Exercise 1 (An Application of Congruence Closure): (2 + 3 + 3 = 8 points)

Consider the following code fragment of an imperative program:

```
a = c;
d = f[f[c]];
f[b] = f[c];
if ( b == f[a] ) {
   (*)
}
```

The fragment has been translated to the following set of term equalities $\mathcal{E}$.

$$\begin{aligned} a &\equiv c \\ d &\equiv f(f(c)) \\ f(b) &\equiv f(c) \\ b &\equiv f(a) \end{aligned}$$

a) Show via $\leftrightarrow_{\mathcal{E}}^*$ that $d \equiv_{\mathcal{E}} f(c)$ holds.

b) Show via congruence closure that $d \equiv_{\mathcal{E}} f(c)$ holds.

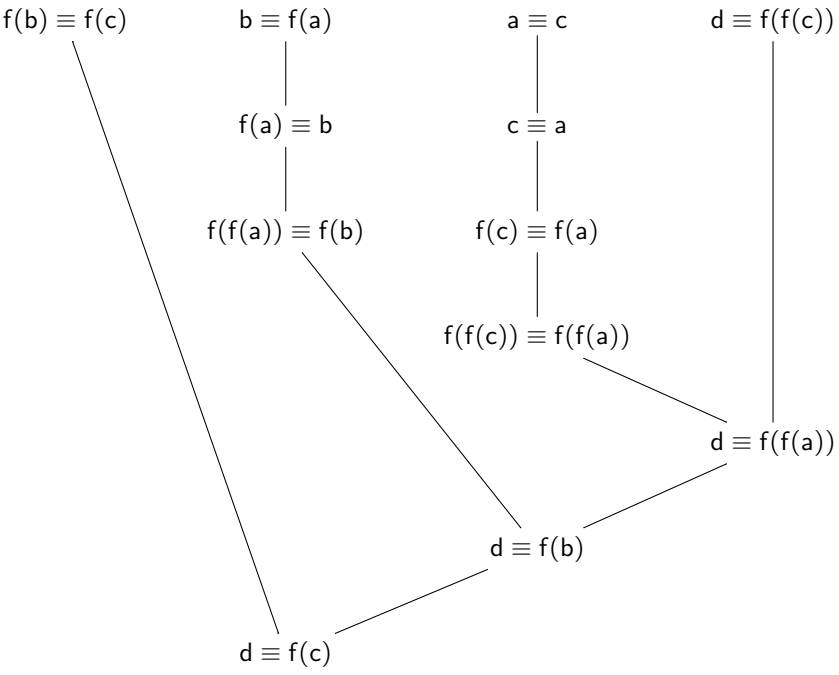
c) Give initial values for the variables `a, b, c, d` and for the array `f`, such that at the position `(*)` the value of `d` is not equal to that of `f[c]`. What is the problem?

Solution:

a)

$$\underline{d} \rightarrow_{\mathcal{E}} f(f(\underline{c})) \leftarrow_{\mathcal{E}} f(\underline{f(a)}) \leftarrow_{\mathcal{E}} \underline{f(b)} \rightarrow_{\mathcal{E}} f(c)$$

b)



c)

Array f				a	b	c	d	line of code	intermediate steps
0	1	2	3						
1	2	3	4	0	2	1	0	(Start)	
1	2	3	4	$\boxed{1}$	2	$\boxed{1}$	0	a = c	
1	2	3	4	1	2	1	$\boxed{3}$	d = f[f[c]]	2 = f[c], 3 = f[2]
1	2	$\boxed{2}$	4	1	2	1	3	f[b] = f[c]	2 = f[c]
1	2	2	4	1	2	1	3	if(b == f[a]){	
1	2	2	4	1	2	1	3	(*)	3 = d, 2 = f[c], $\boxed{\boxed{\texttt{f[c]} \neq \texttt{d}}}$

The problem is the destructive update in the line `f[b] = f[c]` . After its evaluation, the equality `d = f[f[c]]` may not hold any more. Hence, the simple translation of the code fragment was incorrect, since side effects were not taken into account.

