Formal Reasoning 2025
Solutions Exam
(12/01/26)

There are six sections, with together 16 multiple choice questions and 6 open questions. Each multiple choice question is worth 3 points, and each open question is worth 7 points. The mark for this test is the number of points divided by ten, The first 10 points are free. Good luck!

Propositional logic

1. Give a formula of propositional logic that expresses that the weekend is Saturday and Sunday. Use for the dictionary:

WE	it is weekend
Sat	it is Saturday
Sun	it is Saturday

   (a) $\mathsf{WE} \leftrightarrow \mathsf{Sat} \land \mathsf{Sun}$

   (b) is correct (b) $\mathsf{WE} \leftrightarrow \mathsf{Sat} \lor \mathsf{Sun}$

   (c) $\mathsf{WE} \equiv \mathsf{Sat} \land \mathsf{Sun}$

   (d) $\mathsf{WE} \equiv \mathsf{Sat} \lor \mathsf{Sun}$

   Answer (b) is correct.

   As the question is about formulas of propositional logic, the options with '$\equiv$' are immediately wrong, as these are statements about formulas and not formulas themselves.

   Now if we look at the '$\leftrightarrow$' options, it follows that it should be the option with '$\lor$', as otherwise 'it is weekend' implies that it is both Saturday and Sunday at the same time.

2. Give the number of zeroes and ones for each of the columns of the truth table of the propositional formula:

$$(\neg a \lor b) \land (a \lor \neg b)$$

   For each column, both give the formula at the top of that column, as well as the number of zeroes and ones in it. Make sure you do not forget the columns with the atoms at the start of the table.

   As Ans doesn't like 'verbatim' text, the template for the table is given as a Python code block, but don't worry about the syntax highlighting, just type your answers in ASCII.

   Add rows or remove rows as needed. Please keep the bars to separate the columns, as that way we can still parse the table if its alignment is not perfect.

   Answer (b) is correct.

This is the full truth table:

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

So this is the answer:

```
column                           | number of zeroes | number of ones
-------------------------------------------------------------------
a                                | 2                | 2
b                                | 2                | 2
not a                            | 2                | 2
not b                            | 2                | 2
not a or b                       | 1                | 3
a or not b                       | 1                | 3
(not a or b) and (a or not b)    | 2                | 2
```

3. Consider the following two formulas of propositional logic:

$$f_1 := (a \wedge b) \vee (\neg a \wedge c)$$
$$f_2 := (a \to b) \wedge (\neg a \to c)$$

What is the case?

(a) is correct

(a) $f_1 \equiv f_2$, which amounts to $f_1 \vDash f_2$ and $f_2 \vDash f_1$

(b) $f_1 \vDash f_2$, but $f_2 \nvDash f_1$

(c) $f_1 \nvDash f_2$, but $f_2 \vDash f_1$

(d) $f_1 \nvDash f_2$ and $f_2 \nvDash f_1$

Answer (a) is correct.

This is the full truth table:

a	b	c	$\neg a$	$a \wedge b$	$\neg a \wedge c$	$(a \wedge b) \vee (\neg a \vee c)$	$a \to b$	$\neg a \to c$	$(a \to b) \wedge (\neg a \to c)$
0	0	0	1	0	0	0	1	0	0
0	0	1	1	0	1	1	1	1	1
0	1	0	1	0	0	0	1	0	0
0	1	1	1	0	1	1	1	1	1
1	0	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	1	0
1	1	0	0	1	0	1	1	1	1
1	1	1	0	1	0	1	1	1	1

As the columns for f_1 and f_2 are exactly the same, we have that $f_1 \equiv f_2$ and hence also that $f_1 \vDash f_2$, and $f_2 \vDash f_1$.

