Formal Reasoning 2021
Solutions Test Blocks 1, 2 and 3: Additional Test

(12/01/22)

1. Someone wants to write $a \leftrightarrow b$ using only negation and disjunction, and is considering the following two options:

$$a \leftrightarrow b \equiv \neg(a \vee b) \vee \neg(\neg a \vee \neg b)$$
$$a \leftrightarrow b \equiv \neg(\neg a \vee b) \vee \neg(a \vee \neg b)$$

Which of these logical equivalences is/are correct?

(a) Both are correct.

(b) is correct — (b) The first is correct, but the second is incorrect.

(c) The second is correct, but the first is incorrect.

(d) Both are incorrect.

Answer (b) is correct.

Let us create two truth tables:

a	b	$\neg a$	$\neg b$	$a \leftrightarrow b$	$a \vee b$	$\neg a \vee \neg b$	$\neg(a \vee b)$	$\neg(\neg a \vee \neg b)$	$\neg(a \vee b) \vee \neg(\neg a \vee \neg b)$
0	0	1	1	1	0	1	1	0	1
0	1	1	0	0	1	1	0	0	0
1	0	0	1	0	1	1	0	0	0
1	1	0	0	1	1	0	0	1	1

a	b	$\neg a$	$\neg b$	$a \leftrightarrow b$	$\neg a \vee b$	$a \vee \neg b$	$\neg(\neg a \vee b)$	$\neg(a \vee \neg b)$	$\neg(\neg a \vee b) \vee \neg(a \vee \neg b)$
0	0	1	1	1	1	1	0	0	0
0	1	1	0	0	1	0	0	1	1
1	0	0	1	0	0	1	1	0	1
1	1	0	0	1	1	1	0	0	0

So the first equivalence holds because the corresponding columns are exactly the same, but the second doesn't hold because the corresponding columns are not the same (in fact they are exactly opposites).

We can also show that the first one holds by rewriting the formula:

$$\begin{aligned}
\neg(a \vee b) \vee \neg(\neg a \vee \neg b) &\equiv \neg(a \vee b) \vee \neg(\neg b \vee \neg a) \\
&\equiv \neg(a \vee b) \vee (\neg\neg b \wedge \neg\neg a) \\
&\equiv \neg(a \vee b) \vee (b \wedge a) \\
&\equiv (\neg a \wedge \neg b) \vee (b \wedge a) \\
&\equiv a \leftrightarrow b
\end{aligned}$$

The last step may seem a bit large, but it should be clear[1] that $a \leftrightarrow b$ holds exactly when a and b both do not hold, or when a and b both hold. And we can rewrite the second formula to $\neg(a \leftrightarrow b)$, which is clearly not equivalent to $a \leftrightarrow b$.

$$\begin{aligned}
\neg(\neg a \vee b) \vee \neg(a \vee \neg b) &\equiv \neg(a \to b) \vee \neg(\neg b \vee a) \\
&\equiv \neg(a \to b) \vee \neg(b \to a) \\
&\equiv \neg((a \to b) \wedge (b \to a)) \\
&\equiv \neg(a \leftrightarrow b)
\end{aligned}$$

[1] If this is not clear to you, here we have the full proof using rewrite rules:

$$\begin{aligned}
a \leftrightarrow b &\equiv (a \to b) \wedge (b \to a) &&\equiv (\neg a \vee b) \wedge (\neg b \vee a) \\
&\equiv (\neg a \wedge (\neg b \vee a)) \vee (b \wedge (\neg b \vee a)) &&\equiv ((\neg a \wedge \neg b) \vee (\neg a \wedge a)) \vee ((b \wedge \neg b) \vee (b \wedge a)) \\
&\equiv ((\neg a \wedge \neg b) \vee \bot) \vee (\bot \vee (b \wedge a)) &&\equiv (\neg a \wedge \neg b) \vee (b \wedge a)
\end{aligned}$$

Note that we wrote it in this order because in the other direction it is a bit strange to add the '$\bot$' parts, whereas in this direction they come naturally.

It is clear that none of the other options can be correct at the same time.

2. Translate the following sentence into a formula of predicate logic with equality as well as possible.

   There is at most one king in the Netherlands.

   (Consider for instance the situation when the sovereign is female: in that case there are zero kings.) Use for the dictionary:

N	the domain of Dutch people
$K(x)$	x is king

   There are several solutions possible, for instance

   $$(\forall x \in N\, \neg K(x)) \vee (\exists x \in N\ (K(x) \wedge (\forall y \in N\ (K(y) \to x = y))))$$

   which means

   Either there are no kings or there is exactly one king.

   or

   $$\exists x \in N\ (\forall y \in N\ (K(y) \to x = y))$$

   which means

   There is a special Dutch person for whom it holds that every Dutch person who is king, is that special person.