Exercise 2 (The Algorithm CONGRUENCE_CLOSURE): (4 points)

Consider the set of term equalities $\mathcal{E}$ consisting of the following ground identities:

$$\begin{aligned} a &\equiv b \\ c &\equiv f(c) \\ f(b) &\equiv g(b) \\ d &\equiv c \\ f(g(b)) &\equiv f(d) \end{aligned}$$

Decide $g(c) \equiv_{\mathcal{E}} f(f(a))$ using the Algorithm CONGRUENCE_CLOSURE from the lecture. Give the set S and as intermediate results also the sets L during each iteration of Step 4. You can omit singleton sets in L.

Solution:

$S = \{a, b, c, d, f(a), f(b), f(c), f(d), f(f(a)), f(g(b)), g(b), g(c)\}$.
L_i is the set L prior to step 4 in i^{th} iteration, K_{i+1} is the corresponding set K directly after step 4. In each L_i and K_i, singleton sets are omitted (indicated by ...).

- $L_0 = \{\{a, b\}, \{c, d, f(c)\}, \{f(b), g(b)\}, \{f(g(b)), f(d)\}\} \cup \ldots$
- $K_1 = L_0 \cup \{\{f(a), f(b)\}, \{f(c), f(d)\}\} \cup \ldots$
- $L_1 = \{\{a, b\}, \{c, d, f(c), f(d), f(g(b))\}, \{f(a), f(b), g(b)\}, \} \cup \ldots$
- $K_2 = L_1 \cup \{\{f(a), f(b)\}, \{f(c), f(d)\}, \{f(f(a)), f(g(b))\}\} \cup \ldots$
- $L_2 = \{\{a, b\}, \{c, d, f(c), f(d), f(g(b)), f(f(a))\}, \{f(a), f(b), g(b)\}\} \cup \ldots$
- $K_3 = L_2 \cup \{\{f(a), f(b)\}, \{f(c), f(d)\}, \{f(f(a)), f(g(b))\}\} \cup \ldots$
- $L_3 = L_2$

Since $g(c)$ and $f(f(a))$ are contained in different elements of L_3, we have $g(c) \not\equiv_{\mathcal{E}} f(f(a))$.

Exercise 3 (Congruence Closure for Satisfiability): (6 + 2 + 1 = 9 points)

The goal of this exercise is to develop a decision procedure for the decidability of propositional logic with equalities and uninterpreted functions. Given a signature Σ and a set of variables $\mathcal{V}$, the set of all formulas in this logic $\mathcal{F}$ is the smallest set such that:

- if $s, t \in \mathcal{T}(\Sigma, \mathcal{V})$, then $s \equiv t \in \mathcal{F}$
- if $\varphi, \psi \in \mathcal{F}$, then $\varphi \wedge \psi \in \mathcal{F}$ and $\varphi \vee \psi \in \mathcal{F}$
- if $\varphi \in \mathcal{F}$, then $\neg\varphi \in \mathcal{F}$

So we have, e.g., $\neg(\neg(x_1 \equiv x_2) \vee \neg(x_2 \equiv x_3)) \wedge x_4 \equiv x_5 \wedge \neg(\mathsf{f}(x_1) \equiv \mathsf{f}(x_2)) \wedge \neg(x_5 \equiv x_1) \in \mathcal{F}$.
Given a formula $\varphi \in \mathcal{F}$, our goal is to prove or disprove that there is an interpretation $I = (\mathcal{A}, \alpha, \beta)$ such that $I \models \varphi$. Here, the relation $\models$ is defined as follows:

$$\begin{array}{rcl} I \models s \equiv t & \iff & I(s) = I(t) \\ I \models \varphi \wedge \psi & \iff & I \models \varphi \text{ and } I \models \psi \\ I \models \varphi \vee \psi & \iff & I \models \varphi \text{ or } I \models \psi \\ I \models \neg\varphi & \iff & I \not\models \varphi \end{array}$$

a) Use the congruence closure procedure to develop a decision procedure for propositional logic with equalities and uninterpreted functions. So given $\varphi \in \mathcal{F}$, the procedure has to decide whether there is an interpretation $I = (\mathcal{A}, \alpha, \beta)$ with $I \models \varphi$.