We can also prove the equivalence by using logical laws:

	$(a \wedge b) \vee (\neg a \wedge c)$	distributive law
$\equiv$	$((a \wedge b) \vee \neg a) \wedge ((a \wedge b) \vee c)$	distributive law
$\equiv$	$((a \vee \neg a) \wedge (b \vee \neg a)) \wedge ((a \wedge b) \vee c)$	law of excluded middle
$\equiv$	$(\text{true} \wedge (b \vee \neg a)) \wedge ((a \wedge b) \vee c)$	law of identity
$\equiv$	$(b \vee \neg a) \wedge ((a \wedge b) \vee c)$	distributive law
$\equiv$	$(b \vee \neg a) \wedge ((a \vee c) \wedge (b \vee c))$	associativity law
$\equiv$	$((b \vee \neg a) \wedge (a \vee c)) \wedge (b \vee c)$	distributive law
$\equiv$	$(\neg a \vee (b \vee b)) \wedge ((a \vee c) \vee c)$	idempotence law
$\equiv$	$(\neg a \vee b) \wedge ((a \vee c) \vee c)$	material implication
$\equiv$	$(a \to b) \wedge ((a \vee c) \vee c)$	associativity law
$\equiv$	$(a \to b) \wedge (a \vee (c \vee c))$	idempotence law
$\equiv$	$(a \to b) \wedge (a \vee c)$	double negation law
$\equiv$	$(a \to b) \wedge (\neg\neg a \vee c)$	material implication
$\equiv$	$(a \to b) \wedge (\neg a \to c)$	

Note that not all of these are in the course notes, so it was not the idea that you could come up with this solution.

Predicate logic

4. Formalize the following sentence as a formula of predicate logic:

   January is in the winter.

   Use for the dictionary:

D	the domain of days
$J(x)$	x is in January
$W(x)$	x is in the winter

   As the domain is in terms of days, this sentence should be quantified in terms of days:

   Each day that is in January is in the winter.

   And this can easily be translated to the formula:

   $$\forall d \in D\ [J(d) \to W(d)]$$

5. Consider the formula of predicate logic:

   $$f := \forall x, y, z \in N\, [R(x,y) \to R(y,z) \to R(x,z)]$$

   What does *not* hold?

   (a) $((\mathbb{N}, <), I) \vDash f$, with I mapping N to $\mathbb{N}$ and $R(x,y)$ to $x < y$

   (b) $((\mathbb{N}, \leq), I) \vDash f$, with I mapping N to $\mathbb{N}$ and $R(x,y)$ to $x \leq y$

   (c) $((\mathbb{N}, =), I) \vDash f$, with I mapping N to $\mathbb{N}$ and $R(x,y)$ to $x = y$

   (d) is correct — (d) $((\mathbb{N}, \neq), I) \vDash f$, with I mapping N to $\mathbb{N}$ and $R(x,y)$ to $x \neq y$

Answer (d) is correct.

Because of the right-associativity of the implication, the formula is equivalent to

$$\forall x, y, z \in N\, [R(x, y) \to (R(y, z) \to R(x, z))]$$

which in turn is equivalent to

$$\forall x, y, z \in N\, [(R(x, y) \wedge R(y, z)) \to R(x, z)]$$

In natural language, this can be understood by translating these formulas to

> *For all $x, y, z \in N$ it holds that if $R(x, y)$ holds, then if also $R(y, z)$ holds, then $R(x, z)$ holds.*

and

> *For all $x, y, z \in N$ it holds that if $R(x, y)$ and $R(y, z)$ both hold, then $R(x, z)$ holds.*

And this formula (or sentence) represents the property of transitivity. Now note that $<$, $\leq$, and $=$ are indeed transitive relations, but $\neq$ is not.

The property of transitivity has been discussed in relation to the correspondence theorem between Kripke frames and axioms. However, it is not needed to understand this to come up with this counter example. Let $x := 3$, $y := 7$, and $z := 3$. According to the formula it should be that if $3 \neq 7$, which is true, and if $7 \neq 3$, which is also true, then $3 \neq 3$, which is definitely not true.

6. We want to formalize the following *true* sentence as a formula of predicate logic with equality:

> *Only zero is divisible by all natural numbers.*

We use for the dictionary:

N	the domain of natural numbers
zero	the number zero
divides(x, y)	y is a multiple of x

Which of the following formulas is a correct formalization of this?