3. We are looking for a regular expression r such that

   $$\mathcal{L}(r) = \{w \in \{a, b\}^* \mid w \text{ contains an even number of } a\text{'s}\}$$

   Which of the following regular expressions is a good solution for this?

   (a) is correct

   (a) $r = ((ab^*a)^*b^*)^*$
   (b) $r = (aa \cup b)^*$
   (c) $r = (ab^*a)^* \cup b^*$
   (d) $r = ((aa)^*b^*)^*$

   Answer (a) is correct.

   (a) $r = ((ab^*a)^*b^*)^*$:
   Every word created with this regular expression consists of blocks that are either just b's or two a's with an arbitrary number of b's in between. So you can get exactly all words that have an even number of a's.

   (b) $r = (aa \cup b)^*$:
   With this expression you cannot create the word aba.

   (c) $r = (ab^*a)^* \cup b^*$:
   With this expression you cannot create the word $abab$.

   (d) $r = ((aa)^*b^*)^*$:
   With this expression you cannot create the word aba.

So the first answer is correct.

It is clear that none of the other options can be correct at the same time.

4. Consider the context-free grammar G_4:

$$S \to aA \mid bS \mid \lambda$$
$$A \to aS \mid bA$$

The language of this grammar is:

$$\mathcal{L}(G_4) = \{w \in \{a, b\}^* \mid w \text{ contains an even number of } a\text{'s}\}$$

Someone wants to prove that $ab \notin \mathcal{L}(G_4)$ using the property

$$P(w) := \big[w \text{ contains an even number of symbols from the set } \{a, A\}\big]$$

Is this a suitable invariant for proving this?

(a) Yes.

(b) No.

And now in more detail: is this a suitable invariant for proving this?

(a) is correct

(a) Yes, because S has an even number of symbols from $\{a, A\}$, and every step in a production adds zero or two to the count of symbols from $\{a, A\}$.

(b) Yes, because every word in $\mathcal{L}(G_4)$ has an even number of a's and no A's.

(c) No, because the right hand side aS only contains a single symbol from $\{a, A\}$.

(d) No, because S does not contain a symbol from $\{a, A\}$.

Answer (a) is correct.

It is clear that S contains zero elements of $\{a, A\}$ and zero is even. The rule $S \to aA$ adds two elements of $\{a, A\}$. All other rules add zero elements of $\{a, A\}$. So if we have a word v for which $P(v)$ holds, and hence it has an even number of elements of $\{a, A\}$, and a word v' such that $v \to v'$, then $P(v')$ also holds since the number of elements of $\{a, A\}$ is increased by an even number and hence it stays an even number. Hence property P is an invariant.

In addition, $P(ab)$ does not hold, so it is indeed a useful invariant for proving that $ab \notin \mathcal{L}(G_4)$.

The second answer makes no sense because although this property holds, it is not relevant for being a useful invariant.

The third and the fourth answer make no sense because the answer is 'yes'.

5. How many states has the smallest deterministic finite automaton M_5 that recognizes the language:

$$\mathcal{L}(M_5) = \{w \in \{a, b\}^* \mid w \text{ contains } aa, \text{ and } w \text{ contains an even number of } b\text{'s}\}$$

(a) Less than six states.

(b) is correct

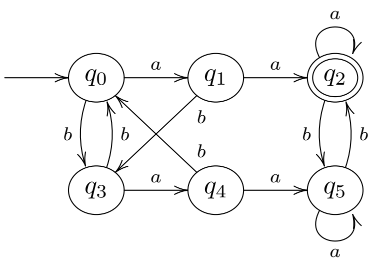
(b) Six states.

(c) More than six states.

(d) This language cannot be recognized by a deterministic finite automaton.

Answer (b) is correct.

A deterministic automaton that recognizes the language is:



There are two independent properties that need to be memorized: whether we have aa (hence at least two consecutive a's) and whether the number of b's is even. So in total this gives six different situations:

- q_0: there was no series of two consecutive a's read, an even number of b's was read, and the last symbol read was not an a.
- q_1: there was no series of two consecutive a's read, an even number of b's was read, and the last symbol read was an a.
- q_2: there was a series of two consecutive a's read and an even number of b's was read.
- q_3: there was no series of two consecutive a's read, an odd number of b's was read, and the last symbol read was a b.
- q_4: there was no series of two consecutive a's read, an odd number of b's was read, and the last symbol read was an a.
- q_5: there was a series of two consecutive a's read and an odd number of b's was read.