Hints:

- Look for a way to reduce satisfiability of $\varphi \in \mathcal{F}$ to satisfiability of $\varphi' \in \mathcal{F}$ where φ' does not contain variables.
- You can use the following lemma for ground terms s_i, t_i, u_j, v_j with $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$:
  If there exist algebras $A_1, \ldots, A_m$ with $A_i \models u_1 \equiv v_1 \wedge \cdots \wedge u_n \equiv v_n \wedge \neg s_i \equiv t_i$ $(i \in \{1, \ldots, m\})$, then there exists also an algebra A with $A \models u_1 \equiv v_1 \wedge \cdots \wedge u_n \equiv v_n \wedge \neg s_1 \equiv t_1 \wedge \cdots \wedge \neg s_m \equiv t_m$ (and vice versa).

b) Use your algorithm from **a)** to prove or disprove that that the formula $\neg(\neg(x_1 \equiv x_2) \vee \neg(x_2 \equiv x_3)) \wedge x_4 \equiv x_5 \wedge \neg(\mathsf{f}(x_1) \equiv \mathsf{f}(x_2)) \wedge \neg(x_5 \equiv x_1)$ is satisfiable.

c) Give a formula $\varphi \in \mathcal{F}$ and a *finite* carrier $\mathcal{A}$ such that φ is satisfiable for all infinite carriers, but there is no interpretation $I = (\mathcal{A}, \alpha, \beta)$ such that $I \models \varphi$.

Solution:

a)
a) given a formula φ, construct φ' by replacing each variable x in φ with a fresh constant c_x
b) let $\varphi_1 \vee \cdots \vee \varphi_n$ be the DNF of φ'
c) for each $i \in \{1, \ldots, n\}$
   i. let $\varphi_i = \varphi_i^+ \wedge \varphi_i^-$ with $\varphi_i^+ = s_1 \equiv t_1 \wedge \cdots \wedge s_m \equiv t_m$ and $\varphi_i^- = \neg(q_1 \equiv p_1) \wedge \cdots \wedge \neg(q_\ell \equiv p_\ell)$
   ii. let $\mathcal{E} = \{s_1 \equiv t_1, \ldots, s_m \equiv t_m\}$
   iii. check whether there is a $j \in \{1, \ldots, \ell\}$ with $q_j \equiv_\mathcal{E} p_j$ by congruence closure
   iv. if this is not the case, return $\top$
d) return $\bot$

b)
- Input: $\neg(\neg(x_1 \equiv x_2) \vee \neg(x_2 \equiv x_3)) \wedge x_4 \equiv x_5 \wedge \neg(\mathsf{f}(x_1) \equiv \mathsf{f}(x_2)) \wedge \neg(x_5 \equiv x_1)$

- a) (replacing variables with constants) yields $\neg(\neg(c_{x_1} \equiv c_{x_2}) \vee \neg(c_{x_2} \equiv c_{x_3})) \wedge c_{x_4} \equiv c_{x_5} \wedge \neg(\mathsf{f}(c_{x_1}) \equiv \mathsf{f}(c_{x_2})) \wedge \neg(c_{x_5} \equiv c_{x_1})$
- b) (conversion to DNF) yields: $c_{x_1} \equiv c_{x_2} \wedge c_{x_2} \equiv c_{x_3} \wedge c_{x_4} \equiv c_{x_5} \wedge \neg(\mathsf{f}(c_{x_1}) \equiv \mathsf{f}(c_{x_2})) \wedge \neg(c_{x_5} \equiv c_{x_1})$
- c i.) yields $\varphi_1^+ = c_{x_1} \equiv c_{x_2} \wedge c_{x_2} \equiv c_{x_3} \wedge c_{x_4} \equiv c_{x_5}$ and $\varphi_1^- = \neg(\mathsf{f}(c_{x_1}) \equiv \mathsf{f}(c_{x_2})) \wedge \neg(c_{x_5} \equiv c_{x_1})$
- c ii.) yields $\mathcal{E} = \{c_{x_1} \equiv c_{x_2}, c_{x_2} \equiv c_{x_3}, c_{x_4} \equiv c_{x_5}\}$
- c iii.) yields $\top$, since we have $\mathsf{f}(c_{x_1}) \equiv_{\mathcal{E}} \mathsf{f}(c_{x_2})$
- since there are no other conjunctive clauses left, d) returns $\bot$

c) Let $\mathcal{A} = \{\mathcal{O}\}$ and $\varphi = \neg(x \equiv y)$. For each infinite carrier, φ is clearly satisfiable by assigning two different values to x and y. However, there is no interpretation $I = (\mathcal{A}, \alpha, \beta)$ such that $I \models \varphi$.