(a) $\forall x \in N\, [\,\mathsf{divides}(x, \mathsf{zero})\,]$

(b) $\forall x \in N\, [\,\mathsf{divides}(\mathsf{zero}, x)\,]$

(c) $\forall x \in N\, [\,\forall y \in N\, \mathsf{divides}(x, y) \leftrightarrow x = \mathsf{zero}\,]$

(d) is correct

(d) $\forall x \in N\, [\,\forall y \in N\, \mathsf{divides}(y, x) \leftrightarrow x = \mathsf{zero}\,]$

Answer (d) is correct.

Note that the name divides may be a bit confusing when we talk about division by zero, however, as divides is only a predicate name in a formula that gets translated to something in terms of *multiples*, the statement is indeed true.

Now let us look at the different options:

(a) $\forall x \in N\,[\,\mathsf{divides}(x, \mathsf{zero})\,]$ This states that every natural number is a divisor of zero, so zero is indeed divisible by all natural numbers. However, it lacks the fact that zero is the only number for which this holds.

(b) $\forall x \in N\,[\,\mathsf{divides}(\mathsf{zero}, x)\,]$ This states that every natural number is divisible by zero, which is not true, as one is not divisible by zero.

(c) $\forall x \in N\,[\,\forall y \in N\,\mathsf{divides}(x, y) \leftrightarrow x = \mathsf{zero}\,]$ This states that zero is the only number that divides all natural numbers. As we have already seen, one is not divisible by zero, so this cannot be true.

(d) $\forall x \in N\,[\,\forall y \in N\,\mathsf{divides}(y, x) \leftrightarrow x = \mathsf{zero}\,]$ This is indeed correct. It combines the fact that zero is divisible by all natural numbers with the claim that zero is the only natural number that is divisible by all natural numbers.

Discrete mathematics

7. Does there exist a graph that has a Hamiltonian circuit, but has no Eulerian path?

(b) is correct

(a) Yes, there is a graph with three vertices that has this property.

(b) Yes, there is a graph with four vertices that has this property.

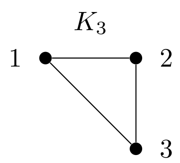
(c) Yes, but any example has to have at least five vertices.

(d) No, such a graph does not exist.

Answer (b) is correct.

If a graph with three vertices has a Hamiltonian circuit, it must be the graph

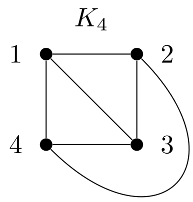
$$K_3 := \langle \{1, 2, 3\}, \{(1, 2), (1, 3), (2, 3)\} \rangle$$



as this is the smallest and largest graph with three vertices that has a cycle. However, the same cycle is also an Eulerian circuit. So in particular, K_3 has an Eulerian path, which excludes the first option.

A graph with four vertices that has a Hamiltonian circuit, but doesn't have an Eulerian path, is the graph

$$K_4 := \langle \{1, 2, 3, 4\}, \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\} \rangle$$



It clearly has a Hamiltonian circuit: $1 \to 2 \to 3 \to 4 \to 1$. However, as it is a connected graph with at least two vertices, we can apply Euler's theorem, and since all four vertices have degree three, the graph doesn't have an Eulerian path, so the second option is correct.

Note that this automatically excludes the third and the fourth option.

8. Consider the graph $G_8 := \langle V_8, E_8 \rangle$, where

$$\begin{aligned} V_8 &:= \{\langle x, y, z\rangle \mid x, y, z \in \{0, 1\}\} \\ E_8 &:= \{(\langle x, y, z\rangle, \langle x', y', z'\rangle) \mid |x - x'| + |y - y'| + |z - z'| = 1\} \end{aligned}$$

So $\langle 0, 0, 0\rangle$ and $\langle 0, 1, 0\rangle$ are connected by an edge as $|0 - 0| + |0 - 1| + |0 - 0| = 0 + 1 + 0 = 1$, but $\langle 0, 0, 0\rangle$ and $\langle 1, 0, 1\rangle$ are not connected as $|0 - 1| + |0 - 0| + |0 - 1| = 1 + 0 + 1 = 2$.

