Mathematical and Logical Foundations of Computer Science
Solutions

Second Class Test 2021/22

Mathematical and Logical Foundations of Computer Science

Question 1 [Relations, functions, induction & linear equations]

(a) Let $V = \{1, 2, 3, 4, 5, 6, 7\}$ and let $E = \{(1, 7), (2, 1), (4, 1), (6, 5), (6, 6)\}$ be a binary relation on V. Find the equivalence closure of this relation and state the equivalence classes. (It may help to draw a diagram.) **[4 marks]**

(b) The coefficients of the following system are taken from GF(2). Solve it using Gaussian elimination.

$$\begin{aligned} x_1 + x_2 + x_4 &= 0 \\ x_1 + x_3 + x_4 &= 1 \\ x_2 + x_5 &= 0 \\ x_1 + x_2 + x_3 + x_5 &= 0 \\ x_1 + x_3 &= 0 \end{aligned}$$

[5 marks]

(c) Let S be the smallest subset of $\{\mathtt{a}, \mathtt{b}\}^*$ such that all of the following conditions are satisfied:

- $\varepsilon \in S$.
- If $w \in S$, then $\mathtt{aab}w \in S$.
- If $w \in S$, then any anagram of w is also in S. (An anagram of w is a string that arises from w by a permutation of the letters.)

(i) Show that `aabababaa` is in S. **[2 marks]**

(ii) Argue that `ababab` is not in S by giving a property and proving that all elements of S satisfy this property. **[5 marks]**

(d) Consider the following Java methods. Do they represent functions? If yes, are they injective, surjective, or bijective? Justify your answers.

```
int doubleInt(int number) { return number * 2; }
float addOneToFloat(float number) { return number + 1.0; }
```

[4 marks]

Model answer / LOs / Creativity:

(a) The equivalence closure results from taking the reflexive, symmetric, and transitive closures (in this order). It is equal to the following set:

$$\{(1,1),(2,2),(3,3),(4,4),(5,5),(6,6),(7,7),(1,2),(2,1),(1,4),(4,1),\\ (1,7),(7,1),(2,4),(4,2),(2,7),(7,2),(4,7),(7,4),(5,6),(6,5)\}$$

The equivalence classes are $\{1,2,4,7\}$, $\{3\}$, and $\{5,6\}$.

(b) Gaussian elimination in compressed form should look somewhat like the following.

$$\left(\begin{array}{ccccc|c} 1&1&0&1&0&0\\ 1&0&1&1&0&1\\ 0&1&0&0&1&0\\ 1&1&1&0&1&0\\ 1&0&1&0&0&0 \end{array}\right) \to \left(\begin{array}{ccccc|c} 1&1&0&1&0&0\\ 0&1&1&0&0&1\\ 0&1&0&0&1&0\\ 0&0&1&1&1&0\\ 0&1&1&1&0&0 \end{array}\right) \quad \text{add row 1 to rows 2,4,5}$$

$$\to \left(\begin{array}{ccccc|c} 1&1&0&1&0&0\\ 0&1&1&0&0&1\\ 0&0&1&0&1&1\\ 0&0&1&1&1&0\\ 0&0&0&1&0&1 \end{array}\right) \quad \text{add row 2 to rows 3,5}$$

$$\to \left(\begin{array}{ccccc|c} 1&1&0&1&0&0\\ 0&1&1&0&0&1\\ 0&0&1&0&1&1\\ 0&0&0&1&0&1\\ 0&0&0&1&0&1 \end{array}\right) \quad \text{row 4} \leftarrow \text{row 3} + \text{row 4}$$

$$\to \left(\begin{array}{ccccc|c} 1&1&0&1&0&0\\ 0&1&1&0&0&1\\ 0&0&1&0&1&1\\ 0&0&0&1&0&1\\ 0&0&0&0&0&0 \end{array}\right) \quad \text{row 5} \leftarrow \text{row 5} + \text{row 4}$$

$$\to \left(\begin{array}{ccccc|c} 1&1&0&1&0&0\\ 0&1&0&0&1&0\\ 0&0&1&0&1&1\\ 0&0&0&1&0&1\\ 0&0&0&0&0&0 \end{array}\right) \quad \text{row 2} \leftarrow \text{row 2} + \text{row 3}$$

$$\to \left(\begin{array}{ccccc|c} 1&0&0&1&1&0\\ 0&1&0&0&1&0\\ 0&0&1&0&1&1\\ 0&0&0&1&0&1\\ 0&0&0&0&0&0 \end{array}\right) \quad \text{row 1} \leftarrow \text{row 1} + \text{row 2}$$

$$\rightarrow \left(\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \quad \text{row 1} \leftarrow \text{row 1} + \text{row 4}$$

