Homework 4
CS 474

Alexandra Levinshteyn
(Discussed with Rachel Dickinson and Aarthy Padmanaban)

December 10th, 2024

All code for this homework is in the following Google Colab: `https://colab.research.google.com/drive/1yIX6efw4goXPCpIrL9VENI0Ob0IiZpso?usp=sharing`.

It can also be found in the following GitHub file: `https://github.com/AlexandraLevinshteyn/Alexandra-Levinshteyn-CS-474/blob/main/CS%20474%20Homework%204/CS%20474%20Homework%204%20Code.ipynb`.

Problem 1

Task 1

I will follow Example 8.12 in the notes. We have the following group axioms:

1. Associativity: $\forall x, y, z. f(f(x, y), z) = f(z, f(y, z))$
2. Identity $\forall x. f(x, e) = x \wedge f(e, x) = x$
3. Inverse: $\forall x \exists y. f(x, y) = e \wedge f(y, x) = e$

We note that the first two axioms are already in the desired prenex rectified normal form and universal. However, the third axiom isn't yet in this form. We have to skolemize it by replacing y with a new function $g(x)$, which effectively returns the inverse of x:

$$\forall x. f(x, g(x)) = e \wedge f(g(x), x) = e.$$

We want to prove the following formula φ:

$$\varphi : \forall e'.((\forall x. f(x, e') = x \wedge f(e', x) = x) \implies (e = e'))).$$

We need to bring this formula into prenex form:

$$\begin{aligned}\varphi &\equiv \forall e'.(\neg(\forall x. f(x, e') = x \wedge f(e', x) = x) \vee (e - e'))) \\ &\equiv \forall e'.((\exists x.(\neg(f(x, e') = x) \vee \neg(f(e', x) = x)) \vee (e = e'))) \\ &\equiv \forall e'.\exists x.(\neg(f(x, e') = x) \vee \neg(fe', x = x) \vee 9e = e')).\end{aligned}$$

We then negate φ to find that

$$\neg\varphi \equiv \exists e'.\forall x.(f(x, e') = x \wedge f(e', x) = x \wedge \neg(e = e')).$$

We proceed to skolemize the above by replacing the quantified variable e' by a new constant c:

$$\forall x.(f(x, c) = x) \wedge f(c, x) = x \wedge \neg(e = c)).$$

This results in a set X containing four universal formulas:

1. $\forall x, y, z. f(f(x, y), z) = f(x, f(y, z))$ (Eq. 1)
2. $\forall x. f(x, e) = x \wedge f(e, x) = x$ (Eq. 2)
3. $\forall x. f(x, g(x)) = e \wedge f(g(x), x) = e$ (Eq. 3)
4. $\forall x.(f(x, c) = x \wedge f(c, x) = x \wedge \neg(e = c))$ (Eq. 4)

We want to check if these four formulas are simultaneously satisfiable.

In order to do this, we want to instantiate with the depth 0 ground terms by constants e and c. We want the formulas where all quantified variables are replaced by all possible combinations of e and c. We end up with the following formulas:

1. Eq. (1) with $x = e$, $y = e$, $z = e$: $f(f(e, e), e) = f(e, f(e, e))$
2. Eq. (1) with $x = e$, $y = e$, $z = c$: $f(f(e, e), c) = f(e, f(e, c))$
3. Eq. (1) with $x = e$, $y = c$, $z = e$: $f(f(e, c), e) = f(e, f(c, e))$
4. Eq. (1) with $x = e$, $y = c$, $z = c$: $f(f(e, c), c) = f(e, f(c, c))$
5. Eq. (1) with $x = c$, $y = e$, $z = e$: $f(f(c, e), e) = f(c, f(e, e))$
6. Eq. (1) with $x = c$, $y = e$, $z = c$: $f(f(c, e), c) = f(c, f(e, c))$
7. Eq. (1) with $x = c$, $y = c$, $z = e$: $f(f(c, c), e) = f(c, f(c, e))$
8. Eq. (1) with $x = c$, $y = c$, $z = c$: $f(f(c, c), c) = f(c, f(c, c))$
9. Eq. (2) with $x = e$: $f(e, e) = e \wedge f(e, e) = e$
10. Eq. (2) with $x = c$: $f(c, e) = c \wedge f(e, c) = c$
11. Eq. (3) with $x = e$: $f(e, g(e)) = e \wedge f(g(e), e) = e$
12. Eq. (3) with $x = c$: $f(c, g(c)) = e \wedge f(g(c), c) = e$
13. Eq. (4) with $x = e$: $f(e, c) = e \wedge f(c, e) = e \wedge \neg(e = c)$
14. Eq. (4) with $x = c$: $f(c, c) = c \wedge f(c, c) = c \wedge \neg(e = c)$