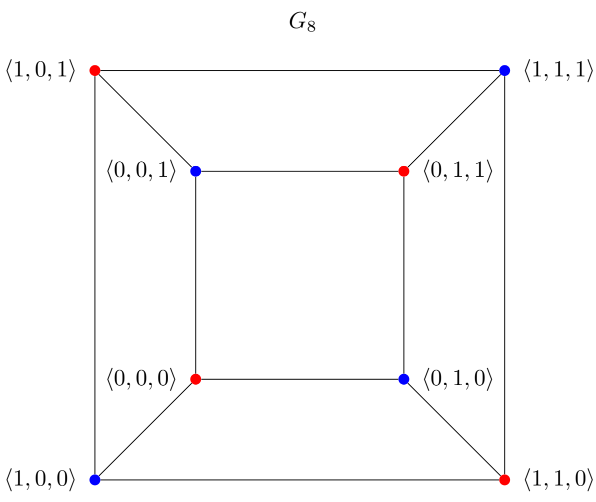
This graph has eight vertices and twelve edges. What is the case?

(a) is correct

(a) G_8 is planar and bipartite
(b) G_8 is planar, but not bipartite
(c) G_8 is bipartite, but not planar
(d) G_8 is neither planar, nor bipartite

Answer (a) is correct.

The graph G_8 is just the hypercube graph. And as the image shows, it is both planar and bipartite.



9. The *Golomb sequence* can be recursively defined as:

$$\begin{array}{rcll} a(1) & = & 1 & \\ a(n+1) & = & 1 + a(n+1-a(a(n))) & \text{for } n \geq 1 \end{array}$$

What is the value of $a(6)$?

(a) 3

(b) is correct (b) 4

(c) 5

(d) none of the above

Answer (b) is correct.

See this table:

$$\begin{array}{lclclclclclclcl}
a(1) & = & 1 \\
a(2) & = & a(1+1) & = & 1+a(1+1-a(a(1))) & = & 1+a(1+1-a(1)) & = & 1+a(1+1-1) & = & 1+a(1) & = & 1+1 & = & 2 \\
a(3) & = & a(2+1) & = & 1+a(2+1-a(a(2))) & = & 1+a(2+1-a(2)) & = & 1+a(2+1-2) & = & 1+a(1) & = & 1+1 & = & 2 \\
a(4) & = & a(3+1) & = & 1+a(3+1-a(a(3))) & = & 1+a(3+1-a(2)) & = & 1+a(3+1-2) & = & 1+a(2) & = & 1+2 & = & 3 \\
a(5) & = & a(4+1) & = & 1+a(4+1-a(a(4))) & = & 1+a(4+1-a(3)) & = & 1+a(4+1-2) & = & 1+a(3) & = & 1+2 & = & 3 \\
a(6) & = & a(5+1) & = & 1+a(5+1-a(a(5))) & = & 1+a(5+1-a(3)) & = & 1+a(5+1-2) & = & 1+a(4) & = & 1+3 & = & 4 \\
a(7) & = & a(6+1) & = & 1+a(6+1-a(a(6))) & = & 1+a(6+1-a(4)) & = & 1+a(6+1-3) & = & 1+a(4) & = & 1+3 & = & 4 \\
a(8) & = & a(7+1) & = & 1+a(7+1-a(a(7))) & = & 1+a(7+1-a(4)) & = & 1+a(7+1-3) & = & 1+a(5) & = & 1+3 & = & 4 \\
a(9) & = & a(8+1) & = & 1+a(8+1-a(a(8))) & = & 1+a(8+1-a(4)) & = & 1+a(8+1-3) & = & 1+a(6) & = & 1+4 & = & 5
\end{array}$$

10. Prove using induction that $n! \geq 2^n$ for all $n \geq 4$.

Follow the template as closely as possible.

0 **Proposition:**

$n! \geq 2^n$ for all $n \geq 4$.

1 **Proof** by induction on n.

We first define our predicate P as:

2 $$P(n) \quad := \quad [\, n! \geq 2^n \,]$$

3 **Base Case.** We show that $P(4)$ holds, i.e. we show that

$4! \geq 2^4$

4 This indeed holds, because $4! = 24 \geq 16 = 2^4$.

5 **Induction Step.** Let k be any natural number such that $k \geq 4$.