This shows x_5 is free. As GF(2) only has two elements (0 and 1), the solutions are

$$x_5 = 0, \quad x_4 = 1, \quad x_3 = 1, \quad x_2 = 0, \quad x_1 = 1$$

and

$$x_5 = 1, \quad x_4 = 1, \quad x_3 = 0, \quad x_2 = 1, \quad x_1 = 0.$$

(c) (i) The string `aababababaa` can be produced as

$$\varepsilon \Rightarrow \texttt{aab} \Rightarrow \texttt{aabaab} \Rightarrow \texttt{ababaa} \Rightarrow \texttt{aabababaa}$$

with the 'anagram' rule used in the second-to-last step, or as

$$\varepsilon \Rightarrow \texttt{aab} \Rightarrow \texttt{aabaab} \Rightarrow \texttt{aabaabaab} \Rightarrow \texttt{aabababaa}$$

with the 'anagram' rule used in the final step. (There are also other possibilities, which involve more steps in total.)

(ii) The property is that every string in S contains twice as many `a`s as `b`s. Formally, let $\#_{\texttt{a}}(w)$ be the number of `a`s in the string w and let $\#_{\texttt{b}}(w)$ be the number of `b`s, then the property is

$$\#_{\texttt{a}}(w) = 2\#_{\texttt{b}}(w) \text{ for all } w \in S.$$

We can prove this by structural induction:

base case: For ε, we have $\#_{\texttt{a}}(\varepsilon) = 0 = 2\#_{\texttt{b}}(\varepsilon)$.

inductive case 1: Suppose $\#_{\texttt{a}}(w) = 2\#_{\texttt{b}}(w)$. Then

$$\#_{\texttt{a}}(\texttt{aab}w) = 2 + \#_{\texttt{a}}(w) = 2 + 2\#_{\texttt{b}}(w) = 2(1 + \#_{\texttt{b}}(w)) = 2\#_{\texttt{b}}(\texttt{aab}w).$$

inductive case 2: Suppose $\#_{\texttt{a}}(w) = 2\#_{\texttt{b}}(w)$. A permutation of the letters does not affect the total number of any type of letter, so if w' is an anagram of w, then $\#_{\texttt{a}}(w') = \#_{\texttt{a}}(w) = 2\#_{\texttt{b}}(w) = 2\#_{\texttt{b}}(w')$.

This completes the proof that all elements of S satisfy the property.

Finally, we have $\#_{\texttt{a}}(\texttt{ababab}) = 3$ and $\#_{\texttt{b}}(\texttt{ababab}) = 3$, therefore `ababab` does not satisfy the property and is not in S.

(d) Both methods are functions: `doubleInt` is defined for every `int` and it is clearly single-valued. Similarly, `addOneToFloat` is defined for every `float` and is also single-valued.

`doubleInt` is not surjective because it only produces even integers. By the pigeonhole principle, it then cannot be injective either since the domain and co-domain have the same finite cardinality. Alternatively, we expect, for example, `doubleInt(2^31)` to be 0.

`addOneToFloat` is not injective because if the absolute value of `number` is small enough, then the output will be 1 by rounding even if the input was non-zero. Again, the domain and co-domain have the same finite cardinality, so `addOneToFloat` cannot be surjective either.

A function is bijective if it is both injective and surjective, so neither function is bijective.

Learning outcomes: "Solve mathematical problems in algebra and set theory" (a, b, c); "Apply mathematical techniques to solve a problem within a computer science setting" (c, d).

Question 2 [SAT & Predicate Logic]

(a) Let p, q, r, s be atoms capturing the states of four cells, which can either be filled or empty: p is true if the cell is filled, and false if the cell is empty, and similarly for the other atoms. Consider the following formula:

$$(p \vee \neg q) \wedge (p \vee r) \wedge (p \vee s) \wedge (q \vee \neg p) \wedge (q \vee r) \wedge (q \vee s)$$
$$\wedge(\neg r \vee \neg p) \wedge (\neg r \vee \neg q) \wedge (\neg r \vee s) \wedge (\neg s \vee \neg p) \wedge (\neg s \vee \neg q) \wedge (\neg s \vee r)$$

(i) Using DPLL, find a valuation that shows that the above formula is satisfiable. Justify your answer as we did in the SAT lecture. **[6 marks]**

(ii) Is the formula valid? Justify your answer. **[2 marks]**

(iii) Explain in one sentence what property about the states of the four cells p, q, r, and s, this formula captures. **[2 marks]**

(b) Consider the following signature:

- Function symbols: `zero` (arity 0); `succ` (arity 1)
- Predicate symbols: $<$ (arity 2)

We will use infix notation for the binary symbol $<$. Consider the following formulas that capture properties of the above symbols:

- let S_1 be $\forall y.(\exists x.x < y) \rightarrow 0 < y$
- let S_2 be $\forall x.x < \texttt{succ}(x)$

For simplicity we write 0 for `zero`, 1 for `succ(zero)`, 2 for `succ(succ(zero))`, etc.