We can then proceed to put these formulas into Z3. We have the following results.

```
[2] # Problem 1 Task 1

    # Defining constants and functions
    e = Const('e', IntSort())
    c = Const('c', IntSort())
    f = Function('f', IntSort(), IntSort(), IntSort())
    g = Function('g', IntSort(), IntSort())

    # Defining all 14 formulas
    eqs = []

    for x in [e, c]:
      for y in [e, c]:
        for z in [e, c]:
          eq = f(f(x, y), z) == f(x, f(y, z))
          eqs.append(eq)

    for x in [e, c]:
      eq = And(f(x, e) == x, f(e, x) == x)
      eqs.append(eq)

    for x in [e, c]:
      eq = And(f(x, g(x)) == e, f(g(x), x) == e)
      eqs.append(eq)

    for x in [e, c]:
      eq = And(f(x, c) == x, f(c, x) == x, Not(e == c))
      eqs.append(eq)

    # Checking if all 14 formulas together are satisfiable
    s = Solver()
    s.add(eqs)

    result = s.check()

    if result == unsat:
        print("Under group axioms, there is no other identity.")
    else:
        print("Under group axioms, there may or may not be another identity.")
```

```
Under group axioms, there is no other identity.
```

Task 2

I will follow Example 8.13 in the notes but use a different form for the statement we are trying to prove. We can take the three group axioms brought into the correct form from Task 1.

We want to prove the following formula φ, which essentially states that if y and z both act as inverses to x, then $y = z$:

$$\varphi : \forall x, y, z.(f(x, y) = e \wedge f(y, x) = e \wedge f(x, z) = e \wedge f(z, x) = e) \implies (y = z).$$

We can write this formula as

$$\begin{aligned}\varphi &\equiv \forall x, y, z.\neg(f(x, y) = e \wedge f(y, x) = e \wedge f(x, z) = e \wedge f(z, x) = e) \vee (y = z)\\ &\equiv \forall x, y, z.\neg(f(x, y) = e) \vee \neg(f(y, x) = e) \vee \neg(f(x, z) = e) \vee \neg(f(z, x) = e) \vee (y = g(x))\\ &\equiv \neg\exists x, y, z.(f(x, y) = e) \wedge (f(y, x) = e) \wedge (f(x, z) = e) \wedge (f(z, x) = e) \wedge \neg(y = z).\end{aligned}$$

Negating the above gives us that

$$\neg\varphi \equiv \exists x, y, z.(f(x, y) = e) \wedge (f(y, x) = e) \wedge (f(x, z) = e) \wedge (f(z, x) = e) \wedge \neg(y = z).$$

Skolemizing using three new constant symbols b, c, and d gives us

$$(f(b, c) = e) \wedge (f(c, b) = e) \wedge (f(b, d) = e) \wedge (f(d, b) = e) \wedge \neg(c = d).$$

This results in a set X containing three universal formulas and one simple formula:

1. $\forall x, y, z.f(f(x, y), z) = f(x, f(y, z))$ (Eq. 1)
2. $\forall x.f(x, e) = x \wedge f(e, x) = x$ (Eq. 2)
3. $\forall x.f(x, g(x)) = e \wedge f(g(x), x) = e$ (Eq. 3)
4. $(f(b, c) = e) \wedge (f(c, b) = e) \wedge (f(b, d) = e) \wedge (f(d, b) = e) \wedge \neg(c = d)$ (Eq. 4)

We want to check if these four formulas are simultaneously satisfiable.

In order to do this, we want to instantiate with the depth 0 ground terms by constants e, b, c, and d. We want the formulas where all quantified variables are replaced by all possible combinations of e, b, c, and d. We end up with 64 formulas from Eq. 1, 4 formulas from Eq. 2, 4 formulas from Eq. 3, and 1 formula from Eq. 4. I will not be writing all of them out because 73 formulas total is a lot. You can see how they are defined in the code on the next page.