6 Assume that we already know that $P(k)$ holds, i.e. we assume that

$k! \geq 2^k$ (Induction Hypothesis IH)

7 We now show that $P(k+1)$ also holds, i.e. we show that

$(k+1)! \geq 2^{k+1}$

8 This indeed holds, because

$$\begin{array}{rcll} (k+1)! & = & (k+1) \cdot k! & \text{factorial} \\ & \stackrel{\text{IH}}{\geq} & (k+1) \cdot 2^k & \text{by IH} \\ & \geq & 2 \cdot 2^k & k \geq 4 \text{ so } k+1 \geq 2 \\ & = & 2^{k+1} & \text{elementary math} \end{array}$$

9 Hence it follows by induction that $P(n)$ holds for all $n \geq 4$.

11. The *Catalan numbers* can be defined by:

$$C_n := \frac{1}{n+1}\binom{2n}{n}$$

for $n \geq 0$.

It turns out that these are integers, despite the division by $n+1$. What is another formula for these numbers?

(a) $C_n = n!$

(b) is correct

(b) $C_n = \frac{(2n)!}{n!(n+1)!}$

(c) $C_n = n^2 - 2n + 2$

(d) none of the above

Answer (b) is correct.

We know that

$$\binom{a}{b} = \frac{a!}{b!(a-b)!}$$

So

$$\binom{2n}{n} = \frac{(2n)!}{n!(2n-n)!} = \frac{(2n)!}{n!n!}$$

So

$$\frac{1}{n+1}\binom{2n}{n} = \frac{1}{n+1}\frac{(2n)!}{n!n!} = \frac{(2n)!}{n!(n+1)!}$$

And to show that the other options are indeed incorrect, we compute the first of these values:

formula n	0	1	2	3	4
$C(n)$	1	1	2	5	14
$n!$	1	1	2	6	24
$n^2 - 2n + 2$	2	1	2	5	10

Languages

12. We consider the language L_{12} that is defined with alphabet $\Sigma = \{a, b\}$ by:

$$L_{12} := \left(\overline{\{a\}}\right)^*$$

In this, we of course have:

$$\overline{\{a\}} = \{w \in \{a, b\}^* \mid w \neq a\}$$

What is the case?

(a) $L_{12} = \varnothing$

(b) $L_{12} = \{a\}$

(c) is correct

(c) $L_{12} = \overline{\{a\}}$

(d) $L_{12} = \Sigma^*$

Answer (c) is correct.

Let us see which words are in $\overline{\{a\}}$:

$$\lambda, \quad b, \quad aa, ab, ba, bb, \quad aaa, aab, aba, baa, abb, bab, bba, bbb, \quad \dots$$

In other words, $\overline{\{a\}} = \Sigma^* \setminus \{a\}$.

Now for $\left(\overline{\{a\}}\right)^*$ we need to take finite concatenations. However, if $w_1, w_2 \in \overline{\{a\}}$, then $w_1 w_2 \in \overline{\{a\}}$. We only need to check that $w_1 w_2 \neq a$. Given the length that means that either $w_1 = \lambda$ and $w_2 = a$, or $w_1 = a$ and $w_2 = \lambda$. However as $a \notin \overline{\{a\}}$, this cannot happen.

So $L_{12} = \left(\overline{\{a\}}\right)^* = \overline{\{a\}}$.

13. What is a regular expression for the language:

$$\overline{\{a\}} = \{w \in \{a, b\}^* \mid w \neq a\}$$

(a) $\overline{a}$

(b) b^*

(c) is correct

(c) $\lambda \cup (aa \cup ab \cup b)(a \cup b)^*$

(d) This is not a regular language, and therefore it can not be described by a regular expression.

Answer (c) is correct.

We have seen before that

$$\overline{\{a\}} = \lambda, \quad b, \quad aa, ab, ba, bb, \quad aaa, aab, aba, baa, abb, bab, bba, bbb, \quad \dots$$

In particular it is clear that the expressions a and b^* do not generate $\overline{\{a\}}$.