(i) Provide a constructive Natural Deduction proof of:

$$S_1 \to S_2 \to 0 < 2$$

(Hint: you can prove this formula without $[\forall I]$ and $[\exists E]$.) **[6 marks]**

(ii) Explain why the following tree is not a Natural Deduction proof. Justify your answer.

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{S_2}^{\,1}}{x < \mathtt{succ}(x)}\,[\forall E]}{\forall y.y < \mathtt{succ}(x)}\,[\forall I]}{\exists x.\forall y.y < x}\,[\exists I]}{S_2 \to \exists x.\forall y.y < x}\;1\;[\to I]$$

(Hint: keep in mind that the $\forall$ and $\exists$ rules have side conditions.) **[4 marks]**

Model answer / LOs / Creativity:

(a) (i) Here is a run of the DPLL algorithm:

- $(p \vee \neg q) \wedge (p \vee r) \wedge (p \vee s) \wedge (q \vee \neg p) \wedge (q \vee r) \wedge (q \vee s) \wedge (\neg r \vee \neg p) \wedge (\neg r \vee \neg q) \wedge (\neg r \vee s) \wedge (\neg s \vee \neg p) \wedge (\neg s \vee \neg q) \wedge (\neg s \vee r)$
- select $p = \mathbf{T}$
- we remove: $\cancel{(p \vee \neg q)} \wedge \cancel{(p \vee r)} \wedge \cancel{(p \vee s)} \wedge (q \vee \cancel{\neg p}) \wedge (q \vee r) \wedge (q \vee s) \wedge (\neg r \vee \cancel{\neg p}) \wedge (\neg r \vee \neg q) \wedge (\neg r \vee s) \wedge (\neg s \vee \cancel{\neg p}) \wedge (\neg s \vee \neg q) \wedge (\neg s \vee r)$
- we obtain: $q \wedge (q \vee r) \wedge (q \vee s) \wedge \neg r \wedge (\neg r \vee \neg q) \wedge (\neg r \vee s) \wedge \neg s \wedge (\neg s \vee \neg q) \wedge (\neg s \vee r)$
- select $q = \mathbf{T}$
- we remove: $\cancel{q} \wedge \cancel{(q \vee r)} \wedge \cancel{(q \vee s)} \wedge \neg r \wedge (\neg r \vee \cancel{\neg q}) \wedge (\neg r \vee s) \wedge \neg s \wedge (\neg s \vee \cancel{\neg q}) \wedge (\neg s \vee r)$
- we obtain: $\neg r \wedge \neg r \wedge (\neg r \vee s) \wedge \neg s \wedge \neg s \wedge (\neg s \vee r)$
- select $r = \mathbf{F}$
- we remove: $\cancel{\neg r} \wedge \cancel{\neg r} \wedge \cancel{(\neg r \vee s)} \wedge \neg s \wedge \neg s \wedge (\neg s \vee \cancel{r})$
- we obtain: $\neg s \wedge \neg s \wedge \neg s$
- select $s = \mathbf{F}$
- we remove: $\cancel{\neg s} \wedge \cancel{\neg s} \wedge \cancel{\neg s}$
- we obtain: no more clauses
- **SAT**

(ii) The formula is not valid. For example, set $p = \mathbf{T}$, $q = \mathbf{T}$, $r = \mathbf{T}$, and $s = \mathbf{T}$. This makes the formula false because for example $\neg r \vee \neg p$ is false.

(iii) The property captures that either the two first cells p and q are filled while the last two r and s are empty, or vise versa, that the two first cells p and q are empty while the last two r and s are filled.

(b) (i) Here is a proof of $S_1 \to S_2 \to 0 < 2$:

$$\dfrac{\dfrac{\dfrac{\overline{\forall y.(\exists x.x < y) \to 0 < y}^{\,1}}{(\exists x.x < 2) \to 0 < 2}[\forall E] \quad \dfrac{\dfrac{\overline{\forall x.x < \mathtt{succ}(x)}^{\,2}}{1 < 2}[\forall E]}{\exists x.x < 2}[\exists I]}{\dfrac{0 < 2}{S_2 \to 0 < 2}\,2\,[\to I]}[\to E]}{S_1 \to S_2 \to 0 < 2}\,1\,[\to I]$$

(ii) This is not a Natural Deduction because we cannot use $[\forall I]$ to derive $\forall y.y < \mathtt{succ}(x)$ from $x < \mathtt{succ}(x)$ because x is free in $\forall y.y < \mathtt{succ}(x)$.

Learning outcomes: "Understand and apply algorithms for key problems in logic such as satisfiability." (a); "Write formal proofs for propositional and predicate logic" (a & b); "Apply logical techniques to solve a problem within a computer science setting" (a & b).