We can then proceed to put these formulas into Z3. We have the following results.

```
[5]  # Problem 1 Task 2

     # Defining constants and functions
     e = Const('e', IntSort())
     b = Const('b', IntSort())
     c = Const('c', IntSort())
     d = Const('d', IntSort())
     f = Function('f', IntSort(), IntSort(), IntSort())
     g = Function('g', IntSort(), IntSort())

     # Defining all formulas
     eqs = []

     for x in [e, b, c, d]:
       for y in [e, b, c, d]:
         for z in [e, b, c, d]:
           eq = f(f(x, y), z) == f(x, f(y, z))
           eqs.append(eq)

     for x in [e, b, c, d]:
       eq = And(f(x, e) == x, f(e, x) == x)
       eqs.append(eq)

     for x in [e, b, c, d]:
       eq = And(f(x, g(x)) == e, f(g(x), x) == e)
       eqs.append(eq)

     eq = And(f(b, c) == e, f(c, b) == e, f(b, d) == e, f(d, b) == e, Not(c == d))
     eqs.append(eq)

     # Checking if all formulas together are satisfiable
     s = Solver()
     s.add(eqs)

     result = s.check()

     if result == unsat:
         print("Under group axioms, every element has only one inverse.")
     else:
         print("Under group axioms, this may or may not be the case.")
```

```
Under group axioms, every element has only one inverse.
```

Problem 2

Part (a)

Let $T_{\mathbb{N}}$ refer to the theory of natural numbers with $+$ and T_f refer to the theory of uninterpreted functions.

We start off with the following $\Sigma_f \cup \Sigma_{\mathbb{N}}$ formula φ:

$$\varphi \equiv y \leq x \wedge x \leq y \wedge f(y) = f(7) \wedge x \leq 5.$$

We follow the Nelson-Oppen method from the textbook. Note that

$$\begin{aligned} x &\leq y \\ y &\leq x \\ x &\leq 5 \end{aligned}$$

are already $T_{\mathbb{N}}$ literals. Let's replace $f(y) = f(7)$ with $f(y) = f(w)$ and $w = 7$. Then, we have the $\Sigma_{\mathbb{N}}$ formula

$$\varphi_{\mathbb{N}} : x \leq y \wedge y \leq x \wedge x \leq 5 \wedge w = 7$$

and the Σ_f formula

$$\varphi_f : f(y) = f(w).$$

We note that $\varphi_{\mathbb{N}}$ and φ_f share the variables y and w. Furthermore, $\varphi_{\mathbb{N}} \wedge \varphi_f$ is $(T_{\mathbb{N}} \cup T_f)$-equisatisfiable to φ.

Part (b)

We have the $\Sigma_{\mathbb{N}}$ formula

$$\varphi_{\mathbb{N}} : x \leq y \wedge y \leq x \wedge x \leq 5 \wedge w = 7$$

and the Σ_f formula

$$\varphi_f : f(y) = f(w)$$

with

$$V = \text{shared}(\varphi_{\mathbb{N}}, \varphi_f) = \{y, w\}.$$

There are 2 equivalence relations to consider, which we list by stating the partitions:

1. $\{\{y, w\}\}$, i.e., $y = w$: $\varphi_{\mathbb{N}} \wedge \alpha(V, E)$ is $T_{\mathbb{N}}$-unsatisfiable since it cannot be the case that both $y = w$ and $y \leq x \wedge x \leq 5 \wedge w = 7$ (as $y \leq 5$ and $y = 7$ are contradictions).

2. $\{\{y\}, \{w\}\}$, i.e., $y \neq w$: $\varphi_{\mathbb{N}} \wedge \alpha(V, E)$ is $T_{\mathbb{N}}$-satisfiable ($x = y = 4$ and $w = 7$ for example) and $\varphi_f \wedge \alpha(V, E)$ is T_f-satisfiable ($y = 4$, $w = 7$, and $f(4) = f(7) = 1$ for example).

From the second partition, we find that φ is $(T_{\mathbb{N}} \cup T_f)$-satisfiable.