Note that $\overline{\{a\}}$ is almost $\mathcal{L}((a \cup b)^*)$, except that this language does include a. We can exclude this word a by listing all words of length one and two separately, and then add arbitrary a's and b's. Note that we have to take λ separately, as otherwise a can be created.

$$\lambda \cup (b \cup aa \cup ab \cup ba \cup bb)(a \cup b)^*$$

However, if we look at this expression, we can see that ba and bb are not needed as we have b followed by $(a \cup b)^*$. So we can reduce it to

$$\lambda \cup (b \cup aa \cup ab)(a \cup b)^*$$

which in turn generates the same language as

$$\lambda \cup (aa \cup ab \cup b)(a \cup b)^*$$

14. What is a context-free grammar for the language:

$$\overline{\{a\}} = \{w \in \{a, b\}^* \mid w \neq a\}$$

(a) $S \to SbS \mid aa \mid \lambda$

(b) $S \to AA \mid b \mid \lambda$
$A \to AA \mid a \mid b$

(b) is correct

(c) $S \to \overline{A}$
$A \to a$

(d) The complement of a context-free grammar is not context-free, which means this language can not be produced by a context-free grammar.

Answer (b) is correct.

The first option cannot produce the word ab.

The third option is syntactically incorrect.

The fourth option makes no sense, as there is no such thing as the complement of a context-free grammar. Note that it also doesn't hold for context-free languages: Σ^* is regular and hence context-free, and $\overline{\Sigma^*} = \emptyset$, which is also a regular language and hence it has a context-free grammar.

So the second option must be correct...

And it is. The words λ and b are produced by $S \to \lambda$ and $S \to b$. Words of length n where $n \geq 2$ are produced by

$$S \to AA \to \underbrace{AAA \to AAAA \to \overbrace{A \cdots A}^{n}}_{n-2 \text{ times } A \to AA} \to \ldots$$

followed by n times $A \to a$ or $A \to b$.

Note that due to $S \to AA$ it is not possible to produce the word a.

15. Consider the grammar G_{15}:

$$\begin{array}{lcl} S & \to & aA \\ A & \to & B \mid \lambda \\ B & \to & B \mid bB \end{array}$$

Someone claims being able to show that $ba \notin \mathcal{L}(G_{15})$ using the property

$$P_{15}(w) = \text{the word } w \text{ does not contain the subword } ba$$

as an invariant for this language. Is this correct?

(a) Yes, that is correct. There are no words in the language $\mathcal{L}(G_{15})$ that contain the symbol b.

(b) Yes, that is correct. It is not possible to have a production starting in S that produces a word containing ba.

(c) is correct

(c) No, that is not correct. The property P_{15} is not an invariant for the language L_{15}.

(d) No, that is not correct. Although the property P_{15} is an invariant for the language L_{15}, the word ba does not satisfy it, which means it cannot be used to show that this word is not in the language.

Answer (c) is correct.

The property P_{15} is not an invariant. Note that $P_{15}(bAa)$ holds. However, $bAa \to ba$ by applying $A \to \lambda$. And obviously, $P_{15}(ba)$ does not hold.

As it is not an invariant, the 'Yes' options are incorrect, even though the explanations in these options are correct.

The other 'No' option is also incorrect. It is actually needed that $P_{15}(ba)$ does not hold to show that $ba \in \mathcal{L}(G_{15})$.

Did you see that this is a grammar that generates the language $\{a\}$?

16. Again, consider the grammar:

$$\begin{array}{lcl} S & \rightarrow & aA \\ A & \rightarrow & B \mid \lambda \\ B & \rightarrow & B \mid bB \end{array}$$

For this grammar, give all:

- λ-rules
- chain rules
- useless symbols

There is one λ-rule: $A \rightarrow \lambda$.

There are two chain rules: $A \rightarrow B$ and $B \rightarrow B$.

There are two useless symbols: B and b.