6. There are six isomorphisms from the graph K_3 to itself (an isomorphism of a graph to itself is called an *automorphism* for that graph):

$1 \mapsto 1$	$1 \mapsto 1$	$1 \mapsto 2$	$1 \mapsto 2$	$1 \mapsto 3$	$1 \mapsto 3$
$2 \mapsto 2$	$2 \mapsto 3$	$2 \mapsto 1$	$2 \mapsto 3$	$2 \mapsto 1$	$2 \mapsto 2$
$3 \mapsto 3$	$3 \mapsto 2$	$3 \mapsto 3$	$3 \mapsto 1$	$3 \mapsto 2$	$3 \mapsto 1$

How many automorphisms are there for the graph K_n?

(a) is correct

(a) $n!$

(b) $2n$

(c) $n^2 - n$

(d) None of the above is correct.

Answer (a) is correct.

As all vertices in the graph K_n are completely identical with respect to the structure of the graph, we may map each vertex to each vertex. So for the first vertex we have n options, for the second $n-1$, for the third $n-2$, and so on until for the last we only have 1 option left. So this gives $n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1 = n!$ isomorphisms from K_n to itself, or automorphisms for K_n.

For $n = 3$, both $2n$ and $n^2 - n$ also give the proper solution of six, but that does not hold for larger n.

7. In an induction proof, for all $k \geq 0$ we assume $P(k)$ (the induction hypothesis: IH). So why aren't we done with this, because it seems that we have then already that the property holds for all natural numbers?

   (a) Because the assumption $P(k)$ only may be used when proving $P(k+1)$ for the same k.

   (b) Because in the induction step it does not say 'assume that $P(k)$ for all $k \geq 0$, and then prove ...', but 'for all $k \geq 0$, assume that $P(k)$ holds and then prove ...'. In other words, the 'assume' and the 'for all' are in the wrong order for using the induction hypothesis for all k as a general assumption.

   (c) Because the induction scheme amounts to the predicate logic formula:

   $$P(0) \wedge \big(\forall k \in \mathbb{N}\, (P(k) \to P(k+1))\big) \to \forall n \in \mathbb{N}\, P(n)$$

   and not to the formula:

   $$P(0) \wedge \big((\forall k \in \mathbb{N}\, P(k)) \to P(k+1)\big) \to \forall n \in \mathbb{N}\, P(n)$$

   (which is illegal anyway, because the k is not in scope at the $k+1$.)

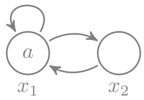
   (d) These three answers above are all correct, as they are all different ways of saying the same thing.

(d) is correct

Answer (d) is correct.

The formulas in predicate logic are translations of the statements in natural language in the second option. And in the first option it is also indicated that for each k for which you assume that $P(k)$ holds, you have to prove $P(k+1)$. So it is not that you assume $P(k)$ to hold for all k at the same time and then have some general proof of $P(k+1)$. (Due to the scope error mentioned in the third option, it is already difficult to formulate this situation in natural language.)

8. Consider the Kripke model $\mathcal{M}_8$:



We want to know whether the statements

$$\mathcal{M}_8 \vDash \Box\Diamond a \qquad \text{and} \qquad \mathcal{M}_8 \vDash \Diamond\Box a$$

hold or not.

In order to get to the right conclusion you have to fill out the following $\Vdash$-table first and then answer the questions about the two statements given. (Sorry for the randomization of 0 and 1, which is a feature in Cirrus that we cannot turn off.)

model	world $\Vdash$	a	$\Box a$	$\Diamond a$	$\Box\Diamond a$	$\Diamond\Box a$
$\mathcal{M}_8$	x_1					
$\mathcal{M}_8$	x_2					

The statement $\mathcal{M}_8 \vDash \Box\Diamond a$ [Choose from: holds / does not hold].

And the statement $\mathcal{M}_8 \vDash \Diamond\Box a$ [Choose from: holds / does not hold].

This is the complete $\Vdash$-table.

model	world $\Vdash$	a	$\Box a$	$\Diamond a$	$\Box\Diamond a$	$\Diamond\Box a$
$\mathcal{M}_8$	x_1	1	0	1	1	1
$\mathcal{M}_8$	x_2	0	1	1	1	0

So the statement $\mathcal{M}_8 \vDash \Box\Diamond a$ holds, as both $\mathcal{M}_8, x_1 \Vdash \Box\Diamond a$ and $\mathcal{M}_8, x_2 \Vdash \Box\Diamond a$ hold.

And the statement $\mathcal{M}_8 \vDash \Diamond\Box a$ does not hold, as $\mathcal{M}_8, x_2 \Vdash \Diamond\Box a$ does not hold.