Now, let's find a satisfying model. First, note that $\varphi_{\mathbb{N}}$ forces $x = y$. We also need to satisfy $x \leq 5$. So, we can choose $x = y = 4$. We are forced to choose $w = 7$. Note that this also satisfies $y \neq w$. Also, φ_f forces $f(y) = f(w)$. We can choose to set $f(v) = 1$ for all $v \in \mathbb{N}$, so that $f(y) = f(w) = 1$. In the original formula, we have no need for w. This results in the following satisfying model for φ:

1. $x = 4$

2. $y = 4$

3. $f(v) = 1$ for all $v \in \mathbb{N}$

Problem 3

Part (a)

Consider any fixpoint F of f. The result of $f(S)$ will always include 2, so any fixpoint F must include 2. Other than that, we simply double each element in F. So, if F includes 2, $f(F)$ will include 4, so F must include 4. Then, $f(F)$ will include 8, so F must include 8, and so on.

In general, F must include every power of 2, starting with 2. We prove this inductively. The base case is done above: F contains 2. The inductive hypothesis assumes that F must contain every power of 2 from 2^1 to 2^i. The inductive step then shows that F must contain 2^{i+1} since F containing 2^i will force $f(F)$ to contain 2^{i+1}, meaning that we want F to contain 2^{i+1} for it to be a fixpoint. This completes the proof.

Define the set

$$G = \{2^i | i \geq 1, i \in \mathbb{N}\}$$

of elements that F must contain. Then, any fixpoint F of f must follow this property:

$$G \subseteq F.$$

We note that G itself is a fixpoint of f:

$$\begin{aligned}
f(G) &= \{2\} \cup \{y | y = 2x, x \in G\} \\
&= \{2\} \cup \{y | y = 2x, x = 2^i, i \geq 1, i \in \mathbb{N}\} \\
&= \{2\} \cup \{2 \cdot 2^i | i \geq 1, i \in \mathbb{N}\} \\
&= \{2\} \cup \{2^i | i \geq 2, i \in \mathbb{N}\} \\
&= \{2^i | i \geq 1, i \in \mathbb{N}\} \\
&= G.
\end{aligned}$$

Thus, the least fixpoint of f is

$$\mathbf{lfp}(f) = G = \{2^i | i \geq 1, i \in \mathbb{N}\}.$$

Part (b)

Assume that we have two sets A and B such that $A \subseteq B$. Then, we consider

$$f(A) = \{100\} \cup \{y \in X | \exists z \in \mathbb{N}, z \text{ is prime}, yz \in A\}$$

and

$$f(B) = \{100\} \cup \{y \in X | \exists z \in \mathbb{N}, z \text{ is prime}, yz \in B\}.$$

Consider any element $a \in f(A)$. We have multiple cases. The first case is that $a = 100$. In this case, we know that $a \in f(B)$. The other case is that there exists some prime $z' \in \mathbb{N}$ such that $az' \in A$. Set this prime z' in stone. Since $A \subseteq B$, we know that $az' \in B$. Thus, there exists a prime $z' \in \mathbb{N}$ such that $az' \in B$. In other words, $a \in f(B)$. So, for any element $a \in f(A)$, we can show that $a \in f(B)$. Therefore, we have that

$$A \subseteq B \implies f(A) \subseteq f(B),$$

meaning that f is monotonic.

Let us now consider the following sequence:

$$T^0 = \emptyset, T^{i+1} = f(T^i).$$

We have the following:

