - $\langle a_0,\sigma \rangle \rightarrow n_0'$, $\langle a_1,\sigma \rangle \rightarrow n_1'$ and $b_2 = (n_0' \leq n_1')$

  We have $\phi(a_0)$ and $\phi(a_1)$ by induction hypothesis. Therefore, $n_0 = n_0'$, and $n_1 = n_1'$. Hence, $b_1 = (n_0 \leq n_1) = (n_0' \leq n_1') = b_2$.
- Case b = $\neg b'$
  - $b_1 = b_2 = \sigma(\neg b')$ by the rules for evaluating negation.
- Case b = $(b_3 \wedge b_4)$

  By the rules to evaluate AND operations, we have:
  - $\langle b_5,\sigma \rangle \rightarrow b_3$, $\langle b_6,\sigma \rangle \rightarrow b_4$ and $b_1 = (b_3 \wedge b_4)$
  - $\langle b_5,\sigma \rangle \rightarrow b_3'$, $\langle b_6,\sigma \rangle \rightarrow b_4'$ and $b_2 = (b_3' \wedge b_4')$

  We have $\phi(b_0)$ and $\phi(b_1)$ by induction hypothesis. Therefore, $b_3 = b_3'$, and $b_4 = b_4'$. Hence, $b_1 = (b_3 \wedge b_4) = (b_3' \wedge b_4') = b_2$.

［6］ All the examples of well-founded relations that we have seen in the class satisfied the property that there are at most finitely many elements that are smaller than an element. That is, the well-founded relations ($\prec$, D) had the property that for every element $d \in D$, $\{d' \in D \mid d' \prec d\}$ is a finite set. Do all well-founded relations satisfy this property? If yes, briefly argue why that is the case. If no, give a counterexample that disproves the claim

The definition of well-founded relation states: “Let A be some set. A binary relation $\prec \subseteq AxA$ is said to be well-founded if there is no infinite sequence of elements $a_0, a_1, a_2, ...$ of A such that $a_{i+1} \prec a_i$ for each $i \in \mathbb{N}$”. Therefore, we can already see that every well-founded relation satisfy the property that there are at most finitely many elements that are smaller than an element.

However we can also define a well-founded relation as follows: “A binary relation $\prec \subseteq AxA$ is well-founded iff for any $Q \subseteq A$ such that $Q \neq \emptyset$, there is an element $m \in Q$ such that $\forall b \prec m.b \notin Q$. (I.e., any non-empty subset of A has a minimal element.)”.

Proof:
“If”: Let $a_0, a_1, a_2, ...$ be an infinite descending chain. Then, the set $Q = \{a_i \mid i \in \mathbb{N}\}$ is non-empty and has no minimal element.
“Only if”: We are going to construct a chain of elements as follows. Take a_0 to be any element of Q. We assume $a_n \prec \ldots \prec a_0$ has been constructed. Either there is some $b \prec a_n$ such that $b \in Q$ or there is not. If not, stop the construction. In other case, take $a_{n+1} = b$. As $\prec$ is well-founded the chain $\ldots \prec a_i \prec \ldots \prec a_1 \prec a_0$ cannot be infinite. It is finite, of the form $a_n \prec \ldots \prec a_0$ with $\forall b \prec a_n \notin Q$. Take the required minimal element m to be a_n.