Automata

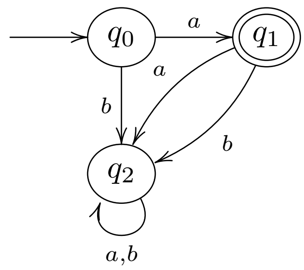
17. Give a deterministic finite automaton (DFA) M_{17} for the language:

$$\overline{\{a\}} = \{w \in \{a,b\}^* \mid w \neq a\}$$

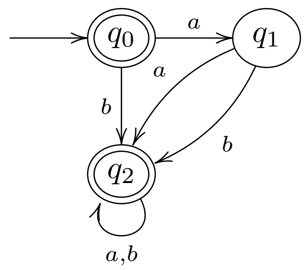
Either create an image in `TikZiT` and copy-paste it into Ans, or give the automaton as a tuple $\langle \Sigma, Q, q_0, F, \delta \rangle$, by writing:

```
M17 = <Sigma, Q, q0, F, delta>
Sigma = { ... }
Q = { q0, ... }
F = { ... }
delta(q0,a) = ...
...
```

As the language L_{17} is actually the complement of a simpler language $\{a\}$, it makes sense to create a DFA for this simpler language first:



And then swap final states and non-final states. This gives the DFA M_{17}:



This automaton can be written as the tuple

```
M17 = <Sigma, Q, q0, F, delta>
Sigma = { a, b}
Q = { q0, q1, q2 }
F = { q0, q2 }
delta(q0,a) = q1
delta(q0,b) = q2
delta(q1,a) = q2
delta(q1,b) = q2
delta(q2,a) = q2
delta(q2,b) = q2
```

18. Does the language

$$\overline{\{a\}} = \{w \in \{a, b\}^* \mid w \neq a\}$$

have a right linear context-free grammar?

(a) is correct

(a) Yes, this language has a right linear context-free grammar, but there are also context-free languages that do not have a right linear context-free grammar.
(b) Yes, all context-free languages also have a right linear context-free grammar.
(c) No, but it does have a context-free grammar that is not right linear.
(d) No, it does not have a context-free grammar.

Answer (a) is correct.

Indeed, there exists a right linear context-free grammar that produces $\overline{\{a\}}$:

$$\begin{array}{lcl} S & \rightarrow & aA \mid B \mid \lambda \\ A & \rightarrow & aB \mid bB \\ B & \rightarrow & aB \mid bB \mid \lambda \end{array}$$

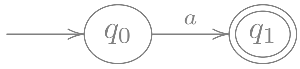
In addition, the language $\{a^n b^n \mid n \in \mathbb{N}\}$ is context-free, as it is produced by this grammar:

$$S \rightarrow aSb \mid \lambda$$

However, this language is not regular, so there doesn't exist a right-linear context-free grammar that produces this language.

This obviously means that the other options are incorrect.

19. The language $\{a\}$ is recognized by a non-deterministic finite automaton (NFA) with only two states:



Does that mean that its complement

$$\overline{\{a\}} = \{w \in \{a,b\}^* \mid w \neq a\}$$

is also recognized by a non-deterministic finite automaton with two states?

(a) Yes, one can transform an NFA to recognize the complement of its language by flipping all states between final and non-final.

(b) Yes, but that automaton is not what one gets from the above automaton by flipping the states between final and non-final.

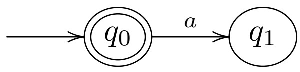
(c) No, two states are not enough, but this language *is* recognized by a non-deterministic finite automaton with more than two states.

(c) is correct

(d) No, this language is not recognized by any non-deterministic finite automaton.

Answer (c) is correct.

If we simply swap the final states and non-final states, we get this NFA:

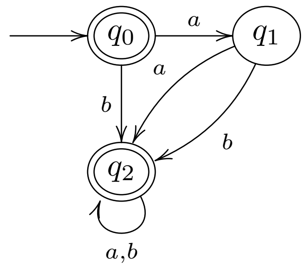


However, this automaton only accepts the word λ! Now, let us try to add some arrows to accept the words aa, and aaa.

- As a should not be accepted, there can't be an a-loop on q_0.
- As aa should be accepted, there has to be an a-arrow from q_1 to q_0 as that is the only final state.
- Note that it can't be done with an a-loop on q_1 and a λ-transition from q_1 to q_0, as this would lead to acceptance of a.
- It is also not possible to add a λ-transition from q_0 to q_1, as that would also lead to acceptance of a.
- So we have $\delta(q_0, a) = \{q_1\}$ and $\delta(q_1, a) = \{q_0\}$, but there are no other a-transitions and no λ-transitions.
- Now this means that aaa ends up in q_1 and hence it is not accepted, where it should have been.