$$
\begin{aligned}
T^0 &= \emptyset \\
T^1 &= f(\emptyset) \\
&= \{100\} \cup \{y \in X | \exists z \in \mathbb{N}, z \text{ is prime}, yz \in \emptyset\} \\
&= \{100\} \\
T^2 &= f(\{100\}) \\
&= \{100\} \cup \{y \in X | \exists z \in \mathbb{N}, z \text{ is prime}, yz \in \{100\}\} \\
&= \{100\} \cup \{50, 20\} \\
&= \{20, 50, 100\} \\
T^3 &= f(\{20, 50, 100\}) \\
&= \{100\} \cup \{y \in X | \exists z \in \mathbb{N}, z \text{ is prime}, yz \in \{20, 50, 100\}\} \\
&= \{100\} \cup \{10, 4, 25, 10, 50, 20\} \\
&= \{4, 10, 20, 25, 50, 100\} \\
T^4 &= f(\{4, 10, 20, 25, 50, 100\}) \\
&= \{100\} \cup \{y \in X | \exists z \in \mathbb{N}, z \text{ is prime}, yz \in \{4, 10, 20, 25, 50, 100\}\} \\
&= \{100\} \cup \{2, 5, 2, 10, 4, 5, 25, 10, 50, 20\} \\
&= \{2, 4, 5, 10, 20, 25, 50, 100\} \\
T^5 &= f(\{2, 4, 5, 10, 20, 25, 50, 100\}) \\
&= \{100\} \cup \{y \in X | \exists z \in \mathbb{N}, z \text{ is prime}, yz \in \{2, 4, 5, 10, 20, 25, 50, 100\}\} \\
&= \{100\} \cup \{1, 2, 1, 5, 2, 10, 4, 5, 25, 10, 50, 20\} \\
&= \{1, 2, 4, 5, 10, 20, 25, 50, 100\}. \\
T^6 &= f(\{1, 2, 4, 5, 10, 20, 25, 50, 100\}) \\
&= \{100\} \cup \{y \in X | \exists z \in \mathbb{N}, z \text{ is prime}, yz \in \{1, 2, 4, 5, 10, 20, 25, 50, 100\}\} \\
&= \{100\} \cup \{1, 2, 1, 5, 2, 10, 4, 5, 25, 10, 50, 20\} \\
&= \{1, 2, 4, 5, 10, 20, 25, 50, 100\}.
\end{aligned}
$$

At this point, we notice that $T^5 = T^6$ and so $T^i = T^5$ for all $i \geq 6$. By Tarski-Knaster, we have that the least fixpoint of f is

$$\mathbf{lfp}(f) = T^\infty = \bigcup_i T^i = \{1, 2, 4, 5, 10, 20, 25, 50, 100\}.$$

Part (c)

Assume that we have two sets A and B such that $A \subseteq B$. Then, we consider

$$f(A) = A \cup \{0\} \backslash \{1\}$$

and

$$f(B) = B \cup \{0\} \backslash \{1\}.$$

Consider any element $a \in f(A)$. We have multiple cases. The first case is that $a = 0$. In this case, we know that $a \in f(B)$. The other case is that $a \in A$. Since we remove 1 always, we can assume that $a \neq 1$. Also, as $A \subseteq B$, $a \in B$. As $a \in B$ and $a \neq 1$, $a \in f(B)$. So, for any element $a \in f(A)$, we can show that $a \in f(B)$. Therefore, we have that

$$A \subseteq B \implies f(A) \subseteq f(B),$$

meaning that f is monotonic.

Let us now consider the following sequence:

$$T^0 = \emptyset, T^{i+1} = f(T^i).$$

We have the following:

$$\begin{aligned}
T^0 &= \emptyset \\
T^1 &= f(\emptyset) \\
&= \emptyset \cup \{0\} \backslash \{1\} \\
&= \{0\} \\
T^2 &= f(\{0\}) \\
&= \{0\} \cup \{0\} \backslash \{1\} \\
&= \{0\}.
\end{aligned}$$

At this point, we notice that $T^2 = T^1$ and so $T^i = T^1$ for all $i \geq 2$. By Tarski-Knaster, we have that the least fixpoint of f is

$$\mathbf{lfp}(f) = T^\infty = \bigcup_i T^i = \{0\}.$$

Part (d)

Consider the following sets:

$$A = \emptyset$$
$$B = \{1\}.$$

We have that

$$\begin{aligned}
f(A) &= \emptyset \cup \{1\} \cup \{2x | x \in \emptyset\} \backslash \{x | x \in \emptyset, x \text{ is odd}\} \\
&= \emptyset \cup \{1\} \cup \emptyset \backslash \emptyset \\
&= \{1\} \\
f(B) &= \{1\} \cup \{1\} \cup \{2x | x \in \{1\}\} \backslash \{x | x \in \{1\}, x \text{ is odd}\} \\
&= \{1\} \cup \{1\} \cup \{2\} \backslash \{1\} \\
&= \{2\}.
\end{aligned}$$

So, we clearly have that $A \subseteq B$ but not that $f(A) \neq f(B)$. Thus, f is not monotonic.