So it is not possible to have an NFA with two states that accepts this language!

However, there does exist an NFA with three states that accepts this language, namely the DFA that we have already seen before:



Modal logic

20. In doxastic logic, $\Box f$ means that you believe that f is true, while $\Box\neg f$ means that you believe that f is not true. But which formula formalizes the third option, where you are agnostic about the truth of f, i.e., where you believe that f might be true but that it also might be false.

    (a) $\Diamond f$
    (b) $\Diamond\neg f$
    (c) $\neg\Box f$
    (d) is correct (d) $\neg\Box f \wedge \Diamond f$

    Answer (d) is correct.

    In doxastic logic, *you believe that f might be true* can be formalized by $\Diamond f$. Likewise, *you believe that f might be false* is formalized by $\Diamond\neg f$, which is equivalent to $\neg\Box f$. Combining this, we get $\neg\Box f \wedge \Diamond f$.

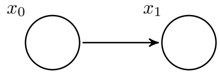
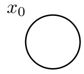
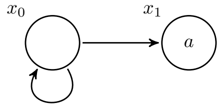
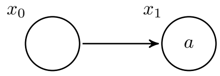
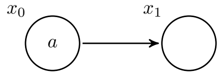
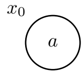
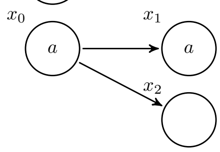
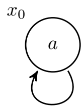
    Note that the other three options formalize one of the two parts of the sentence, but not the whole sentence.

21. In a world x_i of a Kripke model, one can consider three formulas: a, $\Diamond a$ and $\Box a$. These all can be either true or false, which gives eight possible combinations. Which of these combinations is not possible?

    (a) $x_i \nVdash a$, $x_i \Vdash \Diamond a$, $x_i \Vdash \Box a$
    (b) $x_i \Vdash a$, $x_i \nVdash \Diamond a$, $x_i \Vdash \Box a$
    (c) $x_i \nVdash a$, $x_i \nVdash \Diamond a$, $x_i \Vdash \Box a$
    (d) is correct (d) All of the above combinations are possible.

    Answer (d) is correct.

All eight possibilities are possible, as can be seen in the table below:

$x_0 \Vdash a$	$x_0 \Vdash \Diamond a$	$x_0 \Vdash \Box a$	Kripke model
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

22. Explain why in LTL the formulas $\mathcal{GF}a$ and $\mathcal{FG}a$ mean something different. You may do this informally using words, or you may define an LTL Kripke model (by defining the valuation $V(x_i)$ for all $i \in \mathbb{N}$) in which one of these formulas is true, but the other is not.

The formula $\mathcal{GF}a$ means: it is always the case that a will become true. This is actually the same as a is infinitely often true.

The formula $\mathcal{FG}a$ means: there will be a moment that a will globally be true. This is actually the same as, from a certain moment on, a will always be true.

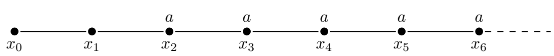
In particular, $\mathcal{FG}a$ implies $\mathcal{GF}a$. However, $\mathcal{GF}a$ does not imply $\mathcal{FG}a$.

This can be seen from the following models.

In the following model, where

$$V(x_i) = \begin{cases} \emptyset & \text{if } i = 0 \text{ or } i = 1 \\ \{a\} & \text{if } i \geq 2 \end{cases}$$

both $\mathcal{GF}a$ and $\mathcal{FG}a$ are true:



In the following model, where

$$V(x_i) = \begin{cases} \emptyset & \text{if } i = 0 \text{ or } i \text{ is odd} \\ \{a\} & \text{if } i \geq 2 \text{ and } i \text{ is even} \end{cases}$$

the formula $\mathcal{G}\mathcal{F}a$ holds, but the formula $\mathcal{F}\mathcal{G}a$ does not: