Formal Reasoning
Exercises

Engelbert Hubbers and Freek Wiedijk
Translated Fall of 2016 by Kelley van Evert

This file will be updated every week with the latest solutions!

August 26, 2025

Contents

Chapter 1

Propositional logic

Exercise 1.A
Form sentences in our formal language that correspond to the following English sentences:

(i) It is neither raining, nor is the sun shining.

> Note that if in the exercise it is not explicitly stated that you should use the official notation for formulas, you may omit superfluous parentheses.

Solution: $\neg R \wedge \neg S$: It is not raining and the sun is not shining.

(ii) The sun shines unless it rains.
Solution: The meaning of the word 'unless' can be a bit unclear. Often it is read as 'The sun shines, but not when it is raining,' but other interpretations can be given. Specifically, the following interpretations are not logically equivalent (see Definition 1.13):

- $R \to \neg S$: When it rains, the sun does not shine. Notice that nothing is said about the situation in which the sun does shine, but does this fit with the meaning of 'unless'?
- $\neg R \to S$: When it is not raining, the sun shines.
- $S \leftrightarrow \neg R$: The sun shines if and only if it is not raining.
- $\neg S \to R$: If the sun doesn't shine, it must rain.
- $S \to \neg R$: When the sun shines, it cannot rain.

Which one of these interpretations do you think matches the original sentence the best?

(iii) Either the sun shines, or it rains. (But not both simultaneously.)
Solution: Here, too, are multiple possibilities, though in this case they are logically equivalent. The first resembles the original English sentence more, though.

- $(S \vee R) \wedge \neg (S \wedge R)$: the sun shines or it rains, and not both at the same time,
- $(S \wedge \neg R) \vee (\neg S \wedge R)$: either the sun shines and it doesn't rain, or the sun doesn't shine and it rains.

Notice that the formulas $S \leftrightarrow \neg R$ and $\neg (S \leftrightarrow R)$ have the same truth table as these solutions. But because the form of these formulas doesn't resemble the original English sentence much, we think these formulas are less good translations.
Simply $S \vee R$ is wrong, because it doesn't account for the 'but not both simultaneously' part.

(iv) There is only a rainbow if the sun is shining and it is raining.
Solution: This solution seems best: $RB \to (S \wedge R)$: if there is a rainbow, then the sun must be shining and it must be raining. (Because only when the sun is shining and it is raining, there can be a rainbow.)

The following solution seems less good, but might be defensible: $\text{RB} \leftrightarrow (\text{S} \wedge \text{R})$: there is a rainbow if and only if the sun shines and it is raining. In this translation, the word 'only' is somewhat unusually interpreted.

Furthermore, it is clear that $(\text{S} \wedge \text{R}) \rightarrow \text{RB}$ is not a good translation of the sentence. Because this would correspond to 'There is a rainbow if the sun is shining and it is raining,' leaving away the 'only' as if it meant nothing special.

(v) If I'm outside, I get wet, but only if it rains.

Solution: There are many acceptable solutions in this case. Although we know that typically sentences like 'A, but B' are translated to '$A \wedge B$,' and usually sentences like 'A only if B' are translated to '$A \rightarrow B$', it is quite unclear how 'A, but only if B' should be translated. We could interpret 'A, but only if B' for instance as:

i) 'A if (also) B', or

ii) 'A only if B'.

Furthermore, in the original sentence it is unclear where the implicit parentheses are located. So this gives another two possibilities of interpreting the sentence:

I) ((If I'm outside, I get wet), but only if it rains.)

II) (If I'm outside, (I get wet, but only if it rains).)

If we combine these options, we get the following interpretations of the sentence and the corresponding formulas:

i+I) "I get wet when I'm outside, if it (also) rains." $\text{R} \rightarrow (\text{Out} \rightarrow \text{W})$ or "I get wet when I'm outside and it (also) rains." $(\text{Out} \wedge \text{R}) \rightarrow \text{W}$

ii+I) "If I get wet wet when I'm outside, then it must be raining." $(\text{Out} \rightarrow \text{W}) \rightarrow \text{R}$

> Note that formulas of the form $(A \rightarrow B) \rightarrow C$ usually have a more subtle meaning than you expect and should only be used with great care.

i+II) "If I'm outside, then I get wet if it (also) rains." $\text{Out} \rightarrow (\text{R} \rightarrow \text{W})$

ii+II) "If I'm outside, then I get wet only if it rains." $\text{Out} \rightarrow (\text{W} \rightarrow \text{R})$

Note that the bi-implication $(\text{Out} \rightarrow \text{W}) \leftrightarrow \text{R}$ is not in this list although both its parts $(\text{Out} \rightarrow \text{W}) \rightarrow \text{R}$ and $\text{R} \rightarrow (\text{Out} \rightarrow \text{W})$ are. The reason we don't consider this a really good solution is that both directions of this bi-implication are accepted under different interpretations!

Exercise 1.B

Can you also express $f \leftrightarrow g$ using the other connectives? If so, show how.

Solution: The formula $f \leftrightarrow g$ can also be expressed as $(f \rightarrow g) \wedge (g \rightarrow f)$ or as $(f \vee g) \rightarrow (f \wedge g)$. Also $(f \wedge g) \vee (\neg f \wedge \neg g)$ is logically equivalent.

Exercise 1.C

Translate the following formal sentences into English:

(i) $\text{R} \leftrightarrow \text{S}$

Solution: It rains if and only if the sun shines. (Or you could say something as 'it rains exactly when the sun shines.') Note that the sentence 'If it rains then the sun shines and if the sun shines, it rains' is perfectly well a translation of the formula $(\text{R} \rightarrow \text{S}) \wedge (\text{S} \rightarrow \text{R})$, which is in turn logically equivalent to $\text{R} \leftrightarrow \text{S}$, but nevertheless it is not so good of a translation of $\text{R} \leftrightarrow \text{S}$.

(ii) $\text{RB} \rightarrow (\text{R} \wedge \text{S})$

Solution: If there is a rainbow, then it rains and the sun shines.

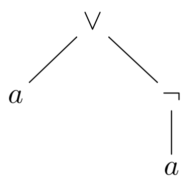
(iii) Out → ¬ In
Solution: If I'm outside, I'm not inside.
(iv) Out ∨ In
Solution: I'm outside or inside (or both).

Exercise 1.D
Draw the parse trees and give the truth tables for:
(i) $a \vee \neg a$

> Make sure that your truth tables are created properly:
>
> - Always start with a column for each atomic proposition.
> - Each connective in the formula should have its own column.
> - Columns are separated by lines.
> - Rows are also separated by lines, but if your sheet has already horizontal lines, you may use these existing lines.
> - There should be a row for each possible valuation.
> - The rows should be ordered in such a way that if you take the 0's and 1's to be bits in a bit string, the first row coincides with the value 0 and the last row coincides with the value $2^n - 1$, where n is the number of atomic propositions.
> - If it is easier for you to repeat columns, this is allowed, but this is not obligatory.
> - Also not obligatory, but you may add parentheses to make the structure of the formula easier to parse (read correctly).

Solution:

The parse tree:

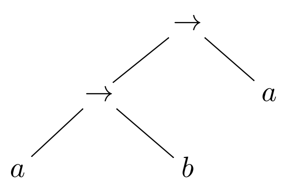


And the truth table:

a	$\neg a$	$a \vee \neg a$
0	1	1
1	0	1

(ii) $(a \to b) \to a$
Solution:

The parse tree:

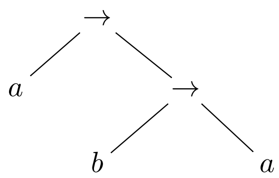


And the truth table:

a	b	$a \to b$	$(a \to b) \to a$
0	0	1	0
0	1	1	0
1	0	0	1
1	1	1	1

(iii) $a \to (b \to a)$
Solution:

The parse tree:

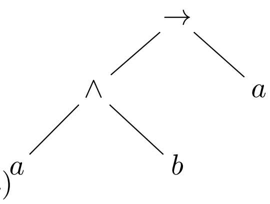


And the truth table:

a	b	$b \to a$	$a \to (b \to a)$
0	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

(iv) $a \land b \to a$
Solution:

The parse tree:

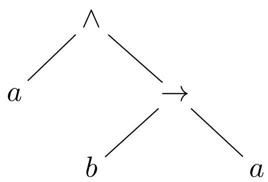


And the truth table:

a	b	$(a \land b)$	$(a \land b) \to a$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

(v) $a \land (b \to a)$
Solution:

The parse tree:

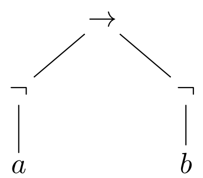


And the truth table:

a	b	$b \to a$	$a \land (b \to a)$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	1	1

(vi) $\neg a \to \neg b$
Solution:

The parse tree:



And the truth table:

a	b	$\neg a$	$\neg b$	$\neg a \to \neg b$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	1
1	1	0	0	1

Exercise 1.E
Which of the following propositions are logically true?

> You can find out whether a proposition is true by writing out its truth table. If the proposition's column is filled only with 1's, then the formula is logically true. If not, then the formula is not logically true.
> *Do not only write out the truth table as your answer. Always make sure to directly answer what was asked, by explicitly indicating which property of the truth table you have used to derive this conclusion!*

(i) $a \lor \neg a$

Solution: From the truth table of Exercise 1.D (i) you can conclude that $a \vee \neg a$ is logically true, because the last column contains only 1's.

(ii) $a \to (a \to a)$

Solution: Logically true. Note how the last column of its truth table is filled only with 1's:

a	$a \to a$	$a \to (a \to a)$
0	1	1
1	1	1

(iii) $a \to a$

Solution: Logically true. Note how the second column of the truth table above contains only 1's.

(iv) $(a \to b) \to a$

Solution: From the 0's in the corresponding truth table of Exercise 1.D (ii) you can conclude that $(a \to b) \to a$ is not logically true. Note that having only one 0 already implies that the formula is not logically true.

(v) $a \to (b \to a)$

Solution: From the truth table of Exercise 1.D (iii) you can conclude that $a \to (b \to a)$ is logically true, because the last column contains only 1's.

(vi) $a \wedge b \to a$

Solution: From the truth table of Exercise 1.D (iv) you can conclude that $a \wedge b \to a$ is logically true, because the last column contains only 1's.

(vii) $a \vee (b \to a)$

Solution: Not logically true. This follows from the 0 on the second row of the last column of its truth table:

a	b	$b \to a$	$a \vee (b \to a)$
0	0	1	1
0	1	0	0
1	0	1	1
1	1	1	1

(viii) $a \vee b \to a$

Solution: Not logically true. Again, this follows from the 0 on the second row of the last column of its truth table:

a	b	$a \vee b$	$(a \vee b) \to a$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	1	1

Exercise 1.F

Let f and g be *arbitrary* propositions. Find out whether the following statements hold. Explain your answers.

Note that all these statement are of the form 'if. . . , then. . . '.

- If the statement holds, you should give a general explanation why. In this explanation you may not assume anything about the structure of the arbitrary formulas f and g. In particular you don't know which atomic propositions are used within these formulas, so you don't know how many rows the corresponding truth tables have, so you cannot *write out* these truth tables. But you can *reason about* these truth tables!

- A proof for an 'if…, then…' statement works by first assuming the 'if' part, and then showing that the 'then' part must necessarily hold.

- If the statement does not hold, you can simply give a counterexample by choosing specific formulas f and g. In this case it is also possible to give a general explanation why the statement cannot be true, but that is usually much more difficult to formulate than giving an explanation about a specific counterexample.
  - To give a counterexample, you should pick some specific f and g in such a way that the 'if' part holds for these f and g, but the 'then' part fails to hold.

(i) If $\models f$ and $\models g$, then $\models f \wedge g$.
Solution: True.
Assume that $\models f$ and $\models g$. Which means that for every model v it holds that $v(f) = 1$ and $v(g) = 1$. Then for any such model v, you can calculate $v(f \wedge g)$ according to the truth table for $\wedge$, and you will find that $v(f \wedge g) = 1$. So apparently, given our assumptions, we have that for every model v, $v(f \wedge g) = 1$. Which then means that, by definition, $\models f \wedge g$. So the statement is true.

(ii) If not $\models f$, then $\models \neg f$.
Solution: Not true.
To find a counterexample for which we have 'not $\models f$,' we have to pick an f for which at least one model v_1 has $v_1(f) = 0$. For this v_1 it then automatically holds that $v_1(\neg f) = 1$. But to be sure that '$\models \neg f$' does not hold, we also need a model v_2 for which $v_2(\neg f) = 0$. This can only be the case if $v_2(f) = 1$. So, we can only find a counterexample if f is true in certain models and false in others. Luckily, there are plenty such f. For example, simply take $f = a$. Then we have 'not $\models a$', and yet the conclusion '$\models \neg a$' does not hold. So, we have found a counterexample, proving that the statement in question is not true.

(iii) If $\models f$ or $\models g$, then $\models f \vee g$.
Solution: True.
Assume that $\models f$ or $\models g$. So, we actually have three distinct cases, and for each of these cases we have to prove the same conclusion, namely $\models f \vee g$.

(1) $\models f$ is true. This means that for every v we have $v(f) = 1$. But for each such v, we then also have $v(f \vee g) = 1$, because the truth of f is enough to cause the truth of the composition $f \vee g$. So then we have $\models f \vee g$.

(2) $\models g$ is true. This case is exactly the same as the previous case (but the other way around).

(3) Both $\models f$ and $\models g$ are true. Then in particular, we have $\models f$, and in the case where we prove $\models f \vee g$ from $\models f$ we of course never relied on whether $\models g$ holds or does not hold. So we can use the same proof again.

So, in all three cases, the conclusion holds. So, the conclusion holds under the composite assumption '$\models f$ or $\models g$'. So, the statement holds.

(iv) If (if $\models f$, then $\models g$), then $\models f \rightarrow g$.
Solution: Not true.
So, we need a counterexample for which '(if $\models f$, then $\models g$)' holds. This sub-statement is again of the 'if…, then…' form and can thus be made true in two ways:

(1) By the falsity of $\models f$.

(2) By the truth of $\models f$ as well as $\models g$.

If in one of these situations we can show that $\models f \to g$ does not hold, we have found a counterexample.
Let us take $f = a$ and $g = b$. Now if we look at the models of $a \to b$, we see that there are four such models: $v_1(a) = 0$ and $v_1(b) = 0$, $v_2(a) = 0$ and $v_2(b) = 1$, $v_3(a) = 1$ and $v_3(b) = 0$, $v_4(a) = 1$ and $v_4(b) = 1$.

In a truth table:

model	a	b	$a \to b$
v_1	0	0	1
v_2	0	1	1
v_3	1	0	0
v_4	1	1	1

But because for example $v_1(a) = 0$, we immediately have that $\models a$ does not hold. So, for this specific choice for f and g we have '(if $\models f$, then $\models g$)'. However, if we compute $v_3(a \to b)$, we find $v_3(a \to b) = 0$. Which means that $\models f \to g$ does not hold. And thus we have indeed found a counterexample. So, the statement does not hold.

(v) If $\models \neg f$, then not $\models f$.
Solution: True.
This time over, we will give the proof without even using the word 'model', but by describing what happens in the truth tables, without writing them out in full. Because we have to the prove the statement for every f, you actually can't even write out the full truth table, because you don't know what f might look like. This doesn't matter, however, because you can reason about the 1's and 0's in the tables, and that's enough. (If the statement holds, of course.)
Assume that $\models \neg f$. Then the column of $\neg f$ contains only 1's. So, the column for f must show only 0's, and thus by definition $\models f$ does not hold. And thus, the statement holds.

(vi) If $\models f \vee g$, then $\models f$ or $\models g$.
Solution: Not true.
We will try to find a counterexample such that $\models f \vee g$, but not $\models f$ and also not $\models g$. Take $f = a$ and $g = \neg a$. Then there are only two different models: v_1 with $v_1(a) = 0$ and v_2 with $v_2(a) = 1$. It follows immediately that $v_1(\neg a) = 1$ and $v_2(\neg a) = 0$. It follows from $v_1(a) = 0$ that $\models a$ does not hold. And from $v_2(\neg a) = 0$ it also follows that $\models \neg a$ does not hold. However, $v_1(a \vee \neg a) = 1$ and $v_2(a \vee \neg a) = 1$, so we do have $\models f \vee g$. So, this is a valid counterexample. So, the statement does not hold.

(vii) If $\models f \to g$, then (if $\models f$, then $\models g$).
Solution: True.
Assume $\models f \to g$. Then for every model v we have $v(f \to g) = 1$. By looking at the way that $\to$ works, we see that there are two ways in which $v(f \to g) = 1$ can happen:

(1) $v(f) = 0$ (and it doesn't matter what $v(g)$ is)

(2) $v(f) = 1$ and $v(g) = 1$.

For both cases, we must show that 'if $\models f$, then $\models g$.'

(1) Suppose that we have a model v such that $v(f) = 0$. Then that means immediately that $\models f$ cannot be true, and thus the statement 'if $\models f$, then $\models g$' holds.

(2) So now we only have to look at the situation in which all models give $v(f) = 1$. But then, it follows from the case distinction that we made, that $v(g) = 1$ holds for all models. And thus we have 'if $\models f$, then $\models g$.'

Because in both cases, the conclusion 'if $\models f$, then $\models g$' holds, the original statement is true.

(viii) If $\models f \leftrightarrow g$, then ($\models f$ if and only if $\models g$).
Solution: True.
A proof with truth tables again.
Assume that $\models f \leftrightarrow g$. This means that in the truth table, in every row, the value in

the column of f and the value in the column of g are the same. (Either they are both 0, or both 1.) If we assume furthermore that $\models f$, then f has only 1's in its column. But then, the same holds for g because their values were the same. And thus we can conclude that $\models g$. This last part works equally well the other way around: if $\models g$ is true, then g has all 1's in its column, and thus f has all 1's too, and thus $\models f$ is true. So, the whole statement is true.

(ix) If ($\models f$ if and only if $\models g$), then $\models f \leftrightarrow g$.
Solution: Not true.
Take $f = a$ and $g = b$. We then have four different models again, as in item (iv). In particular, we have $v_1(a) = 0$ and $v_1(b) = 0$. So, $\models a$ does not hold, and also $\models b$ does not hold. But, we do have the truth of the statement '$\models f$ if and only if $\models g$'! When we look at v_3, we see that $v_3(a \leftrightarrow b) = 0$, and thus $\models f \leftrightarrow g$ is false. So, we have found a counterexample. So, the statement is false.

Exercise 1.G
For each of the following couples of propositions, show that they are logically equivalent to each other.

Solution: To see whether two propositions are logically equivalent, we can inspect their truth tables. The propositions are logically equivalent in the case that their columns are exactly the same.

(i) $(a \wedge b) \wedge c$ and $a \wedge (b \wedge c)$
Solution: These propositions are logically equivalent because the last two columns are exactly the same:

a	b	c	$a \wedge b$	$b \wedge c$	$(a \wedge b) \wedge c$	$a \wedge (b \wedge c)$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

(ii) $(a \vee b) \vee c$ and $a \vee (b \vee c)$
Solution: These propositions are logically equivalent too, because the last two columns are exactly the same:

a	b	c	$a \vee b$	$b \vee c$	$(a \vee b) \vee c$	$a \vee (b \vee c)$
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	0	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Exercise 1.H
Let f and g be propositions. Is the following statement true? $f \equiv g$ if and only if $\models f \leftrightarrow g$.

Solution: Let's split the statement up in two parts:

1. First we show that 'If $f \equiv g$ then $\models f \leftrightarrow g$' holds. If we assume that $f \equiv g$ holds, then f and g have identical columns. But then, the column of $f \leftrightarrow g$ has only 1's in it, and so $\models f \leftrightarrow g$ is true.

2. Then we show that 'If $\models f \leftrightarrow g$, then $f \equiv g$' holds. If we assume that $\models f \leftrightarrow g$ is true. Then the column of $f \leftrightarrow g$ is filled with only 1's. But then that means that, for every model, the truth value for f is the same as the truth value for g. And that is exactly what $f \equiv g$ expresses, so we have the truth of $f \equiv g$.

Both parts hold, and thus the original statement holds as well.

Exercise 1.I
Are the following statements true?

Solution: Like before, we can easily solve this exercise by writing out the truth tables for the propositions involved, and then checking the definition of 'logical entailment.'

(i) $a \wedge b \models a$
Solution: $a \wedge b \models a$ is true, because in every model where $a \wedge b$ holds, a holds as well.

(ii) $a \vee b \models a$
Solution: $a \vee b \models a$ is not true; the statement says that in every model in which $a \vee b$ is true, also a is true. But this is not the case, and we can find models in which $a \vee b$ but not a. For example, take the model v with $v(a) = 0$ and $v(b) = 1$.

(iii) $a \models a \vee b$
Solution: $a \models a \vee b$ is true: if a is true in a model, then $a \vee b$ must always also be true in that model.

(iv) $a \wedge \neg a \models b$
Solution: $a \wedge \neg a \models b$ is true: if $a \wedge \neg a$ is true in a model, then b must necessarily also be true in that model. This seems like a weird and untrue statement, for example because $a \wedge \neg a$ is never true in any model. However, the 'if..., then...' construction makes the statement true, exactly because of the fact that the 'if' part is never true. It is what is called a 'vacuous truth.' You can also look at it the other way around: if the statement $a \wedge \neg a \models b$ would not be true, then you must be able to find a model in which b is not true, and $a \wedge \neg a$ is. And that, of course, would be impossible.

Chapter 2

Predicate logic

Exercise 2.A
Give two possible translations for the following sentence.

Sharon loves Maud; a nice man loves this intelligent character.

Solution: This can be a complicated exercise, if you are willing to bend your mind around the (somewhat forced) ambiguities in this English sentence. It isn't entirely clear what is meant by *this intelligent character*. (And also not entirely clear what is meant by *a nice man* ....) Which is to say, you won't be able to translate the sentence without giving your own interpretation of its meaning. Though of course there is a difference between an acceptable interpretation and a simply wrong one: by which we mean to say, your interpretation should be *defensible*. As far as we see (or intended) it, the possible ambiguities are as follows:

- Interpreting *a nice man loves ...* as either:
  1. *there is some nice man who loves ...*, or as
  2. *a nice man would love ...* (which is the same as saying *any nice man would love ...*).
- Interpreting *loves this intelligent character* as either:
  3. *loves Maud*, or
  4. *loves Sharon*, or
  5. *loves Maud, who is intelligent*, or
  6. *loves Sharon, who is intelligent*, or
  7. *loves Maud's intelligent character* (which would translate to *loves any woman who is as intelligent as Maud*), or
  8. *loves Sharon's intelligent character* (which would translate to *loves any woman who is as intelligent as Sharon*), or
  9. *loves this intelligent character* (which would translate to *loves any woman who is intelligent*), or

...leading to a whole range of possible translations, of which we will here list only a few:

(i) Interpreting as per (2) and (9):

$$L(s, m) \wedge \forall x \in M\,[N(x) \rightarrow \forall w \in W\,[I(w) \rightarrow L(x, w)]]$$

(ii) Interpreting as per (2) and (6):

$$L(s, m) \wedge \forall x \in M\,[N(x) \rightarrow (L(x, s) \wedge I(s))]$$

(iii) Interpreting as per (1) and (5):

$$L(s,m) \wedge \exists x \in M \left[N(x) \wedge (L(x,m) \wedge I(m))\right]$$

(iv) Interpreting as per (1) and (9):

$$L(s,m) \wedge \exists x \in M \left[N(x) \wedge \forall w \in W \left[I(w) \rightarrow L(x,w)\right]\right]$$

(v) Interpreting as per (2) and (8):

$$L(s,m) \wedge \forall x \in M \left[N(x) \rightarrow \forall w \in W \left[(I(w) \leftrightarrow I(s)) \rightarrow L(x,w)\right]\right]$$

Note that this indeed does not say anything about the intelligence of Sharon, but just that any man would love any woman *as intelligent as* Sharon. (Which would be up to the nice man's opinion, probably.)

(vi) Interpreting as per (1) and (3):

$$L(s,m) \wedge \exists x \in M \left[N(x) \wedge L(x,m)\right]$$

Exercise 2.B

Translate the following sentences to English.

Solution: This kind of translation exercises are best done in two steps. First, you quite literally translate the sentence to English, but then the English sentence will probably be a bit ugly. So then secondly, you play around with the English connectives to make the sentence more readable. But do make sure that this second English sentence still means the same as the original one!

(i) $\exists x \in M \Big[T(x) \wedge \exists w \in W \left[B(w) \wedge I(w) \wedge L(x,w)\right]\Big]$

Solution:

- There is a tall man and there is a beautiful intelligent woman, and this man loves this woman.
- There is a tall man and there is a beautiful intelligent woman, whom is loved by this man.
- There is a tall man who loves this beautiful and intelligent woman. (*Note the use of "this" to specify that this is a specific woman we are talking about, and not any/every beautiful and intelligent woman.*)

(ii) $\exists x \in M \Big[T(x) \wedge \exists w \in W \left[B(w) \wedge \neg I(w) \wedge L(x,w)\right] \wedge \exists w' \in W[I(w') \wedge L(w',x)]\Big]$

Solution:

- There is a tall man, and there is a beautiful but not intelligent woman whom he loves, and there is another woman who is intelligent and loves him.
- There is a tall man who loves this beautiful but unintelligent woman, and who is loved back by another woman, who is intelligent.

  Note that only saying $\exists w \in W$ and $\exists w' \in W$ doesn't exclude the possibility that w and w' are actually the very same person. However, we know that they are distinct people, because one of them is said to be intelligent, and the other not.

Exercise 2.C

Formalize the following sentence.

Sharon is beautiful; there is a guy who feels good about himself whom she loves.

Here, we will treat feeling good about oneself as being in love with oneself.

Solution: Using the dictionary (see page 13 and page 14) this can be formalized as:

$$B(s) \wedge \exists x \in M \ \left[L(x,x) \wedge L(s,x)\right]$$

Exercise 2.D
Formalize the following sentences:

(i) *For every two persons we have: the first one loves the second one only if the first one feels good about him- or herself.*

Solution: Note that we have chosen x to represent 'the first one' and y to represent 'the second one' within this exercise. We could have used different variables, but this choice seems more reasonable than taking, for instance, w for 'the first one' and v for 'the second one'. Note also that 'only if' doesn't mean the same thing as 'if and only if'. So, $\forall x, y \in W \cup M[L(x,y) \rightarrow L(x,x)]$ is correct, but $\forall x, y \in W \cup M[L(x,y) \leftrightarrow L(x,x)]$ is not. (If you want to stress that it is really about 'two distinct' persons, you should add this requirement, for instance like this: $\forall x, y \in W \cup M[\neg(x = y) \wedge L(x,y) \rightarrow L(x,x)]$. But this construction with equalities will only be explained in Section 2.4, so don't worry about it now.)

(ii) *For every two persons we have: the first one loves the second one if this second person feels good about him- or herself.*

Solution: $\forall x, y \in W \cup M[L(y,y) \rightarrow L(x,y)]$

(iii) *For every two persons we have: the first one loves the second one exactly in the case that the second one feels good about him- or herself.*

Solution: $\forall x, y \in W \cup M[L(x,y) \leftrightarrow L(y,y)]$ or $\forall x, y \in W \cup M[L(y,y) \leftrightarrow L(x,y)]$, but because of the matching word order in the sentence and the first option, we consider that one slightly better.

(iv) *There is somebody who loves everyone.*

Solution: $\exists x \in W \cup M\,[\forall y \in W \cup M[L(x,y)]]$

Exercise 2.E

(i) Verify that F_2 does not hold in M_1 under the interpretation I_1. But does F_1 hold?
Solution: Recall the definitions of F_1 and F_2:

$$\begin{aligned} F_1 &:= \forall x \in D \exists y \in D\; K(x,y) \\ F_2 &:= \exists x \in D \forall y \in D\; K(x,y) \end{aligned}$$

So, independent of the structure and interpretation used, F_1 says that for every object x in the structure, there is some object y such that the relation $K(x,y)$ holds. And F_2 says that there is some object x in the structure, such that the relation $K(x,y)$ holds for every y in the structure. And remember: y may also be the same object as x.
So now let's first turn to the truth of F_2 within the structure M_1 and under the interpretation of I_1:

D	all students in the lecture hall
$K(x,y)$	x has a student number lower than y

We have been told already that F_2 will not hold. So, we need to prove that F_2 does indeed not hold. Because F_2 is of the form $\exists x \in D[f]$, such a proof amounts to showing that for every x, the statement f is not true. And because f has the form $\forall y \in D[g]$, proving that f is not true amounts to demonstrating that there is some y such that g is not true. Putting these two together, we have to prove that for all x, there is at least one y, such that $K(x,y)$ is not true.
Because there are but a finite number of students in the lecture hall (let's say n students), we can sort them by their unique student numbers and give them an index ranging from 1 to n. So let's name the students s_1, s_2 up to s_n, where the student s_1 has the lowest student number, and the student s_n has the highest.

To turn things around, let's see if we can prove F_2. If F_2 were to be true, you'd be able to pick out some student who has the lowest student number of anyone in the lecture hall. Obviously, the only contestant would be s_1, right? So, we take $x = s_1$ in our first step to proving F_2 true. And now it remains to be proved that

$$\forall y \in D\ K(s_1, y) .$$

Of course this indeed holds for many students y, but it does not hold for *all students* in the lecture hall, because specifically s_1 is also one of these students, and he/she doesn't have a lower student number than him/herself.
Because s_1 is our only real possibility, but we don't have $K(s_1, s_1)$, we have proved that F_2 is not true (in M_1 under I_1).
What a long proof! Can't we write this down shorter? Yes, we can, for example:

> Sort the students by their unique student number in ascending order: s_1 to up s_n. (So, s_n has the highest student number and s_1 the lowest.) Suppose that F_2 is true. Then there are two possible cases for x:
>
> - If $x = s_i$, for $i \in \{2, \dots n\}$, then $K(s_i, s_1)$ does not hold, so certainly not $\forall y \in D\ K(x, y)$. And so F_2 does not hold for $x = s_i$.
> - If $x = s_1$, then $K(s_1, s_1)$ does not hold, so indeed $\forall y \in D\ K(x, y)$ does not hold either, because $s_1 \in D$ too. And so F_2 does not hold for $x = s_1$.
>
> We have seen that F_2 does not hold for any x, so F_2 does not hold in this structure.

Now the question whether F_1 holds in this structure. In other words: is it true that for every student x we can find some student y with a higher student number? Well, obviously not. Because F_1 is of the shape $\forall x \in D\ [\exists y \in D\ K(x, y)]$, we are satisfied with proving the formula as soon as we can pick out some x such that no y exists for which $K(x, y)$.
So we pick $x = s_n$, the student with the highest student number, and indeed there is no y with a higher student number than x. Thus, we have completed the proof, that F_1 indeed does not hold in structure M_1 under the interpretation I_1.

(ii) Verify that F_1 holds in M_1 under the interpretation I_2. Does F_2 hold as well?
Solution: Recall interpretation I_2:

D	all students in the lecture hall
$K(x, y)$	x isn't older than y

We have to demonstrate the truth of F_1 in the interpretation I_2. This means, that for every x, we must show that we can find a y, such that $K(x, y)$. In other words: we must give a method, or algorithm, by which, for any x, we can choose a y in such a way that $K(x, y)$ holds.
Which is in this case surprisingly simple: choose y to be $y = x$ itself. Then indeed $K(x, y)$, because one is never older than oneself. So indeed, F_1 holds under interpretation I_2.
In a similar way we can show that F_2 holds in structure M_1 under the interpretation I_2. To do that, we must pick a suitable x. Before, we have ordered students according to their student number. Now, let's order the students according to their age. But note that, unlike student numbers, the ages of students need not be unique. (Even if you'd measure age in seconds or an even smaller timescale, instead of years.) So, we are not able to speak of the *oldest* or the *youngest* student, because there may be multiple such students with exactly the same age. But even though we may not have a youngest student, we do have a subset of youngest students, and that's enough for our current purposes. Let's take x to be any of these youngest students (if there are multiple, or else

just the single youngest person). Then indeed for all students y, it holds that $K(x, y)$, or, x is not older than y. So we have proved F_2 in M_1 under I_2.

(iii) Check whether F_1 in M_1 is true under the interpretation of I_3, by looking around in class. And check whether F_2 is true or not, as well.

Solution: Recall I_3:

D	all students in the lecture hall
$K(x, y)$	x is sitting next to y

At the time of writing this document, it can't be stated whether F_1 is true or not, because it depends on the specific situation in the lecture hall. If someone is sitting alone from the rest, he/she would be a counterexample, proving the falsity of F_1. Though if everyone is sitting next to someone else, then you have a method by which for every x you can pick a y such that $K(x, y)$ holds. Namely, pick one of his/her neighbors.

However, F_2 is never true in M_1 under interpretation I_3. Because it is never the case that some student is sitting next to all others. Even if you'd arrange the chairs in such a way that all other students sit next to the person in question, this person would not be sitting next to him- or herself. So, F_2 is definitely false under interpretation I_3.

Exercise 2.F

Verify that G_2 is indeed true in structure M_4 under the interpretation I_8, but not in structure M_3 under the interpretation I_7. Stated differently: verify that $((\mathbb{Q}, <), I_8) \models G_2$ and verify that $((\mathbb{N}, <), I_7) \not\models G_2$.

Solution:

(i) Interpretation I_7 gives G_2 the meaning that for every $x \in \mathbb{N}$ there is a $y \in \mathbb{N}$ such that $y < x$. But this is not true. Counterexample: take $x = 0$, and then there will be no y such $y < 0$.

(ii) Interpretation I_8 gives G_2 the meaning that for every $x \in \mathbb{Q}$ there is a $y \in \mathbb{Q}$ such that $y < x$. This time the statement is true, because one can always take for example $y = x - 1$ in $\mathbb{Q}$.

Exercise 2.G

Define the interpretation I_9 as:

D	$\mathbb{N}$
$K(x, y)$	$x = 2 \cdot y$

Are the formulas G_1 and/or G_2 true under this interpretation?

Solution:

(i) Under interpretation I_9 the formulas G_1 states that for every $x \in \mathbb{N}$ there is a $y \in \mathbb{N}$ such that $x = 2 \cdot y$. Not true: take for example $x = 3$, and then there will be no such y.

(ii) Under interpretation I_9 the formulas G_2 states that for every $x \in \mathbb{N}$ there is a $y \in \mathbb{N}$ such that $y = 2 \cdot x$. True: for any x we can take $y = 2 \cdot x$.

Exercise 2.H

Define the interpretation I_{10} as:

D	$\mathbb{Q}$
$K(x, y)$	$x = 2 \cdot y$

Are the formulas G_1 and/or G_2 true under this interpretation?

Solution:

(i) Under the interpretation I_{10} the formula G_1 states that for every $x \in \mathbb{Q}$ there is a $y \in \mathbb{Q}$ such that $x = 2 \cdot y$. True: for any x we can take $y = \frac{x}{2}$.

(ii) Under the interpretation I_{10} the formula G_2 states that for every $x \in \mathbb{Q}$ there is a $y \in \mathbb{Q}$ such that $y = 2 \cdot x$. True: for any x we can take $y = 2 \cdot x$.

Exercise 2.I
We take as structure the countries of Europe, and the following interpretation I_{11}:

E	the set of countries of Europe
n	The Netherlands
g	Germany
i	Ireland
$B(x, y)$	x borders y
$T(x, y, z)$	x, y, and z share a tripoint (where the borders of all three countries meet)

(i) Formalize the sentence "The Netherlands and Germany share a tripoint."
Solution: This would be any of the equivalent formulas $\exists x \in E\ [T(n, g, x)]$, or $\exists x \in E\ [T(n, g, x)]$, or $\exists x \in E\ [T(x, g, n)]$, or $\exists y \in E\ [T(y, n, g)]$, etc.

(ii) Which of the following formulas are true in this structure and under this interpretation?

(1) $G_3 := \forall x \in E\ \exists y \in E\ [B(x, y)]$
Solution: G_3 states that every country x borders at least some country y. But this is not true, for example for island countries such as Iceland and Malta.

(2) $G_4 := \forall x, y \in E\ [(\exists z \in E\ T(x, y, z)) \to B(x, y)]$
Solution: G_4 states that for every two countries x and y, if they share a tripoint with some country z, then x and y border each other as well. True, if they share a tripoint, then in particular they indeed share a border.

(3) $G_5 := \forall x \in E\ [B(i, x) \to \exists y \in E\ [T(i, x, y)]]$.
Solution: G_5 states that if a country borders Ireland, then it shares a tripoint with Ireland as well. Not true: take the United Kingdom, which borders Ireland (via Northern-Ireland), but does not share a tripoint with Ireland. (However, if i would have been interpreted to mean Iceland, the G_5 would have been *vacuously true*, because $B(i, x)$ is never true for any x, and therefore the implication is true.)

Exercise 2.J
Find a structure M_5 and an interpretation I_{12} such that this formula holds:

$$(M_5, I_{12}) \models \forall x \in D\ \exists y \in E\ [R(x, y) \wedge \neg R(y, x) \wedge \neg R(y, y)]$$

Solution: Two possible solutions for M_5 and I_{12} are:

- The structure $(\mathbb{N}, <)$ with interpretation I_{12}

D	$\mathbb{N}$
E	$\mathbb{N}$
$R(x, y)$	$x < y$

(Verify that the formula holds by taking y to be $x + 1$.)

- The structure

Domain(s)	all people
Relation(s)	being a father of

with interpretation I_{12}

D	all people
E	all people
$R(x, y)$	y is the father of x

Verify that this would not work if $R(x, y)$ would be interpreted as x being y's father.

Note that the domains D and E must be related. We will not further discuss this in this course, but in the course "Logic and Applications" it will be explained that every relation has a *type*, for example $\mathbb{N} \to \mathbb{Q} \to \{0, 1\}$, meaning that x must be an element of $\mathbb{N}$ and y an element of $\mathbb{Q}$, and that the relation $R(x, y)$ then results in a 0 or a 1 depending on the interpretation. In this exercise, however, we write $R(x, y)$ as well as $R(y, x)$, meaning that then we must have $x \in D$, $x \in E$, $y \in D$, as well as $y \in E$! So, defining a structure in which D is the set of cars, E is the set of bicycles, and defining $R(x, y)$ to mean that 'x is faster than y' would be a bit weird because cars are not bicycles and neither the other way around. Though of course we could define the superset V to be the set of vehicles and let R be of type $V \to V \to \{0, 1\}$, and things would work correctly again. If you don't understand this remark, don't worry, it will be treated in full in the course "Logic and Applications."

Exercise 2.K

Consider the interpretation I_{14}:

H	domain of all human beings
$F(x)$	x is female
$P(x, y)$	x is parent of y
$M(x, y)$	x is married to y

Formalize the following sentences into formulas of predicate logic with equality:

(i) *Everyone has exactly one mother.*

Solution:

$$\begin{array}{llll} \forall x \in H & [& \exists y \in H & [\ F(y) \wedge P(y, x) \\ & & &] \\ & & \wedge & \\ & & \forall y, z \in H & [\ (F(y) \wedge P(y, x) \wedge F(z) \wedge P(z, x)) \to y = z \\ & & &] \\ &] & & \end{array}$$

In this formula, x is the "everyone", and y and z play the role of mother. The first y is used to state that indeed x has at least one mother, (namely y), and the second y, together with z, is used to state that x has at most one mother, by stating that for any two people who would both be x's mother, these people are the same person. Note: there is no obligation to use the same variable for the second y as we did for the first y; we could equally well have called this second y for example w. Also note the *scope* of x: in the second part (at most 1 mother) we must be able to refer back to x.

This alternative solution would have worked as well:

$$\begin{array}{llll} \forall x \in H & [\ \exists y \in H & [& F(y) \wedge P(y, x) \\ & & & \wedge \\ & & & \forall z \in H[(F(z) \wedge P(z, x)) \to y = z] \\ & &] & \\ &] & & \end{array}$$

Here, the y is bound only once, slightly more elegantly stating that this is the only mother that x has. (And directly stating that any mother z of x is indeed the same mother.)

(ii) *Everybody has exactly two grandmothers.*

Solution:

$$
\begin{array}{llll}
\forall x \in H & [\quad \exists y, z \in H & [& y \neq z \ \wedge \\
& & & \exists u \in H[F(y) \wedge P(y,u) \wedge P(u,x)] \wedge \\
& & & \exists u \in H[F(z) \wedge P(z,u) \wedge P(u,x)] \\
& &] & \\
& \quad \wedge & & \\
& \quad \forall y, z, v \in H & [& (\\
& & & y \neq z \ \wedge \\
& & & \exists u \in H[F(y) \wedge P(y,u) \wedge P(u,x)] \wedge \\
& & & \exists u \in H[F(z) \wedge P(z,u) \wedge P(u,x)] \wedge \\
& & & \exists u \in H[F(v) \wedge P(v,u) \wedge P(u,x)] \\
& & &) \\
& & & \rightarrow \\
& & & (v = y \vee v = z) \\
& &] & \\
&] & &
\end{array}
$$

Here again, x plays the role of "everybody", and now y and z play the role of the two distinct grandmothers, and u plays the role of the father and/or mother of x, simultaneously being the child of y, resp. z.

(iii) *Every married man has exactly one spouse.*

Solution: Note that a spouse can be both the husband or the wife. So in the given solution x must be a man, but y, z_1 and z_2 may be both male or female.

$$
\begin{array}{ll}
\forall x \in H & [\quad \neg F(x) \wedge \exists y \in H \ [M(x,y)] \\
& \quad \rightarrow \\
& \quad (\forall z_1, z_2 \in H \ [M(x,z_1) \wedge M(x,z_2) \rightarrow z_1 = z_2]) \\
&]
\end{array}
$$

Exercise 2.L

Use the interpretation I_{14} of Exercise 2.K to formalize the following properties.

(i) $C(x,y)$: x and y have had a child together.
Solution: Define

$$C(x,y) := x \neq y \wedge \exists z \in H[P(x,z) \wedge P(y,z)]$$

Then, x and y are indeed two distinct people, and there is a child z who has parents x and y. (If we don't include the $x \neq y$ the sentence is not an exact translation.)

(ii) $B(x,y)$: x is a brother of y (take care: refer also to the next item).
Solution: Define

$$B(x,y) := \neg F(x) \wedge x \neq y \wedge \exists q, r \in H[q \neq r \wedge P(q,x) \wedge P(q,y) \wedge P(r,x) \wedge P(r,y)]$$

So, x is a brother of y if x is male, and not the same person as y, and there are two distinct parents who are the parents of x as well as of y.

(iii) $S(x,y)$: x is a step-sister to y.
Solution: A first try:

$$
\begin{aligned}
S(x,y) := {} & F(x) \wedge x \neq y \wedge \\
& \exists o_1, o_2, o_3 \in H[o_1 \neq o_2 \wedge o_2 \neq o_3 \wedge o_3 \neq o_1 \wedge \\
& P(o_1,x) \wedge \neg P(o_1,y) \wedge P(o_2,x) \wedge P(o_2,y) \wedge \neg P(o_3,x) \wedge P(o_3,y)]
\end{aligned}
$$

So, x is a step-sister to y if x is female, and there are three distinct parents o_1, o_2, o_3, the first two of which are the parents of x, and the last two of which are the parents of y (so x and y share exactly one parent o_2.)
Unfortunately this is not a step-sister, but a half-sister! A step-sister is

> A daughter of one's step-parent by a marriage other than with one's own parent.

And a step-parent is defined as

> A person who is married to one's parent, but is not one's parent.

If we combine these two definitions we get the following formalization for x is a step-sister of y:

$$
\begin{aligned}
&S(x,y) := F(x) \wedge \\
&\quad \exists o_1 \in H[P(o_1,y) \wedge \neg P(o_1,x) \wedge \exists o_2 \in H[P(o_2,x) \wedge \neg P(o_2,y) \wedge M(o_2,o_1)]]
\end{aligned}
$$

This formula expresses that x is a step-sister of y if

- x is female,
- o_1 is a parent of y, but is not a parent of x,
- o_2 is a parent of x, but is not a parent of y,
- and o_2 is married to o_1.

Note that we didn't specify that $x \neq y$, but that follows automatically from $P(o_1,y) \wedge \neg P(o_1,x)$. And from $M(o_2,o_1)$ it follows that $o_2 \neq o_1$, because you cannot be married to your self. Similarly it follows that x, y, o_1 and o_2 are all different persons.

Translate the following formulas back to English.

(iv) $\exists x{\in}H \forall y{\in}H \; P(x,y)$. And is this true?
Solution: "There is a person who is everyone's parent." This is obviously not true.

(v)

$$
\forall z_1 \in H \forall z_2 \in H \; \left[\exists x{\in}H \exists y_1{\in}H \exists y_2{\in}H \; \left[P(x,y_1) \wedge P(y_1,z_1) \wedge P(x,y_2) \wedge P(y_2,z_2) \right] \rightarrow \neg\left(\exists w{\in}H \; [P(z_1,w) \wedge P(z_2,w)]\right) \right]
$$

And is this true?
Solution: "Every two people who share a common grandparent, do not share a child." This is also not true. Note that these two people z_1 and z_2 can be the same person! Furthermore, it is also not unlikely that there are cousins within a family who do share a child.

Exercise 2.M
Given the interpretation I_{15}:

D	$\mathbb{N}$
$A(x,y,z)$	$x + y = z$
$M(x,y,z)$	$x \cdot y = z$

Formalize the following:

Solution: In this exercise we will need formulas that formalize the numbers $x = 0$ and $x = 1$, so let's first solve these.

- We can capture the character of x being 0 neatly by stating that $a \cdot x = x$ for all a. So, define
$$x = 0 := \forall a \in D[M\,(a, x, x)]$$
and $x \neq 0 := \neg\,(x = 0)$.
- Similarly, for $x = 1$ we have the defining property that $x \cdot a = a$ holds for all a. So, define
$$x = 1 := \forall a \in D[M\,(x, a, a)]$$
and $x \neq 1 := \neg\,(x = 1)$.

(i) $x < y$.
Solution: $x < y$ holds precisely in the case that $x + r = y$ and $r \neq 0$.
So, define:
$$x < y := \exists r \in D[A\,(x, r, y) \wedge \neg\,(r = 0)]$$

(ii) $x \mid y$ (x divides y).
Solution: $x|y$ (x divides y) whenever there is some number z such that $x \cdot z = y$. So, define:
$$x|y := \exists z \in D[M\,(x, z, y)]$$

(iii) x is a prime number.
Solution: x is a prime number when x is not 1, and has no other factors other than 1 and itself. So, define:
$$x \text{ is a prime number} := x \neq 1 \wedge \neg\exists y, z \in D[M\,(y, z, x) \wedge y \neq x \wedge z \neq x \wedge y \neq 1 \wedge z \neq 1]$$
We could also simply use the $x|y$ which we already defined above.
$$x \text{ is a prime number} := x \neq 1 \wedge \forall y \in D[(y|x) \rightarrow (y = 1 \vee y = x)]$$

Chapter 3

Discrete mathematics

Exercise 3.A
Prove that in a tree, between any two points v and w, there is exactly one path that connects the two.

Solution: This exercise is almost fully based on Definition 3.5, which says that a tree is precisely a connected graph without cycles. We must prove:

- There is at least one path connecting v and w.
- There is at most one path connecting v and w.

The first of these two items follows directly from the definition of connectedness: any two points are connected by a path, and so too v and w.

The second is easily seen, but a bit harder to prove. A commonly used method, for proving that at most one object of a certain type exists, is to show that having two distinct objects of this type leads to a contradiction. So let us suppose we have two distinct paths connecting v to w:

$$v = x_0 \to x_1 \to x_2 \to \cdots \to x_{n-1} \to x_n = w$$

and

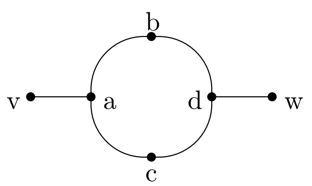
$$v = y_0 \to y_1 \to y_2 \to \cdots \to y_{m-1} \to y_m = w$$

What we will do, is prove that we can use these paths to construct a cycle, which then presents us with an obvious contradiction. The easy argument is that we now must have the following cycle:

$$v = x_0 \to x_1 \to x_2 \to \cdots \to x_{n-1} \to x_n = w = y_m \to y_{m-1} \to y_{m-2} \to \cdots \to y_1 \to y_0 = v$$

But this need not necessarily be a path, because paths may not traverse the same edge multiple times, which might be the case here. (Read the definition of a path carefully: it talks about a sequence of *distinct* edges.)

Consider for example:



Here, there are obviously two paths $v \to a \to b \to d \to w$ and $v \to a \to c \to d \to w$, connecting v and w, but simply connecting the two leads to:

$$v \to a \to b \to d \to w \to d \to c \to a \to v$$

in which not all edges are distinct: (v, a) and (d, w) both occur twice. Hence, it is not a valid path, and in particular not a valid cycle. However, such double occurrences can always be left out, yielding a valid cycle. In the example, that would then be:

$$a \to b \to d \to c \to a$$

And so, we have a contradiction nonetheless, concluding our proof.

Note how the cycle in the example doesn't even include the vertices v and w any more. A formal proof that such a cycle always exists, yielding a contradiction, is a bit too complicated and therefore we omit it, but the rough idea for constructing such a cycle is as follows:

- Remove the initial part shared by the two paths, so that they now start at a vertex v' and then immediately follow different edges. There are still two paths, but now they start at v' instead of v.
- Follow one of the two paths until you reach a vertex that is also on the other path. The two diverging paths will cross again at w if not earlier, so we know that such a vertex always exists. Call it w'. We now have two paths leading from v' to w': one via the path that we followed, and one via the other path.
- Because of the way we constructed these two new paths, we know that we can join them in sequence (reversing the second path), as per the original idea above, and indeed get a true path in which every edge is distinct.
- In particular, this path is also a cycle, by construction, and hence we have the contradiction we sought.

Exercise 3.B

A *bridge* in a graph G is an edge e for which: if you remove e from G, then the number of components of G increases. Prove that in a tree, every edge is a bridge.

Solution: Choose an arbitrary edge in a tree. Say this edge joined the vertices v and w. Then in particular, it is a path from v to w. The previous exercise has taught us that this is the only path from v to w, and hence removing it yields it so that there is no path from v to w any more. So, v and w must now be in separate components, and thus the number of components has increased. So we have seen that any edge is indeed a bridge.

Exercise 3.C

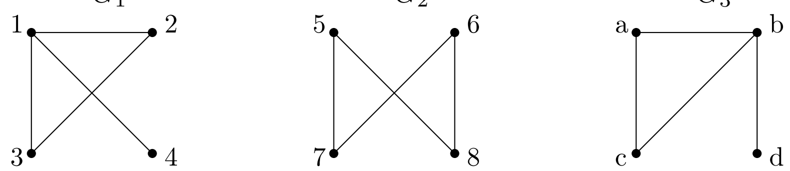
Prove that for all $n \geq 1$, it holds that K_n has exactly $\frac{1}{2}n(n-1)$ edges.

Solution: Every vertex in K_n has degree $n-1$, that is, there are $n-1$ edges leaving it. And as there are n vertices, that means that there is a total of $n(n-1)$ edges. However, because every vertex is a starting point as well as an ending point, we have now counted every edge exactly twice. So, the total number of edges of K_n is $\frac{1}{2}n(n-1)$.

In Exercise 3.S we prove the same theorem formally using mathematical induction.

Exercise 3.D

Check which of the graphs below are isomorphic to each other:

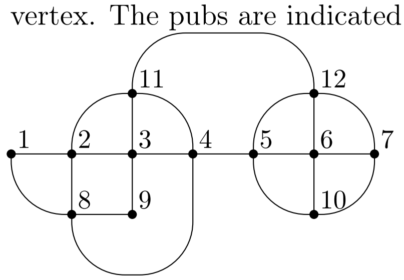


If two graphs are isomorphic, give an isomorphism between the two. If they are not, then explain why such an isomorphism cannot exist.

Solution: If two graphs are isomorphic, they will have the same number of vertices, and these vertices will have the same degree. The vertices in G_2 all have degree 2, which is not the case for the other graphs. Hence, G_2 is not isomorphic to either G_1 or G_3. But, G_1 and G_3 are isomorphic to each other. An isomorphism can be given by φ with $\varphi(1) = b$, $\varphi(2) = a$, $\varphi(3) = c$, and $\varphi(4) = d$, which we can also abbreviate to: $1 \mapsto b$, $2 \mapsto a$, $3 \mapsto c$, and $4 \mapsto d$ without having to explicitly give the isomorphism a name.

Exercise 3.E

Given here is a city map G of a village, on which streets are indicated by edges. There are pubs located on every vertex. The pubs are indicated by vertices, numbered 1 through 12:



Formulate the following questions in terms of Hamiltonian and Eulerian circuits and paths, and answer them as well:

(i) Is it possible to make a walk in such a way that every street is traversed only once? If so, give an example, and if not, explain why.

Solution: Such a walk, where every street, that is, edge, is traversed exactly once, is an Eulerian path. So the question is whether an Eulerian path exists. Theorem 3.13 provides us with an easy way to check this. In order to check this, we need to know the degrees of the pubs, or, vertices.

vertex	degree	vertex	degree	vertex	degree	vertex	degree
1	2	4	4	7	3	10	3
2	4	5	4	8	4	11	4
3	4	6	4	9	2	12	4

Only vertices 7 and 10 have odd degree, so according to Euler's theorem, an Eulerian path indeed exists. A concrete example of such a path would be the following: $7 \to 12 \to 5 \to 10 \to 7 \to 6 \to 12 \to 11 \to 4 \to 8 \to 1 \to 2 \to 11 \to 3 \to 9 \to 8 \to 2 \to 3 \to 4 \to 5 \to 6 \to 10$.

(ii) Is it possible to make a walk, passing every street exactly once, and starting and ending in pub 3? If so, give an example, and if not, explain why.

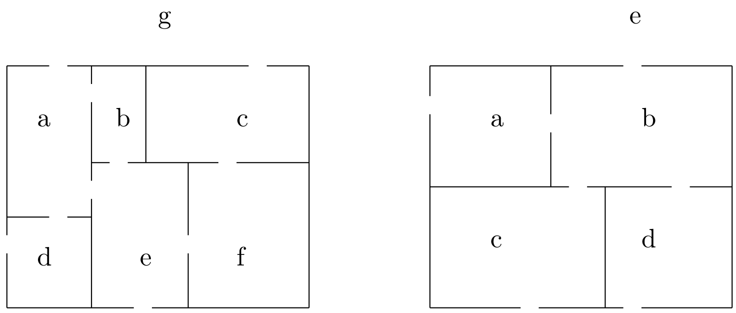
Solution: Now the question translates to whether an Eulerian cycle exists. Again, we use Euler's theorem, which states that such a cycle exists if and only if every vertex has an even degree. We have seen that this is not the case, and hence an Eulerian cycle does not exist. (And thus in particular, no Eulerian cycle exists based around pub 3.)

(iii) Can a pub crawl be organized such that every pub is visited exactly once? If so, give an example, and if not, explain why.

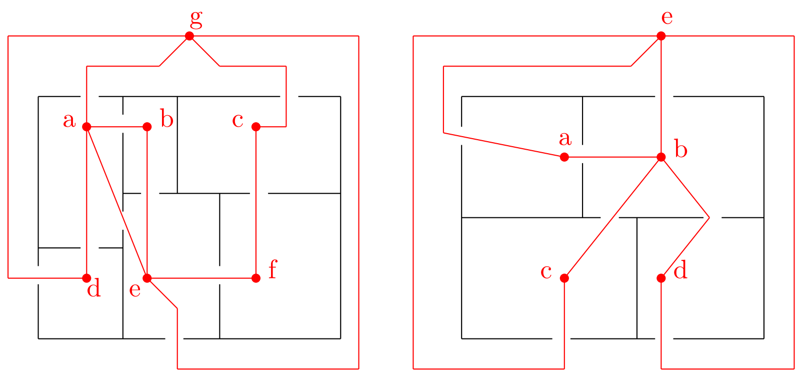
Solution: So now the question is whether a Hamiltonian path exists. And indeed this is the case. Take for example: $1 \to 2 \to 11 \to 12 \to 7 \to 6 \to 10 \to 5 \to 4 \to 3 \to 9 \to 8$. In fact, this Hamiltonian path can be extended to a Hamiltonian cycle by adding the edge $(8, 1)$. This is not always possible. Take for instance the Hamiltonian path $12 \to 7 \to 10 \to 6 \to 5 \to 4 \to 11 \to 3 \to 9 \to 8 \to 1 \to 2$. As there is no edge $(2, 12)$ this path cannot be extended to a Hamiltonian cycle.

Exercise 3.F

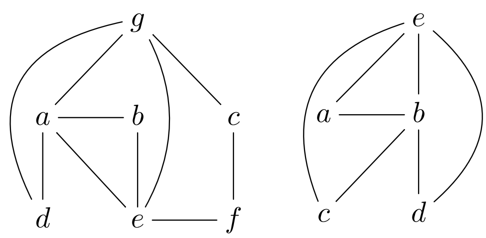
Below, two floor plans of houses are given, in which the rooms and the garden have been given names.



(i) For both floor plans, draw a corresponding graph, where the rooms, including the garden, become vertices, and the doors connecting rooms become edges connecting vertices.
Solution:



Or, abstracting away the floor plan guide:



(ii) For both houses, check whether it is possible to make a stroll through the house in such a way that every door is used exactly once, and you end up in the room where you started out. Explain your answer, and if you argue that such a stroll is possible, give an explicit example.
Solution: Because the doors correspond to edges, the question translates to whether an Eulerian cycle exists. So, as usual, we count the degrees of the vertices to find out.

	a	b	c	d	e	f	g
First house	4	2	2	2	4	2	4
Second house	2	4	2	2	4		

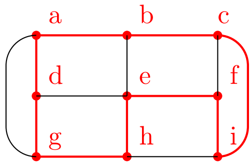
All rooms of both houses indeed have even degrees, and thus, according to Euler's theorem (3.13), Eulerian cycles exist. Concrete examples are $a \to g \to d \to a \to b \to e \to f \to c \to g \to e \to a$ for the first house and $e \to a \to b \to c \to e \to b \to d \to e$ for the second.

(iii) For both houses, check whether a stroll exists that passes through each room, as well as the garden, exactly once, returning to the original room afterwards. Explain your answer and give concrete examples if these strolls indeed exist.
Solution: Now, we are asking whether a Hamiltonian circuit exists. In the first house, such a path is indeed possible: $g \to d \to a \to b \to e \to f \to c \to g$.
However, in the second house, such a path is not possible. In a Hamiltonian circuit, every edge connected to a vertex that has degree 2 must be included, because that vertex will

have to be traversed. If we draw out these edges, (e, c), (c, b), (e, a), (a, b), (e, d), and (d, b), we see that we can't complete the circuit. The vertex e, as well as b, appears in three of these edges and in a Hamiltonian circuit every vertex may only appear in two edges.

Exercise 3.G
Which of the following graphs has a Hamiltonian circuit? Provide such a circuit or explain why a Hamiltonian circuit cannot exist.

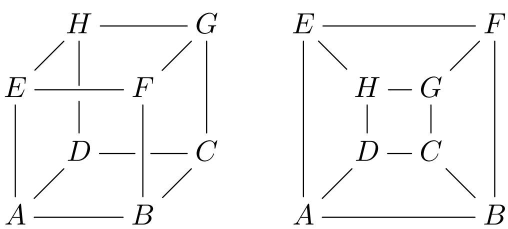
Solution: The first graph contains Hamiltonian circuits, for instance, $a \to b \to c \to i \to f \to e \to h \to g \to d \to a$:



The second graph does not contain any Hamiltonian circuits. We will try to construct such a circuit and show that this leads to a contradiction. Note that we can divide this graph in three parts. A 'left bank' consisting of the vertices a, d and g and the edges (a, d), (d, g) and (a, g). A 'right bank' consisting of the vertices c, f and i and the edges (c, f), (f, i) and (c, i). And three 'bridges' consisting of the vertex b and the edges (a, b) and (b, c), the vertex e and the edges (d, e) and (e, f), the vertex h and the edges (g, h) and (h, i), We have already seen before in Exercise 3.F that any Hamiltonian circuit passes through all edges incident with vertices of degree 2. Because b, e and h have degree 2, each Hamiltonian circuit must contain the three bridges exactly once. Furthermore, if there exists a Hamiltonian circuit in this graph, we know that in particular there exists a Hamiltonian circuit that starts in vertex a, which is on the left bank of the graph. Now if this Hamiltonian circuit starting on the left bank includes all three bridges exactly once, we know for sure that our circuit must end at the right bank. However, because any circuit should end where it starts, this is a contradiction and hence we know that no Hamiltonian circuit can exist in this graph. The graph does contain Hamiltonian paths, for instance: $a \to b \to c \to f \to e \to d \to g \to h \to i$.

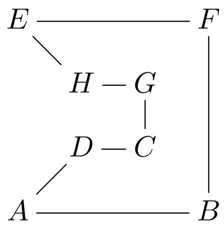
Exercise 3.H
Let $Q_3 = \langle V, E \rangle$ be the three dimensional hypercube graph, where V are the eight corners of a cube and E are the twelve edges connecting each vertex with three other vertices.

(i) Is Q_3 planar?
Solution: Below we give two representations of Q_3. The second one proves that Q_3 is indeed planar.



(ii) Does Q_3 have a Hamiltonian circuit?

Solution: Yes, take for instance $A \to B \to F \to E \to H \to G \to C \to D \to A$:



(iii) Does Q_3 have an Eulerian circuit?
Solution: No, the graph Q_3 has eight vertices of degree 3. And Euler's theorem says that in such a connected graph an Eulerian circuit can only exist if each vertex has an even degree.

Don't forget to explain your answers!

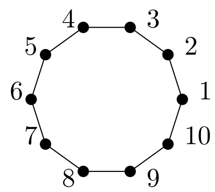
Exercise 3.I

Show that the Petersen graph does contain a Hamiltonian path, but doesn't contain a Hamiltonian cycle.

Solution: It is not difficult to find a Hamiltonian path. Take for instance $ab \to cd \to ae \to bc \to de \to ac \to bd \to ce \to ad \to be$.

Proving that it doesn't have a Hamiltonian cycle is more difficult. First note that the Petersen graph has no cycles of length three or four. We will now show that if the Petersen graph has a Hamiltonian cycle, it also has cycles of length three or four. And therefore such a Hamiltonian cycle cannot exist.

Now assume that we have a Hamiltonian cycle. Then this is basically the cycle graph with ten vertices and ten edges and we may rename the vertices in such a way that we get the following graph.



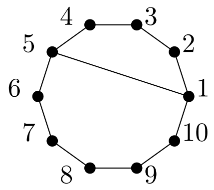
The Petersen graph has fifteen edges, so we must add five more edges to this graph to get the Petersen graph. More specific, since each vertex in the Petersen graph has degree three, we must add exactly one edge to each vertex. We could do this by drawing 'chords', i.e. edges that are completely in the inner part of this graph. If we draw such a chord we get a new cycle with a length less than or equal to six. We make a case distinction.

Each vertex is connected to a vertex that is five steps away. So vertex 1 is connected to 6, 2 to 7, 3 to 8, 4 to 9, and 5 to 10. Now this gives us a cycle of length four: $1 \to 2 \to 7 \to 6 \to 1$, so this can't be the Petersen graph.

At least one vertex is connected to a vertex that is two steps away. So we may assume that vertex 1 is connected to 3. Now we have a cycle of length three: $1 \to 2 \to 3 \to 1$, so this can't be the Petersen graph either.

At least one vertex is connected to a vertex that is three steps away. Here we may assume that vertex 1 is connected to 4. Now we have a cycle of length four: $1 \to 2 \to 3 \to 4 \to 1$, so this can't be the Petersen graph either.

At least one vertex is connected to a vertex that is four steps away. Hence we may assume that vertex 1 is connected to 5. So we have



However, as we have noted before, to each vertex exactly one edge should be added. So in particular we have to add an edge to vertex 6. This is a list of all possible options:

Add $(6, 1)$**.** Now vertex 1 has degree 4, so this can't be right.

Add $(6, 2)$**.** Now we have a cycle of length four: $1 \to 5 \to 6 \to 2 \to 1$.

Add $(6, 3)$**.** Now we have a cycle of length four: $3 \to 4 \to 5 \to 6 \to 3$.

Add $(6, 4)$**.** Now we have a cycle of length three: $4 \to 5 \to 6 \to 4$.

Add $(6, 5)$**.** Now we have multiple edges between vertices 5 and 6.

Add $(6, 6)$**.** Now we have a loop from vertex 6 to itself.

Add $(6, 7)$**.** Now we have multiple edges between vertices 6 and 7.

Add $(6, 8)$**.** Now we have a cycle of length three: $6 \to 7 \to 8 \to 6$.

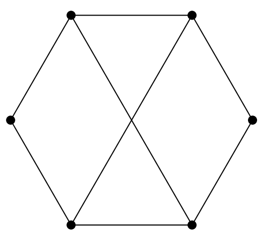
Add $(6, 9)$**.** Now we have a cycle of length four: $6 \to 7 \to 8 \to 9 \to 6$.

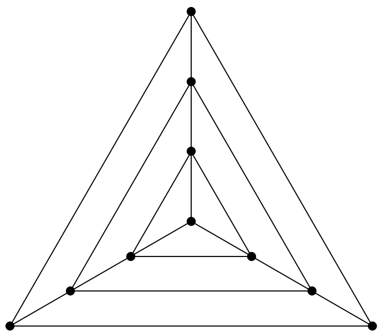
Add $(6, 10)$**.** Now we have a cycle of length four: $1 \to 5 \to 6 \to 10 \to 1$.

So in all of these case we see that the resulting graph cannot be the Petersen graph, so apparently we cannot add an edge to vertex 6, which was obligatory.

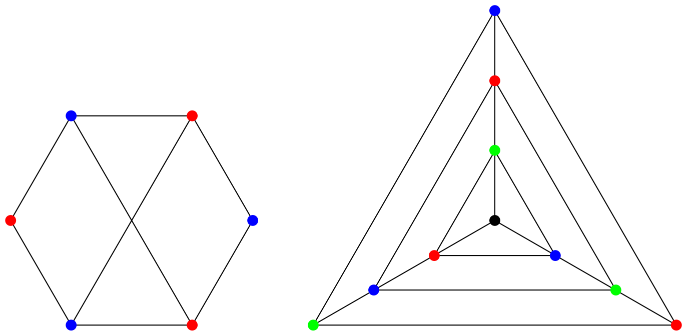
So no matter how we try, it is simply not possible to extend the cycle of length ten to the Petersen graph. Hence the Petersen graph has no Hamiltonian cycle.

Exercise 3.J
Find the chromatic number for the following two graphs. Explain your answer.





Solution: The default method to show that a (finite) graph has a certain chromatic number consists of two stages. First you show that there exists a coloring with this number of colors by explicitly giving it. Then you show that there cannot exist a coloring with fewer colors.



So now we have shown that the chromatic number of the graphs are at most 2 and 4. Obviously, the graph on the left cannot be colored using only one color, since it has (at least) two connected vertices. So its chromatic number is indeed 2. The graph on the right has K_4 as subgraph in the middle. But in K_4 each vertex is a neighbor of each other vertex, so we already need four colors for that part of the graph. So the chromatic number of the whole graph cannot be less than the chromatic number of the subgraph. So its chromatic number is indeed 4.

Exercise 3.K
Show that if a bipartite graph has a Hamiltonian path, the number of red vertices and blue vertices differs at most one.

Solution: If a graph is bipartite then by definition it can be divided into a set of red vertices and a set of blue vertices, such that each edge connects a red vertex with a blue vertex. A Hamiltonian path visits all vertices exactly once. Hence in a bipartite graph this means that in a Hamiltonian path the colors of the visited vertices are constantly changing from red to blue and back, because there are no edges that connect vertices of the same color. Without loss of generality we may assume that the first vertex in the Hamiltonian path is red. Then, if the number of vertices is even, our Hamiltonian path ends with a blue vertex and the number of red and blue vertices is exactly the same. But if the number of vertices is odd, our Hamiltonian path ends with a red vertex and the number of red vertices is exactly one higher than the number of blue vertices.

Exercise 3.L
At a certain university quite a lot of language courses are being offered: Arabic, Bengali, Catalan, Danish, Estonian, Filipino and Greek. When creating the schedule for these courses the schedule maker has to take these requirements into account:

- All languages are being taught each day.
- Each lesson takes 105 minutes.
- The slots for the lessons are 08.45–10.30, 10.45–12.30, 13.45–15.30, 15.45–17.30 and 18.45–20.30.
- The building with five lecture rooms can only be rented as a whole, so the more courses are being taught in parallel, the cheaper it will be for the university.
- Some students have registered for more than one course and hence these courses should not be given in parallel. In the table below the places marked with $*$ indicate that there is at least one student that has registered for both the language in this row as the language in this column.

	A	B	C	D	E	F	G
A		*	*	*			*
B	*		*	*	*		*
C	*	*		*		*	
D	*	*	*			*	
E		*					
F			*	*			*
G	*	*				*	

Give a schedule for the daily lessons that complies with the given requirements. Use graph theory to prove that your schedule is optimal.

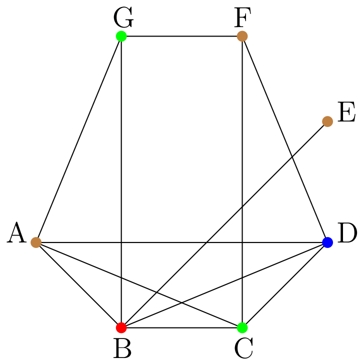
Solution: The minimal number of sequential lessons is the same as the chromatic number of the graph where the vertices are the languages and the edges connect the languages that are followed by (partially) the same students. In other words the edges indicate which languages cannot be scheduled in parallel.

We start with language A and go through the whole table to determine all edges. And as soon as we know that we need a new color, we assign it to a vertex. Because of the symmetry of the table, we only have to take the upper right part of the table into account. This method ensures that we get a minimal coloring.

- Language A must have a color: brown.
- Languages A and B may not be scheduled in parallel, so B needs to get a fresh color: red.
- Languages A and C may not be scheduled in parallel, so C is not brown. But C may still be red.
- Languages A and D may not be scheduled in parallel, so D is not brown. But D may still be red.
- Languages A and G may not be scheduled in parallel, so G is not brown. But G may still be red.
- Languages B and C may not be scheduled in parallel, so C is not brown and not red. So C needs to get a fresh color: green.
- Languages B and D may not be scheduled in parallel, so D is not brown and not red. But D may still be green.
- Languages B and E may not be scheduled in parallel, so E is not red. But E may still be brown or green.
- Languages B and G may not be scheduled in parallel, so G is not brown and not red. But G may still be green.
- Languages C and D may not be scheduled in parallel, so D is not brown, not red and not green. So D needs to get a fresh color: blue.
- Languages C and F may not be scheduled in parallel, so F is not green. But F may still be brown, red or blue.
- Languages D and F may not be scheduled in parallel, so F is not green and not blue. But F may still be brown or red.
- Because the only restriction for E is that it is not scheduled in parallel with B we know that E is not red, but we may choose any of the other colors. We choose brown.

- Languages F and G may not be scheduled in parallel, so G is not brown and not red and it has a color different from the color of F.
- There are no additional restrictions anymore, so we still have some freedom in choosing the colors for F and G. We choose F to be brown and G to be green.

So we have colored our graph with four colors: brown for Arabic, Estonian and Filipino; red for Bengali; green for Catalan and Greek; blue for Danish.



So the schedule maker needs four sequential lessons. A possible schedule would be:

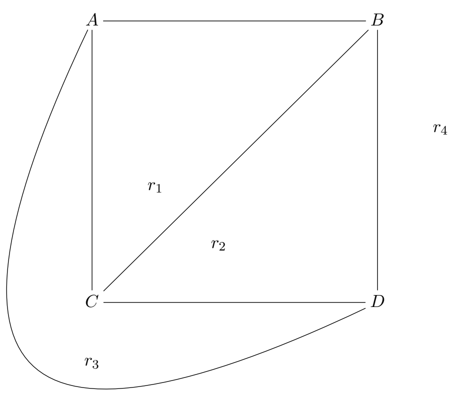
Time	Room 1	Room 2	Room 3	Room 4	Room 5
08.45–10.30	Arabic	Estonian	Filipino		
10.45–12.30		Bengali			
13.45–15.30	Catalan	Greek			
15.45–17.30	Danish				
18.45–20.30					

Completely irrelevant for this particular question, but could you try to check yourself whether the presented graph is planar or not?

Well, what do you think? Could you find a planar representation for this graph? Do you want to search a bit longer? Then don't read the next paragraph...

We prove that such a planar representation does not exist in three ways. The first proof deals with regions and is somewhat ad-hoc, but should be understandable by all of you. The last two proofs use some graph theory that isn't part of this course, so if you don't understand these proofs, that is no problem at all.

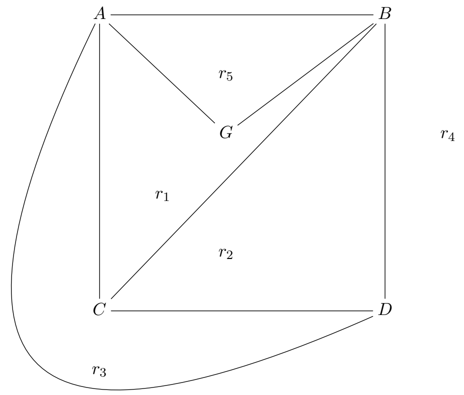
1. Note that our graph contains K_4 as a subgraph. For K_4 we know that this is a planar graph, with basically one planar representation:



We see that this planar representation divides the plane into four regions r_1, r_2, r_3 and r_4. We will try to add the remaining vertices and show that this cannot be done without loosing planarity.

We start by adding G, which is adjacent to A and B. (And F, but since F is not in the graph yet, that is not relevant.)

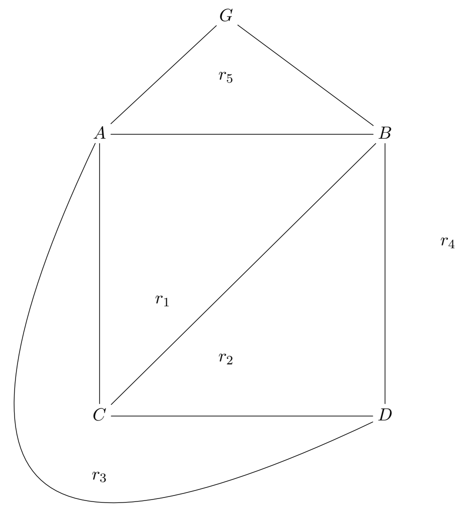
- It is easy to see that G cannot be in r_2, because we cannot add edge $\{A, G\}$ without crossing existing edges.
- It is easy to see that G cannot be in r_3, because we cannot add edge $\{B, G\}$ without crossing existing edges.
- At first sight G can be in r_1 since we can add the edges $\{A, G\}$ and $\{B, G\}$ in a planar way:



  But now we try to add F, which is adjacent to C, D and G.

  - F cannot be in r_1 or r_5 because we cannot add edge $\{D, F\}$ without crossing existing edges.
  - F cannot be in r_2, r_3 or r_4 because we cannot add edge $\{F, G\}$ without crossing existing edges.

  Hence F cannot be added if G is in r_1. So G cannot be in r_1.
- At first sight G can be in r_4 since we can add the edges $\{A, G\}$ and $\{B, G\}$ in a planar way:



  But now we try to add F, which is adjacent to C, D and G.

  - F cannot be in r_1, r_2 or r_3 because we cannot add edge $\{F, G\}$ without crossing existing edges.

- F cannot be in r_4 or r_5 because we cannot add edge $\{C, F\}$ without crossing existing edges.

  Hence F cannot be added if G is in r_4. So G cannot be in r_4.

So G cannot be in either one of the four regions! Hence we cannot extend this graph in a planar way to original graph. This implies that the original graph is not planar.

2. If we look at the subgraph that we get by removing the edges $\{A, B\}$, $\{B, E\}$ and $\{C, D\}$ and vertex E, we get a graph which is isomorphic to $K_{3,3}$. And according to *Kuratowski's Theorem*[1] this implies that the original graph is not planar.

3. If we remove edge $\{B, E\}$ and vertex E, and do a so-called *edge contraction* on edge $\{F, G\}$, which basically means that we merge the vertices F and G into a single vertex FG, but keep all other original edges connected to F or G, then we get a *minor* of the original graph which is isomorphic to K_5. And according to *Wagner's theorem*[2] this implies that the original graph is not planar.

So in particular this graph proves that a chromatic number of 4 is a *necessary* condition for planarity (according to the Four color theorem) but not a *sufficient* one!

Exercise 3.M
The sequence a_n is recursively defined by:

$$\begin{array}{rcl} a_0 & = & 3 \\ a_{n+1} & = & a_n^2 - 2a_n \text{ for } n \geq 0 \end{array}$$

Use this definition to compute the value of a_5.

Solution:

$$\begin{array}{l} a_1 = a_0^2 - 2a_0 = 3^2 - 2 \cdot 3 = 3 \\ a_2 = a_1^2 - 2a_1 = 3^2 - 2 \cdot 3 = 3 \\ a_3 = a_2^2 - 2a_2 = 3^2 - 2 \cdot 3 = 3 \\ a_4 = a_3^2 - 2a_3 = 3^2 - 2 \cdot 3 = 3 \\ a_5 = a_4^2 - 2a_4 = 3^2 - 2 \cdot 3 = 3 \end{array}$$

Exercise 3.N
The sequence b_n is recursively defined by:

$$\begin{array}{rcl} b_0 & = & 4 \\ b_{n+1} & = & b_n^2 - 2b_n \text{ for } n \geq 0 \end{array}$$

Use this definition to compute the value of b_5.

Solution:

$$\begin{array}{lcccc} b_1 = b_0^2 - 2b_0 = & 4^2 - 2 \cdot 4 & = & 16 - 8 & = & 8 \\ b_2 = b_1^2 - 2b_1 = & 8^2 - 2 \cdot 8 & = & 64 - 16 & = & 48 \\ b_3 = b_2^2 - 2b_2 = & 48^2 - 2 \cdot 48 & = & 2304 - 96 & = & 2208 \\ b_4 = b_3^2 - 2b_3 = & 2208^2 - 2 \cdot 2208 & = & 4875264 - 4416 & = & 4870848 \\ b_5 = b_4^2 - 2b_4 = & 4870848^2 - 2 \cdot 4870848 & = & 23725160239104 - 9741696 & = & 23725150497408 \end{array}$$

Exercise 3.O
Consider the sequence c_n for $n \geq 0$, given by the values:

$$1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843, 1364, 2207, 3571, 5778, 9349, \ldots$$

[1] See `https://en.wikipedia.org/wiki/Kuratowski%27s_theorem`.

[2] See `https://en.wikipedia.org/wiki/Wagner%27s_theorem`.

Give a recursive definition for c_n.

Solution: Starting from the third value it appears that each value is the sum of its two predecessors. So we need two initial values and one recursive definition:

$$\begin{aligned} c_0 &= 1 \\ c_1 &= 3 \\ c_{n+2} &= c_{n+1} + c_n \text{ for } n \geq 0 \end{aligned}$$

Exercise 3.P

The Python program in Example 3.32 is not very robust. If we provide for n a negative integer, then Python will give the error *RecursionError: maximum recursion depth exceeded* as the base case is never reached. Now modify the program in such a way that if both m and n are integers, such that $|m| \leq 100$ and $|n| \leq 100$, Python gives the correct result.

Solution: This program should be simple enough for you to understand, even without knowing the exact syntax and semantics of Python. Actually, if you do know the syntax of Python, you are probably able to write the same procedures in a much more concise way, but that was not the goal of this exercise.

```python
import sys;

#sys.setrecursionlimit(100000)

def mult(m, n):
    """This is a recursive function to compute
       the product of m and n"""

    if n == 0:
        return 0
    else:
        if n < 0:
            return -mult(m, -n)
        else:
            return mult(m, n-1) + m


def power(m, n):
    """This is a recursive function to compute
       the value of m to the power n"""

    if n == 0:
        return 1
    else:
        if n < 0:
            return 1 / power(m, -n)
        else:
            return mult(power(m, n-1), m)


m = int(sys.argv[1])
n = int(sys.argv[2])
print("The product of " + str(m) + " and " + str(n) +
      " is " + str(mult(m,n)) + ".")
print("The value of " + str(m) + " to the power " + str(n) +
      " is " + str(power(m,n)) + ".")
```

Exercise 3.Q

Consider the sequence a_n defined by:

$$\begin{aligned} a_0 &= 0 \\ a_{n+1} &= a_n + 2n + 1 \quad \text{for } n \geq 0 \end{aligned}$$

Prove by induction that for all $n \in \mathbb{N}$, $a_n = n^2$.

Solution: We apply the template:

Proposition: 0

$a_n = n^2$ for all $n \geq 0$.

Proof by induction on n. 1

We first define our predicate P as:

$$P(n) \;:=\; \left[a_n = n^2 \right]$$ 2

Base Case. We show that $P(0)$ holds, i.e. we show that 3

$a_0 = 0^2$

This indeed holds, because $a_0 = 0$ by definition and $0 = 0^2$. 4

Induction Step. Let k be any natural number such that $k \geq 0$. 5

Assume that we already know that $P(k)$ holds, i.e. we assume that 6
$a_k = k^2$ (Induction Hypothesis IH)

We now have to show that $P(k+1)$ also holds, i.e. we now have to show that 7
$a_{k+1} = (k+1)^2$

This indeed holds, because if we start rewriting the left-hand side of the equation, apply the induction hypothesis at the proper moment, rewrite further, we eventually end up with the right-hand side of the equation: 8

$$\begin{aligned} a_{k+1} &= a_k + 2k + 1 \text{ (by definition of } a_{k+1}) \\ &= k^2 + 2k + 1 \text{ (apply IH)} \\ &= (k+1)^2 \text{ (elementary algebra)} \end{aligned}$$

Hence it follows by induction that $P(n)$ holds for all $n \geq 0$. 9

Exercise 3.R
The inventor of the chessboard was told by the king of Persia, that he would be rewarded any object of choice. The inventor chose the following: 1 grain of rice on the first field of the chessboard, 2 grains of rice on the second, 4 on the third, and so on, doubling the number of grains for each successive field. The king thought the inventor to be very humble. Now the question is: how many grains of rice did the inventor's choice amount to? Let's formulate it formally: he asked for

$$1 + 2 + 2^2 + 2^3 + 2^4 + \cdots + 2^{63}$$

grains of rice. Can you find a direct formula for the result of this sum for a generalized board with n fields? And can you then prove this formula by induction? [*Hint:* This is what you could do here: Give a recursive definition for s_n, the number of rice grains on the first n fields. Make a table with three columns: one column for n, one for the expression $1 + 2 + \cdots + 2^{n-1}$ and one for s_n. Use this table to guess a direct formula $f(n)$ for s_n. Prove by induction that $f(n) = s_n$ for all $n \geq 1$. Use the direct formula to compute s_{64}.]

Solution: Define the sequence s_n recursively as:

$$\begin{aligned} s_1 &= 1 \\ s_{n+1} &= s_n + 2^n \quad \text{for } n \geq 1 \end{aligned}$$

In particular, s_n now stands for the grains of rice on the first n fields, added together. So, eventually we want to know the value of s_{64}. We follow the hint and try some small cases. This gives the following table:

n	expression	s_n
1	1	1
2	$1+2$	3
3	$1+2+2^2$	7
4	$1+2+2^2+2^3$	15
5	$1+2+2^2+2^3+2^4$	31

Hopefully, after computing this table, you have the presumption that $s_n = 2^n - 1$. We will now try to prove this by induction using the template.

Proposition: 0

$s_n = 2^n - 1$ for all $n \geq 1$.

Proof by induction on n. 1

We first define our predicate P as:

$$P(n) \quad := \quad [s_n = 2^n - 1]$$ 2

Base Case. We show that $P(1)$ holds, i.e. we show that 3

$s_1 = 2^1 - 1$

This indeed holds, because 4

$$2^1 - 1 = 2 - 1 = 1$$

and $s_1 = 1$ by definition.

Induction Step. Let k be any natural number such that $k \geq 1$. 5

Assume that we already know that $P(k)$ holds, i.e. we assume that 6
$s_k = 2^k - 1$ (Induction Hypothesis IH)

We now have to show that $P(k+1)$ also holds, i.e. we now have to show that 7
$s_{k+1} = 2^{k+1} - 1$

This indeed holds, because 8

$$\begin{aligned} s_{k+1} &= s_k + 2^k \text{ (definition of } s_{k+1}) \\ &= \left(2^k - 1\right) + 2^k \text{ (using the IH)} \\ &= 2^k + 2^k - 1 \text{ (elementary algebra)} \\ &= 2 \cdot 2^k - 1 \text{ (elementary algebra)} \\ &= 2^{k+1} - 1 \text{ (elementary algebra)} \end{aligned}$$

Hence it follows by induction that $P(n)$ holds for all $n \geq 1$. 9
Returning to the original question, we now only need to compute s_{64}:

$$s_{64} = 2^{64} - 1 = 18446744073709551615$$

Exercise 3.S
In Exercise 3.C, we proved that for all $n \geq 1$ the complete graph K_n has exactly $\frac{1}{2}n(n-1)$ edges. Now, try to prove this by induction on n.

Solution: The proof follows the usual pattern of a proof by induction. In particular we see here that it can happen that we need some natural language in the definition of our predicate $P(n)$.

Proposition: 0

K_n has exactly $\frac{1}{2}n(n-1)$ edges for all $n \geq 1$.

Proof by induction on n. 1

We first define our predicate P as:

$$P(n) \quad := \quad \left[K_n \text{ has exactly } \tfrac{1}{2}n(n-1) \text{ edges}\right]$$ 2

Base Case. We show that $P(1)$ holds, i.e. we show that 3

K_1 has exactly $\frac{1}{2} \cdot 1 \cdot (1-1)$ edges

This indeed holds, because K_1 consists of one vertex and zero edges. And of course $\frac{1}{2} \cdot 1 \cdot (1-1) = 0$. 4

Induction Step. Let k be any natural number such that $k \geq 1$. 5

Assume that we already know that $P(k)$ holds, i.e. we assume that 6
K_k has exactly $\frac{1}{2} \cdot k \cdot (k-1)$ edges (Induction Hypothesis IH)

We now have to show that $P(k+1)$ also holds, i.e. we now have to show that 7
K_{k+1} has exactly $\frac{1}{2} \cdot (k+1) \cdot (k+1-1)$ edges

This indeed holds, because we know that we can construct K_{k+1} out of K_k by adding a single vertex and k edges from this new vertex to all pre-existing vertices of K_k. With the IH, we know that K_k has exactly $\frac{1}{2}k\,(k-1)$ edges. And because our construction to K_{k+1} adds k edges, we compute that the number of edges of K_{k+1} equals: 8

$$\begin{aligned}
k + \frac{1}{2}k\,(k-1) &= \frac{2}{2}k + \frac{1}{2}k\,(k-1) \text{ (making denominators equal)}\\
&= \frac{2k}{2} + \frac{k\,(k-1)}{2} \text{ (multiplying fractions)}\\
&= \frac{2k + k\,(k-1)}{2} \text{ (adding fractions)}\\
&= \frac{2k + k^2 - k}{2} \text{ (removing parentheses)}\\
&= \frac{k^2 + k}{2} \text{ (simplifying)}\\
&= \frac{(k+1)\,k}{2} \text{ (rewriting)}\\
&= \frac{1}{2}\,(k+1)\,k \text{ (rewriting)}\\
&= \frac{1}{2}\,(k+1)\,(k+1-1) \text{ (rewriting)}
\end{aligned}$$

And that is exactly what we aimed to prove.

Hence it follows by induction that $P(n)$ holds for all $n \geq 1$. 9

Exercise 3.T
Prove by induction, that for all $n \in \mathbb{N}$, $2^n \geq n$.

Solution: In this example we will see that the general template doesn't fit the situation exactly, so we have to modify it slightly.

Proposition: 0

$2^n \geq n$ for all $n \geq 0$.

Proof by induction on n. 1

We first define our predicate P as:

$$P(n) \;:=\; [2^n \geq n]$$ 2

Base Case. We show that $P(0)$ holds, i.e. we show that 3

$2^0 \geq 0$

This indeed holds, because $2^0 = 1$ and $1 \geq 0$. 4

Now if we could use the general template, our proof would go on like this:

Induction Step. Let k be any natural number such that $k \geq 0$. 5

Assume that we already know that $P(k)$ holds, i.e. we assume that 6
$2^k \geq k$ *(Induction Hypothesis IH)*

We now have to show that $P(k+1)$ also holds, i.e. we now have to show that 7
$2^{k+1} \geq k+1$
This indeed holds, because 8

$$\begin{aligned} 2^{k+1} &= 2 \cdot 2^k \\ &\overset{IH}{\geq} 2k \\ &= k+k \\ &\geq k+1 \end{aligned}$$

However, this last step is not true for all of our possible values of k! We started the induction step with assuming that $k \geq 0$, but here it is obviously clear that if $k = 0$ then $k + k = 0 + 0 = 0 \not\geq 1 = 0 + 1 = k + 1$. So we cannot use this induction step to prove $P(1)$ from $P(0)$. However, it does hold for all $k \geq 1$ that we have that $P(k+1)$ follows from $P(k)$. So this means that we can modify the general template by adding a second base case and only after that second base case use the induction step with an adjusted lower bound for k.

Base Case. We show that $P(1)$ holds, i.e. we show that 3

$2^1 \geq 1$

This indeed holds, because $2^1 = 2$ and $2 \geq 1$. 4

Induction Step. Let k be any natural number such that $k \geq 1$. 5

Assume that we already know that $P(k)$ holds, i.e. we assume that 6
$2^k \geq k$ (Induction Hypothesis IH)

We now have to show that $P(k+1)$ also holds, i.e. we now have to show that 7
$2^{k+1} \geq k+1$

This indeed holds, because 8

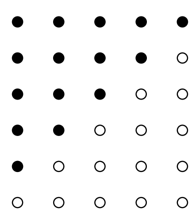
$$\begin{aligned} 2^{k+1} &= 2 \cdot 2^k \\ &\overset{\text{IH}}{\geq} 2k \\ &= k+k \\ &\geq k+1 \text{ (because } k \geq 1) \end{aligned}$$

Hence it follows by induction that $P(n)$ holds for all $n \geq 0$. 9

Exercise 3.U
So we have seen the formula for the sum of the following sequence:

$$1 + 2 + 3 + 4 + 5 + \cdots + n = \frac{(n+1)\,n}{2}.$$

What is the connection with this picture?



Solution: The key lies in surfaces. What is the formula for the surface of triangles? What is the formula for the surface of a rectangle? What, here, are the dimensions, and why must you divide by two?

Exercise 3.V
How many distinct sequences of length n can we make, filled with the numbers 1 through 5? This too, can be solved easily with recursion. Let a_n denote the number of distinct sequences of length n. The simplest sequence, of length one, can of course be made in five ways, so $a_1 = 5$. A sequence of length $n+1$ can be made by first taking a sequence of length n, and then appending a new element after it, for which we have five options. So, $a_{n+1} := 5 \cdot a_n$. Now, recall the definition of raising to a power. Prove by induction, that for all $n \geq 1$, $a_n = 5^n$.

Solution: Just to be clear: the sequences we are talking of here are allowed to contain the numbers 1 through 5 a multiple number of times, as for example this sequence of length seven: $1, 4, 2, 2, 3, 5, 2$. This can of course be inferred from the fact that we are also talking of sequences of length ≥ 6, and such a sequence will have to contain at least some numbers a multiple number of times.

After this observation we can start with the proof, following the normal template.

Proposition: 0

$a_n = 5^n$ for all $n \geq 1$.

Proof by induction on n. 1

We first define our predicate P as:

$$P(n) \quad := \quad [\, a_n = 5^n \,]$$ 2

Base Case. We show that $P(1)$ holds, i.e. we show that 3

$a_1 = 5^1$

This indeed holds, because $a_1 = 5$ by definition and $5 = 5^1$. 4

Induction Step. Let k be any natural number such that $k \geq 1$. 5

Assume that we already know that $P(k)$ holds, i.e. we assume that 6
$a_k = 5^k$ (Induction Hypothesis IH)

We now have to show that $P(k+1)$ also holds, i.e. we now have to show that 7
$a_{k+1} = 5^{k+1}$

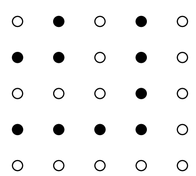
This indeed holds, because 8

$$a_{k+1} = 5 \cdot a_k \overset{\text{IH}}{=} 5 \cdot 5^k = 5^{k+1}$$

Hence it follows by induction that $P(n)$ holds for all $n \geq 1$. 9

Note that here, we only looked at sequences of length 1 or more. However, it also holds for sequences of length 0. The only thing required in the proof, is to add the base case $n = 0$. Check for yourself that indeed the base case $n = 0$ holds. (That is: how many sequences of length 0 exist?)

Exercise 3.W
Prove, by induction, that $1 + 3 + 5 + 7 + \cdots + (2n-1) = n^2$ for $n \geq 1$. What is the connection with the picture below?



Solution: Define the sequence s_n by recursion:

$$\begin{cases} s_1 &= 1 \\ s_n &= s_{n-1} + 2n - 1 \quad \text{for } n \geq 2 \end{cases}$$

In other words $s_n = 1 + 3 + \ldots + (2n-1)$. We want to prove, by induction, that $s_n = n^2$ for $n \geq 1$. Note that in this exercise it is meaningless to talk about the case $n = 0$.

Proposition: 0

$s_n = n^2$ for all $n \geq 1$.

Proof by induction on n. 1

We first define our predicate P as:

$$P(n) \quad := \quad \left[s_n = n^2 \right]$$ 2

Base Case. We show that $P(1)$ holds, i.e. we show that 3

$s_1 = 1^2$

This indeed holds, because $s_1 = 1$ by definition and $1 = 1^2$. 4

Induction Step. Let k be any natural number such that $k \geq 1$. 5

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

$s_k = k^2$ (Induction Hypothesis IH)

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

$s_{k+1} = (k+1)^2$

This indeed holds, because 8

$$\begin{aligned} s_{k+1} &= s_k + 2(k+1) - 1 \text{ (by definition of } s_{k+1}) \\ &= k^2 + 2(k+1) - 1 \text{ (using the IH)} \\ &= k^2 + 2k + 2 - 1 \text{ (elementary algebra)} \\ &= k^2 + 2k + 1 \text{ (elementary algebra)} \\ &= (k+1)^2 \text{ (elementary algebra)} \end{aligned}$$

Hence it follows by induction that $P(n)$ holds for all $n \geq 1$. 9

In the picture, it should be clear that there is a similar connection to surfaces as we have seen before.

Exercise 3.X

We have seen above that $\binom{3}{3} = 1$ and $\binom{3}{4} = 0$. Now use Definition 3.45 to prove the following propositions by induction on n:

(i)

$$\text{for all } l \text{ such that } l > n \text{ it holds that } \binom{n}{l} = 0 \text{ for all } n \geq 0$$

(We use l as k is already needed in the induction scheme.)

Solution: A proof following the template:

Proposition: 0

for all l such that $l > n$ it holds that $\binom{n}{l} = 0$ for all $n \geq 0$.

Proof by induction on n. 1

We first define our predicate P as:

$$P(n) \quad := \quad \left[\text{for all } l \text{ such that } l > n \text{ it holds that } \binom{n}{l} = 0 \right]$$ 2

Base Case. We show that $P(0)$ holds, i.e. we show that 3

for all l such that $l > 0$ it holds that $\binom{n}{l} = 0$

This indeed holds, because if $l > 0$, then $l = (l-1)+1$ where $l-1 \in \mathbb{N}$. So 4

$$\begin{aligned} \binom{0}{l} &= \binom{0}{(l-1)+1} && \text{elementary algebra} \\ &= 0 && \text{equation 2} \end{aligned}$$

Induction Step. Let k be any natural number such that $k \geq 0$. 5

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

for all l' such that $l' > k$ it holds that $\binom{k}{l'} = 0$ (Induction Hypothesis IH)

Note that we use a different variable l' here, as in our proof we are going to instantiate this variable by two different values, both depending on l.

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

for all l such that $l > k+1$ it holds that $\binom{k+1}{l} = 0$

This indeed holds, because if $l > k+1$ then $l = (l-1)+1$ where $l-1 \in \mathbb{N}$ and $(l-1)+1 > k+1$, so $l-1 > k$. 8

$$\begin{aligned} \binom{k+1}{l} &= \binom{k+1}{(l-1)+1} && \text{elementary algebra} \\ &= \binom{k}{l-1} + \binom{k}{(l-1)+1} && \text{equation 4} \\ &= 0 + \binom{k}{(l-1)+1} && \text{IH with } l' = l-1 \text{ and } l-1 > k \\ &= 0 + \binom{k}{l} && \text{algebra} \\ &= 0 + 0 && \text{IH with } l' = l \text{ and } l > k \\ &= 0 && \text{elementary algebra} \end{aligned}$$

(ii)

$$\binom{n}{n} = 1 \text{ for all } n \geq 0$$

[*Hint:* You may need the result of the previous proposition.]
Solution: A proof following the template:

Proposition: 0

$\binom{n}{n} = 1$ for all $n \geq 0$

Proof by induction on n. 1

We first define our predicate P as:

$$P(n) \quad := \quad \left[\binom{n}{n} = 1\right]$$ 2

Base Case. We show that $P(0)$ holds, i.e. we show that 3

$\binom{0}{0} = 1$

This indeed holds, because this is exactly equation 1. 4

Induction Step. Let k be any natural number such that $k \geq 0$. 5

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

$\binom{k}{k} = 1$ (Induction Hypothesis IH)

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

$\binom{k+1}{k+1} = 1$

This indeed holds, because 8

$$\begin{aligned}\tbinom{k+1}{k+1} &= \tbinom{k}{k} + \tbinom{k}{k+1} && \text{equation 4.}\\ &= 1 + \tbinom{k}{k+1} && \text{IH}\\ &= 1 + 0 && \text{previous proposition and } k+1 > k\\ &= 1 && \text{algebra}\end{aligned}$$

Exercise 3.Y
Show that P△$_4$, too, is equal to P△$_1$.

Solution: Proving that these two triangles are equal amounts to proving the following proposition, where in the proof we use the recurrence for binomial coefficients that corresponds to the definition of P△$_1$. (As P△$_1$ only describes how to compute its values without providing any explicit notation for these values, we use the binomial coefficient $\binom{n}{k}$ to denote the value that is placed on coordinate (n, k) according to the definition of P△$_1$. So even though we use the same notation for the coefficients of P△$_1$ as in P△$_2$, this proof is really about P△$_1$, as we use the recurrence for binomial coefficients in the Induction Stepdefined in P△$_1$.)

Proposition: 0

$(1+x)^n = \binom{n}{0}x^0 + \cdots + \binom{n}{n}x^n$ for all $n \geq 0$.

Proof by induction on n. 1

We first define our predicate P as:

$$P(n) \;:=\; \left[(1+x)^n = \tbinom{n}{0}x^0 + \cdots + \tbinom{n}{n}x^n\right]$$ 2

Base Case. We show that $P(0)$ holds, i.e. we show that 3

$(1+x)^0 = \binom{0}{0}x^0$

This indeed holds, because 4

$$(1+x)^0 = 1 = 1 \cdot 1 = \binom{0}{0} \cdot x^0$$

Induction Step. Let k be any natural number such that $k \geq 0$. 5

Assume that we already know that $P(k)$ holds, i.e. we assume that 6
$(1+x)^k = \binom{k}{0}x^0 + \cdots + \binom{k}{k}x^k$ (Induction Hypothesis IH)

We now have to show that $P(k+1)$ also holds, i.e. we now have to show that 7
$(1+x)^{k+1} = \binom{k+1}{0}x^0 + \cdots + \binom{k+1}{k+1}x^{k+1}$

This indeed holds, because 8

$$\begin{aligned}(1+x)^{k+1} &= (1+x)(1+x)^k\\ &= (1+x)^k +\\ &\quad\; x(1+x)^k\\ &\overset{\text{IH}}{=} \tbinom{k}{0}x^0 + \tbinom{k}{1}x^1 + \cdots + \tbinom{k}{k}x^k +\\ &\quad\; x \cdot \left(\tbinom{k}{0}x^0 + \cdots + \tbinom{k}{k-1}x^{k-1} + \tbinom{k}{k}x^k\right)\\ &= \tbinom{k}{0}x^0 + \tbinom{k}{1}x^1 + \cdots + \tbinom{k}{k}x^k +\\ &\qquad\qquad \tbinom{k}{0}x^1 + \cdots + \tbinom{k}{k-1}x^k + \tbinom{k}{k}x^{k+1}\\ &= \tbinom{k}{0}x^0 + \left(\tbinom{k}{1} + \tbinom{k}{0}\right)x^1 + \cdots + \left(\tbinom{k}{k} + \tbinom{k}{k-1}\right)x^k + \tbinom{k}{k}x^{k+1}\\ &= \tbinom{k+1}{0}x^0 + \tbinom{k+1}{1}x^1 + \cdots + \tbinom{k+1}{k}x^k + \tbinom{k+1}{k+1}x^{k+1}\end{aligned}$$

In this last step we used how binomial coefficients are summed in Pascal's triangle $\mathrm{P}\triangle_1$.

$$\begin{aligned}
\binom{k+1}{0} &= \binom{k}{0} \\
\binom{k+1}{1} &= \binom{k}{1} + \binom{k}{0} \\
&\vdots \\
\binom{k+1}{k} &= \binom{k}{k} + \binom{k}{k-1} \\
\binom{k+1}{k+1} &= \binom{k}{k}
\end{aligned}$$

Hence it follows by induction that $P(n)$ holds for all $n \geq 0$. 9

Exercise 3.Z

Demonstrate how the triangle of Pascal can be used to figure out how many distinct ways there are, to pick four objects out of a collection of six. What is the notation for the corresponding binomial?

Solution: In the n-th row of Pascal's triangle, the m-th number from the left stands for the number of distinct ways to pick $(m-1)$ objects from a collection of $(n-1)$ objects. Note how the first number describes how there is exactly 1 way to pick 'nothing at all' from the collection. That may sound stupid, but try to understand it this way: Suppose you would have two ways to pick 'nothing at all' from some collection. Then, these picks should be demonstrably different (by the fact that they are distinct). But, of course, there is no difference. So, it was the same in the first place.

In the case of our question at hand, picking four objects from a collection of six, we are interested in the fifth number from the left of the seventh row of Pascal's triangle:

$$\begin{array}{ccccccccccccccc}
 & & & & & & & 1 & & & & & & & \\
 & & & & & & 1 & & 1 & & & & & & \\
 & & & & & 1 & & 2 & & 1 & & & & & \\
 & & & & 1 & & 3 & & 3 & & 1 & & & & \\
 & & & 1 & & 4 & & 6 & & 4 & & 1 & & & \\
 & & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & & \\
 & 1 & & 6 & & 15 & & 20 & & \boxed{15} & & 6 & & 1 & \\
1 & & 7 & & 21 & & 35 & & 35 & & 21 & & 7 & & 1
\end{array}$$

The corresponding binomial, then, is $\binom{6}{4}$.

Note: there is no 0-th row of the triangle. We can, however, speak of the row with *name* 0, but that will then be the *first* row and not the zeroth! Similarly, we can't speak of the 0-th element of a certain row. Hence we take the fifth element of the seventh row, where you might be tempted to have taken element 4 of row 6—which is something different than the fourth element of the sixth row...

Exercise 3.AA

Assume we have one bill of each of the following types: \$1, \$2, \$5, \$10, \$20, \$50, and \$100. In how many ways can we divide all these bills over three identical purses if there is at most one empty purse?

Solution: So we have seven distinguishable objects that we want to divide over three indistinguishable objects. So we can use the Stirling numbers for this. If we simply wanted to divide those seven bills over the three identical purses the solution would have been:

$$\left\{ 7 \atop 1 \right\} + \left\{ 7 \atop 2 \right\} + \left\{ 7 \atop 3 \right\} = 1 + 63 + 301 = 365$$

However, this solution does not comply to the requirement that there is at most one empty purse. So the real solution is:

$$\left\{ {7 \atop 2} \right\} + \left\{ {7 \atop 3} \right\} = 63 + 301 = 364$$

Exercise 3.AB
Assume we have seven marbles. In how many ways can we distribute all these marbles over three bags (where bags may be empty) if

(i) both the marbles and the bags are distinguishable?
Solution: For each marble we have to choose between three bags, so the solution is $3^7 = 2187$.
Some student came up with this wrong alternative algorithm:

- First divide the seven distinguishable marbles over three indistinguishable bags. This can be done in $\left\{ {7 \atop 1} \right\} + \left\{ {7 \atop 2} \right\} + \left\{ {7 \atop 3} \right\} = 1 + 63 + 301 = 365$ ways.
- Label the indistinguishable bags with three distinct labels. This can be done in $3! = 6$ ways.
- So the total number of ways to divide seven distinguishable marbles over three distinguishable bags is $6 \cdot 365 = 2190 \ldots$?

Can you spot the error in this algorithm? Which specific cases have been counted twice?

(ii) both the marbles and the bags are indistinguishable?
Solution: Because the marbles and the bags are indistinguishable, the only important thing is the amount of marbles in each bag. Unfortunately there is no general formula for this, so we will have to systematically list all 8 possibilities:

7 | 0 | 0	6 | 1 | 0	5 | 2 | 0	5 | 1 | 1
4 | 3 | 0	4 | 2 | 1	3 | 3 | 1	3 | 2 | 2

(iii) the marbles are indistinguishable and the bags are distinguishable? [*Hint:* What do you think that $\boxed{\circ\circ\circ \mid \circ\circ \mid \circ\circ}$ represents?]
Solution: We can list all 36 possibilities, but it may be more interesting to discover the system. The figure above represents the solution that there are three marbles in the first bag, two in the second bag and also two in the third bag. So any solution can be represented by a sequence of nine symbols: seven $\circ$'s for the marbles and two $|$'s for the separation between the bags. So the number of ways to divide the seven marbles over the three bags in this situation is the same as the number of ways we can choose two positions out of nine to place the bag separators. But this is $\binom{9}{2} = 36$.

(iv) the marbles are distinguishable and the bags are indistinguishable?
Solution: So we can either put all seven marbles in one bag, or we can divide them over two bags, or we can divide them over three bags. So we get:

$$\left\{ {7 \atop 1} \right\} + \left\{ {7 \atop 2} \right\} + \left\{ {7 \atop 3} \right\} = 1 + 63 + 301 = 365$$

(v) the marbles are distinguishable and the bags are indistinguishable, but none of the bags may be empty?
Solution: In this case we have to divide all seven marbles over exactly three bags, so we get

$$\left\{ {7 \atop 3} \right\} = 301$$

Exercise 3.AC

(i) Explain by using a combinatorial argument that $\left\{ {n \atop 2} \right\} = 2^{n-1} - 1$ for $n \geq 2$.

Solution: Let $a_1, \ldots, a_n$ be the n objects. And let us label the two boxes with a 0 and a 1. Then for each a_i we have to choose whether we put it in box 0 or box 1. We can represent this by making a table (which looks a lot like our truth tables that we have seen before):

a_1	a_2	a_3	$\ldots$	a_n
0	0	0		0
0	0	0		1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	1		0
1	1	1		1

Each row represents a possible distribution. And we have 2^n rows. However, the first row and the last row are illegal, because everything ends up in only one box. So we get $2^n - 2$ distributions over labeled boxes. However, the boxes weren't labeled! So we have counted every distribution twice! So we get $\frac{2^n-2}{2} = 2^{n-1} - 1$ distributions over indistinguishable boxes.

(ii) Prove by induction that $\left\{ {n \atop 2} \right\} = 2^{n-1} - 1$ for $n \geq 2$.

Solution:

Proposition: 0

$\left\{ {n \atop 2} \right\} = 2^{n-1} - 1$ for all $n \geq 2$.

Proof by induction on n. 1

We first define our predicate P as:

$$P(n) \quad := \quad \left[\left\{ {n \atop 2} \right\} = 2^{n-1} - 1 \right]$$ 2

Base Case. We show that $P(2)$ holds, i.e. we show that 3

$\left\{ {2 \atop 2} \right\} = 2^{2-1} - 1$

This indeed holds, because $\left\{ {2 \atop 2} \right\} = \left\{ {1 \atop 1} \right\} + 2\left\{ {1 \atop 2} \right\} = 1 + 2 \cdot 0 = 1$ and $2^{2-1} - 1 = 2^1 - 1 =$ 4
$2 - 1 = 1$.

Induction Step. Let k be any natural number such that $k \geq 2$. 5

Assume that we already know that $P(k)$ holds, i.e. we assume that 6
$\left\{ {k \atop 2} \right\} = 2^{k-1} - 1$ (Induction Hypothesis IH)

We now have to show that $P(k+1)$ also holds, i.e. we now have to show that 7
$\left\{ {k+1 \atop 2} \right\} = 2^{k+1-1} - 1$

This indeed holds, because 8

$$\begin{aligned}
\left\{ {k+1 \atop 2} \right\} &= \left\{ {k \atop 1} \right\} + 2\left\{ {k \atop 2} \right\} \text{ (by Definition 3.51)} \\
&= 1 + 2\left\{ {k \atop 2} \right\} \text{ (by Definition 3.51)} \\
&= 1 + 2\left(2^{k-1} - 1\right) \text{ (by the IH)} \\
&= 1 + 2^{k+1-1} - 2 \text{ (by elementary algebra)} \\
&= 2^{k+1-1} - 1 \text{ (by elementary algebra)}
\end{aligned}$$

Hence it follows by induction that $P(n)$ holds for all $n \geq 2$. 9

Exercise 3.AD

Explain by using a combinatorial argument that $\left\{ {n \atop n-1} \right\} = \binom{n}{2}$ for $n \geq 2$.

Solution: A combinatorial interpretation of $\left\{ {n \atop n-1} \right\}$ is the number of ways to divide n distinguishable objects over $n-1$ indistinguishable boxes. In particular, this means that two out of the n objects must be together in one box, and all other $n-2$ objects must be on their own in a box. An algorithm to create such a distribution is by choosing the two elements that will be together, glue them together so that there are now $n-1$ distinguishable objects and put each of these objects in an empty box. The first step can be done in $\binom{n}{2}$ ways. The other steps can only be done in one way. So the total number is indeed $\binom{n}{2}$.

Exercise 3.AE
Which of the following sets are partitions of the natural numbers? If it is not a partition, explain why not.

(i) $\{\mathbb{N}\}$
Solution: This is a partition.

(ii) $\{\{37\}, \{42\}, \mathbb{N} \setminus \{37, 42\}\}$
Solution: This is a partition.

(iii) $\{\{x \in \mathbb{N} \mid x \text{ is prime}(x)\}, \{x \in \mathbb{N} \mid x \text{ is not prime}\}\}$
Solution: This is a partition.

(iv) $\{\{1\}, \{2\}, \{3\}, \ldots\}$
Solution: This is not a partition, because $0 \in \mathbb{N}$ but there is no subset that contains 0.

(v) $\{\{0, 1\}, \{1, 2\}, \{2, 3\}, \ldots\}$
Solution: This is not a partition, because $2 \in \mathbb{N}$ but $2 \in \{1, 2\}$ and $2 \in \{2, 3\}$ which violates the rule that every element of $\mathbb{N}$ is in exactly one subset.

(vi) $\{\{x \in \mathbb{N} \mid x \text{ is a multiple of } 2\}, \{x \in \mathbb{N} \mid x \text{ is a multiple of } 3\},$
$\{x \in \mathbb{N} \mid x \text{ is not a multiple of 2 and } x \text{ is not a multiple of } 3\}\}$
Solution: This is not a partition, because $12 \in \mathbb{N}$, and 12 is both a multiple of 2 and 3, so it is in two subsets.

Exercise 3.AF

(i) The triangle of the Stirling numbers of the *first* kind has as its top:

$$\begin{array}{cccccccccccc} &&&&&1&&&&& \\ &&&&0&&1&&&& \\ &&&0&&1&&1&&& \\ &&0&&2&&3&&1&& \\ &0&&6&&11&&6&&1& \\ 0&&24&&50&&35&&10&&1 \end{array}$$

Calculate two more rows in this triangle.
Solution:

$$\begin{array}{ccccccccccccccc} &&&&&&&1&&&&&&& \\ &&&&&&0&&1&&&&&& \\ &&&&&0&&1&&1&&&&& \\ &&&&0&&2&&3&&1&&&& \\ &&&0&&6&&11&&6&&1&&& \\ &&0&&24&&50&&35&&10&&1&& \\ &0&&120&&274&&225&&85&&15&&1& \\ 0&&720&&1764&&1624&&735&&175&&21&&1 \end{array}$$

(ii) Calculate the sum of the numbers in each row of the triangle you just calculated. Can you guess a formula for these sums?

Solution: We get:

$$\begin{array}{ccccccccccccccc}
 & & & & & & 1 & & & & & & & = & 1\\
 & & & & & 0 & + & 1 & & & & & & = & 1\\
 & & & & 0 & + & 1 & + & 1 & & & & & = & 2\\
 & & & 0 & + & 2 & + & 3 & + & 1 & & & & = & 6\\
 & & 0 & + & 6 & + & 11 & + & 6 & + & 1 & & & = & 24\\
 & 0 & + & 24 & + & 50 & + & 35 & + & 10 & + & 1 & & = & 120\\
0 & + & 120 & + & 274 & + & 225 & + & 85 & + & 15 & + & 1 & = & 720
\end{array}$$

And hopefully, you recognize the factorials 0!, 1!, 2!, 3!, 4!, 5!, and 6!, so the formula is probably $n!$.

Exercise 3.AG
Expand the polynomials

$$(1+x)(1+2x)\cdots(1+nx)$$

for $n \in \{0,1,2,3,4\}$. Do you see a relationship with the triangle from the previous exercise?

Solution:

n	$(1+x)(1+2x)\cdots(1+nx)$
0	1
1	$x+1$
2	$2x^2+3x+1$
3	$6x^3+11x^2+6x+1$
4	$24x^4+50x^3+35x^2+10x+1$

This is similar to the Binomial Theorem that we have seen above. We have

$$\underbrace{(1+x)(1+x)\cdots(1+x)}_{n \text{ factors}} = \binom{n}{0}x^n + \binom{n}{1}x^{n-1} + \cdots + \binom{n}{n-1}x + \binom{n}{n}$$

$$\underbrace{(1+x)(1+2x)\cdots(1+nx)}_{n \text{ factors}} = \begin{bmatrix} n+1 \\ 0 \end{bmatrix} x^{n+1} + \begin{bmatrix} n+1 \\ 1 \end{bmatrix} x^n + \cdots + \begin{bmatrix} n+1 \\ n \end{bmatrix} x + \begin{bmatrix} n+1 \\ n+1 \end{bmatrix}$$

Note that the first one is not exactly the same as in our version of Newton's Binomial Theorem (Theorem 3.48): we have to apply the symmetry identity that $\binom{n}{i} = \binom{n}{n-i}$ to get the exact same result. And isn't it weird that for the Stirling version, we need to take the series for $n+1$ instead of n? How can this be correct? Our expansion of $(1+x)(1+2x)\cdots(1+nx)$ doesn't have a term x^{n+1}! If you are worried about this, please recall from Definition 3.52 that $\begin{bmatrix} n \\ 0 \end{bmatrix} = 0$ for all $n > 0$. So in the expansion we see a term of degree $n+1$, but as its coefficient happens to be 0, it simply vanishes.

Exercise 3.AH
There are nice relations between the Stirling numbers of the first and second kind.

(i) Given numbers a_1, a_2, a_3, a_4, we define:

$$\begin{aligned}
b_1 &:= a_1\\
b_2 &:= a_1 + a_2\\
b_3 &:= 2a_1 + 3a_2 + a_3\\
b_4 &:= 6a_1 + 11a_2 + 6a_3 + a_4
\end{aligned}$$

so with coefficients from the triangle of Stirling numbers of the first kind. Now from these numbers b_i we define:

$$
\begin{aligned}
c_1 &:= b_1 \\
c_2 &:= -b_1 + b_2 \\
c_3 &:= b_1 - 3b_2 + b_3 \\
c_4 &:= -b_1 + 7b_2 - 6b_3 + b_4
\end{aligned}
$$

with coefficients from the triangle of Stirling numbers of the second kind, but with the sign alternating between plus and minus.
Express the numbers c_i in terms of the a_i. What do you find?
Solution:

$$
\begin{aligned}
c_1 &= b_1 \\
&= a_1 \\
c_2 &= -b_1 + b_2 \\
&= -a_1 + (a_1 + a_2) \\
&= a_2 \\
c_3 &= b_1 - 3b_2 + b_3 \\
&= a_1 - 3\,(a_1 + a_2) + (2a_1 + 3a_2 + a_3) \\
&= a_1 - 3a_1 - 3a_2 + 2a_1 + 3a_2 + a_3 \\
&= a_3 \\
c_4 &= -b_1 + 7b_2 - 6b_3 + b_4 \\
&= -a_1 + 7\,(a_1 + a_2) - 6\,(2a_1 + 3a_2 + a_3) + (6a_1 + 11a_2 + 6a_3 + a_4) \\
&= -a_1 + 7a_1 + 7a_2 - 12a_1 - 18a_2 - 6a_3 + 6a_1 + 11a_2 + 6a_3 + a_4 \\
&= a_4
\end{aligned}
$$

(ii) Now we reverse the two kinds of Stirling numbers in this relation. If from the numbers c_i from the previous exercise we go on, and define:

$$
\begin{aligned}
d_1 &:= c_1 \\
d_2 &:= c_1 + c_2 \\
d_3 &:= 2c_1 + 3c_2 + c_3 \\
d_4 &:= 6c_1 + 11c_2 + 6c_3 + c_4
\end{aligned}
$$

and then express d_i in terms of b_i, what do we find then?

Solution:

$$
\begin{aligned}
d_1 &= c_1 \\
&= b_1 \\
d_2 &= c_1 + c_2 \\
&= b_1 + (-b_1 + b_2) \\
&= b_2 \\
d_3 &= 2c_1 + 3c_2 + c_3 \\
&= 2b_1 + 3\,(-b_1 + b_2) + (b_1 - 3b_2 + b_3) \\
&= 2b_1 - 3b_1 + 3b_2 + b_1 - 3b_2 + b_3 \\
&= b_3 \\
d_4 &= 6c_1 + 11c_2 + 6c_3 + c_4 \\
&= 6b_1 + 11\,(-b_1 + b_2) + 6\,(b_1 - 3b_2 + b_3) + (-b_1 + 7b_2 - 6b_3 + b_4) \\
&= 6b_1 - 11b_1 + 11b_2 + 6b_1 - 18b_2 + 6b_3 - b_1 + 7b_2 - 6b_3 + b_4 \\
&= b_4
\end{aligned}
$$

Chapter 4

Languages

Exercise 4.A
In Example 4.6 we have seen some examples of language descriptions. Now try describing these languages yourself using formal set notation:

(i) Describe $L_1 \cap L_2$.
Solution: If you don't know much about the representation of sets, please read Section A in the course notes.

We know that L_1 contains all words in $\{a, b\}^*$ with an even number of a's and that L_2 contains all words that start with a certain amount of a's, followed by exactly the same amount of b's. From this it follows that the intersection $L_1 \cap L_2$ is comprised of the words of the shape $a^n b^n$ that contain an even number of a's, so we can write: $L_1 \cap L_2 = \{a^n b^n \mid n \in \mathbb{N} \text{ and } n \text{ is even}\}$. Another way of writing this would be: $L_1 \cap L_2 = \{a^{2n} b^{2n} \mid n \in \mathbb{N}\}$

(ii) Describe $L_2 \cap L_4$.
Solution: We already described which words are in L_2, so we now only give such a description for the words in L_4. Note that L_4 contains all words over the alphabet $\{a, b, c\}$ that are reversible. Hence the intersection $L_2 \cap L_4$ contains all words w of the shape $a^n b^n$ for which $w = w^R$. But $a^n b^n$ is only equal to its reverse $b^n a^n$ if $n = 0$, so that we must always have $w = \lambda$, so that $L_2 \cap L_4 = \{\lambda\}$.

(iii) Describe $L_3 \cap L_4$.
Solution: The words in L_3 all have an odd length and a c in the middle. The w-part before this middle c may contain a's and c's, and it is exactly as long as the v-part after the middle c, which on its turn may contain b's and c's. Note that this c is the same symbol c as in the alphabet and therefore it cannot be a general variable like v and w! $L_3 \cap L_4$ contains all wcv for which: $|w| = |v|$, and w contains no b, and v contains no a, and $wcv = v^R c w^R$. This only holds if w and v are both made up of only c's, so that $L_3 \cap L_4 = \{c^n \mid n \in \mathbb{N} \text{ and } n \text{ is odd}\} = \{c^{2n+1} \mid n \in \mathbb{N}\}$.

Exercise 4.B
Use the definitions of Example 4.6.

(i) Prove that $L_1 = L_1^*$
Solution: To prove that $L_1 = L_1^*$, we show that both $L_1 \subseteq L_1^*$ and $L_1^* \subseteq L_1$. The first inclusion is easy. For every word $w \in L_1$, taking $k = 1$, we have per definition of L_1^* that $w^k = w^1 = w \in L_1^*$, so indeed $L_1 \subseteq L_1^*$. For the second inclusion, let us take a look at a word $w \in L_1^*$. Then by definition of L_1^*, we have $w = w_1 \cdots w_k$, for some $k \geq 0$ and with all $w_i \in L_1$, thus containing an even number of a's. But then of course w has an even number of a's as well, so $w \in L_1$. And thus $L_1^* \subseteq L_1$.

(ii) Does $L_2 L_2 = L_2$ hold? Give a proof, or else a counterexample.
Solution: No, it doesn't hold. $ab \in L_2$, so that $abab \in L_2 L_2$, and obviously $abab \notin L_2$. So $L_2 L_2 \neq L_2$.

(iii) Does $\overline{L_1} = \overline{L_1}^*$ hold? Give a proof, or else a counterexample.

Solution: $\overline{L_1} = \{w \mid w \text{ contains an odd number of } a\text{'s}\}$. The answer then is 'no', because $a \in \overline{L_1}$ and so $aa \in \overline{L_1}^*$ too, whereas obviously $aa \in L_1$, and thus $aa \notin \overline{L_1}$. Hence $\overline{L_1} \neq \overline{L_1}^*$. (Note that $\overline{L_1}^*$ stands for the language where you *first* take the complement and *then* the Kleene closure of L_1.)

(iv) For which languages of Example 4.6 do we have $L = L^R$? (You need only answer, a proof is not necessary.)
Solution: This property holds only for languages L_1 and L_4.
In the case of L_1, reversing a word does not change the amount of a's contained in the word. In the case of L_2 we have $bbaa \in L_2^R$ though $bbaa \notin L_2$. In the case of L_3 we have $bca \in L_3^R$ though $bca \notin L_3$. And finally, in the case of L_4, which contains all palindromes, well, reversed palindromes are still palindromes!

Exercise 4.C

(i) Demonstrate that the operator ?, for which $a^?$ stands for either 0 or 1 times a, doesn't have to be added to the regular expressions, as it can be defined using the existing operators.
Solution: We can define $a^?$ by $a \cup \lambda$.

(ii) What is $\mathcal{L}(\varnothing ab^*)$?
Solution: $\mathcal{L}(\varnothing ab^*) = \mathcal{L}(\varnothing)\mathcal{L}(a)\mathcal{L}(b^*) = \varnothing\{a\}\{b\}^*$, or, the language of words of the shape $w_1w_2w_3$, for which $w_1 \in \varnothing$, $w_2 \in \{a\}$, and $w_3 \in \{b\}^*$. But, as there is no word $w_1 \in \varnothing$, we have $\mathcal{L}(\varnothing ab^*) = \varnothing$, the empty language.

(iii) We define the language $L_5 := \{w \in \{a,b\}^* \mid w \text{ contains at least one } a\}$. Give a regular expression that describes this language.
Solution: $(a \cup b)^* a (a \cup b)^*$. The a that we want to guarantee in every word is the a in the middle of the expression. Before as well as after the guaranteed a, we allow arbitrary sequences of a's and b's (possibly of length 0), these are represented by $(a \cup b)^*$. Note: another possible solution would be $b^*a(a \cup b)^*$, can you explain why?

(iv) Give a regular expression that describes the language L_1 from Example 4.6.
Solution: $(b^*ab^*ab^*)^* \cup b^*$. The idea is that the expression $b^*ab^*ab^*$ describes words with exactly two a's, surrounded by any number of b's in any position. And then taking any number of these words in sequence, $(b^*ab^*ab^*)^*$, we have an expression describing words with an even number of a's. But we still have a problem: words with zero a's but a number of b's, such as bbb, are still excluded. So we add b^* to get $(b^*ab^*ab^*)^* \cup b^*$. There are several other expressions that provide the same language: $(b^*ab^*ab^*)^* b^*$, $b^*(b^*ab^*ab^*)^*$ or $(b \cup ab^*a)^*$. Convince yourself that these expressions produce the same language.

(v) Show that $\mathcal{L}(ab(ab)^*) = \mathcal{L}(a(ba)^*b)$.
Solution: The language $\mathcal{L}(ab(ab)^*)$ contains all words of the shape ab followed by n times ab (with $n \geq 0$). But, ab followed by n times ab is the same as a followed by n times ba, followed by b. And this is exactly $\mathcal{L}(a(ba)^*b)$. A way to prove this more precisely is as follows. As stated, any word of the language $\mathcal{L}(ab(ab)^*)$, has to have the form $ab(ab)^n$, for some $n \in \mathbb{N}$. We can then rewrite this word:

$$\begin{aligned} ab(ab)^n &= ab(ab)\cdots(ab) \\ &= abab\cdots ab \\ &= ababa\cdots bab \\ &= a(ba)(ba)\cdots(ba)b \\ &= a(ba)^n b \end{aligned}$$

resulting in a word that is by definition in $\mathcal{L}(a(ba)^*b)$. Hence $\mathcal{L}(ab(ab)^*) \subseteq \mathcal{L}(a(ba)^*b)$. The proof that $\mathcal{L}(a(ba)^*b) \subseteq \mathcal{L}(ab(ab)^*)$ is analogous.
Note, though, that this general proof, using $\cdots$, actually only holds for $n \geq 3$. Check that such a rewrite is also possible for $n = 0, 1, 2$ (without using $\cdots$).

For those especially formally inclined, who don't like proof sketches with $\cdots$, the proof method of induction will be explained fully in Chapter 3. Put briefly, an induction proof that some property holds for a large number of objects, has two parts: a part that proves the property for an initial object (the base case), and a part that proves that then this proof can be carried on to further objects (the induction step). A proof for the above statement, using induction, would look like this (you don't need to fully understand this).

We want to prove that $ab\,(ab)^n = a\,(ba)^n\,b$ for all $n \in \mathbb{N}$. Because from this it follows that $\mathcal{L}\,(ab\,(ab)^*) \subseteq \mathcal{L}\,(a\,(ba)^*\,b)$. We use induction on n.

Base case. The smallest word has $n = 0$ and is simply ab. Now we have to prove that $ab\,(ab)^0 = a\,(ab)^0\,b$. This is trivial since $ab\,(ab)^0 = ab = a\,(ab)^0\,b$.

Induction step. Suppose now, that we have already proved that $ab\,(ab)^n = a\,(ba)^n\,b$ for some certain $n \in \mathbb{N}$. This is called our induction hypothesis (IH). Now we get to prove that $ab\,(ab)^{n+1} = a\,(ba)^{n+1}\,b$ holds as well. Rewrite as follows:

$$\begin{aligned} ab\,(ab)^{n+1} &= ab\,(ab)^n\,(ab) \\ &= ab\,(ab)^n\,ab \\ &\overset{\text{IH}}{=} a\,(ba)^n\,bab \\ &= a\,(ba)^n\,(ba)\,b \\ &= a\,(ba)^{n+1}\,b \end{aligned}$$

The equality marked with IH is seen to be true by applying the induction hypothesis to the front part of the word.

Wrapping up the induction proof. The base case tells us that the property $ab\,(ab)^n = a\,(ba)^n\,b$ holds for $n = 0$. The induction step then proves that it holds for $n = 1$ as well. And we can repeat the induction step as well, so that the property holds for $n = 2$, too, and in fact any number $n = 3, 4, 5, \ldots$. But then we indeed have a proof of $\mathcal{L}\,(ab\,(ab)^*) \subseteq \mathcal{L}\,(a\,(ba)^*\,b)$. The proof of the reverse inclusion is of course analogous.

Exercise 4.D

Show that the following languages are regular.

(i) $L_6 := \{w \in \{a, b\}^* \mid \text{every } a \text{ in } w \text{ is directly followed by a } b\}$,
Solution: $(b \cup ab)^*$ is a regular expression for L_6. (Note: a word in L_6 doesn't necessarily contain an a.) Another solution would be $(b^*\,(ab)^*)^*$. The expression $(b^*ab)^*$ is very similar, but not correct. Can you think of a word of L_6 that can't be described by this last expression?

(ii) $L_7 :=$ the language of all well-formed natural numbers. These words (numbers) are made up of the symbols $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, but they never start with a 0, except for the word 0 of course.
Solution: A possible regular expression for L_7 is

$$r_1 := (1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)\,(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^* \cup 0$$

(iii) $L_8 :=$ the language of all well-formed integers These words (integers) are made up of the natural numbers (for which you defined a regular expression in the previous question), possibly preceded by a $+$ or $-$ sign. [*Hint:* If you named the expression in the previous question, you may reuse it here.]
Solution: A possible regular expression for L_8 is

$$r_2 := (+ \cup - \cup \lambda)\,(r_1)$$

where r_1 is the expression we defined in the previous question. Do you think the parentheses around r_1 are needed? In other words, if you replace the 'abbreviation' r_1 by the literal expression defined earlier, will the expression r_2 generate exactly the same language as the expression $r_2' := (+ \cup - \cup \lambda)\, r_1$? [*Hint:* Look at the binding strength of the operators used!]

(iv) L_9 := the language of all well-formed arithmetical expressions without parentheses. These contain all natural numbers, possibly interspersed with the operators $+$, $-$, and $\times$, as in for example $7 + 3 \times 29 - 78$.
Solution: A regular expression for L_9 is

$$r_3 := (r_1\,(+ \cup - \cup \times))^*\, r_1$$

Note that this expression also generates expressions without any operator, for example 37. If you want to avoid these, you could unfold the * one step:

$$(r_1\,(+ \cup - \cup \times))^*\, r_1\,(+ \cup - \cup \times)\, r_1$$

Exercise 4.E
Which of the following regular expressions describe the same language? For any two expressions, either show that they describe the same language, or else give a word that exemplifies that this is not the case.
(i) $b^*\,(aab)^*$.
(ii) $b^*\,(baa)^*\,b$.
(iii) $bb^*\,(aab)^*$.

Solution: Let us start by giving names to the corresponding languages: $L := \mathcal{L}\,(b^*\,(aab)^*)$, $L' := \mathcal{L}\,(b^*\,(baa)^*\,b)$, and $L'' := \mathcal{L}\,(bb^*\,(aab)^*)$.
(i) $L \neq L'$, because λ is in L and not in L'.
(ii) $L \neq L''$, because λ is in L and not in L''.
(iii) $L' = L''$, because:

- Any $w \in L'$, has the shape $w = b \dots b\, baa\, \dots\, baa\, b$, with n times b and m times baa where $n \geq 0$ and $m \geq 0$. If $m = 0$ we get n b's from the first part and one b from the third part. So we simply have $n + 1$ b's, which is clearly in L''. And if $m > 0$, we can also read w as $b \dots b\, b\, aab\, \dots\, aab$, with $n + 1$ times b and m times aab, and so we see that $w \in L''$.
  In rewriting rules:

$$\begin{aligned} w &= b^n\,(baa)^m\,b \\ &= b^n\,(baa)\cdots(baa)\,b \\ &= b^n baa \cdots baab \\ &= b^n baab \cdots aabaab \\ &= bb^n aab \cdots aabaab \\ &= bb^n\,(aab)\cdots(aab)\,(aab) \\ &= bb^n\,(aab)^m \in L'' \end{aligned}$$

  Here, as before, the general rewriting proof only holds for $m \geq 3$. Show yourself that the statement also holds for $m = 1$ and $m = 2$.
- If $w \in L''$, then w has the shape $w = b\, b \dots b\, aab\, \dots\, aab$, with $n + 1$ times b and m times aab where $n \geq 0$ and $m \geq 0$. If $m = 0$ we automatically have $w \in L'$. If $m > 0$, we can also read w as $b \dots b\, baa\, \dots\, baa\, b$, with n times b and m times baa, and so $w \in L'$.
  Rewriting rules are analogous to the other direction, above.

Exercise 4.F
The best way to show that a language is context-free is by giving a context-free grammar.

(i) Show that the language of *balanced parentheses expressions* is context-free. By this, we mean the expressions over $\{(,)\}$ where every opening parenthesis is closed with a parenthesis as well, so for example $(()(()))$ and $(())()$ are balanced, but $(()()))$ is not.
Solution: Balanced parentheses expressions such as $(()(()))$ and $(())()$, are generated by the grammar:

$$S \rightarrow SS \mid (S) \mid \lambda$$

or, alternatively:

$$S \rightarrow (S)S \mid \lambda$$

(ii) Show that the language L_1 from Example 4.6 is context-free.
Solution: L_1 is generated by:

$$\begin{aligned} S &\rightarrow bS \mid aT \mid \lambda \\ T &\rightarrow bT \mid aS \end{aligned}$$

(iii) Show that the language L_2 from Example 4.6 is context-free. (Check Example 4.16.)
Solution: L_2:

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

(iv) Show that the language L_3 from Example 4.6 is context-free.
Solution: L_3:

$$\begin{aligned} S &\rightarrow ASB \mid c \\ A &\rightarrow a \mid c \\ B &\rightarrow b \mid c \end{aligned}$$

This grammar builds up the words from the inside and only places the middle c in the last step. (If you put the middle c in place earlier, you lose control of keeping the lengths of w and v the same.)

(v) Show that the language L_4 from Example 4.6 is context-free.
Solution: L_4:

$$S \rightarrow aSa \mid bSb \mid cSc \mid a \mid b \mid c \mid \lambda$$

Here, again, we build up the words from the inside.

Exercise 4.G
Consider the grammar G_2:

$$\begin{aligned} S &\rightarrow AS \mid Sb \mid \lambda \\ A &\rightarrow aA \mid \lambda \end{aligned}$$

(i) Write G_2 as a triple $\langle \Sigma, V, R \rangle$.
Solution:

$$G_2 = \langle \{a, b\}, \{S, A\}, \{S \rightarrow AS, S \rightarrow Sb, S \rightarrow \lambda, A \rightarrow aA, A \rightarrow \lambda\} \rangle$$

(ii) Give a production demonstrating that $aabb \in \mathcal{L}(G_2)$.
Solution: Multiple solutions are possible. Here are two:

$$S \rightarrow AS \rightarrow aAS \rightarrow aaAS \rightarrow aaS \rightarrow aaSb \rightarrow aaSbb \rightarrow aabb$$
$$S \rightarrow AS \rightarrow ASb \rightarrow ASbb \rightarrow Abb \rightarrow aAbb \rightarrow aaAbb \rightarrow aabb$$

(iii) Can you give a production demonstrating that $bbaa \in \mathcal{L}(G_2)$ within three minutes?
Solution: Probably not, because $bbaa \notin \mathcal{L}(G_2)$. We aren't able to prove this yet, but will do so in a later exercise.

Exercise 4.H
Consider the following grammar G_3 for the language L_{13}.

$$\begin{aligned} S &\rightarrow aSb \mid A \mid \lambda \\ A &\rightarrow aAbb \mid abb \end{aligned}$$

The nonterminals are S and A, and $\Sigma = \{a, b\}$.

(i) Give productions of abb and $aabb$.
Solution: Produce abb via $S \rightarrow A \rightarrow abb$. Produce $aabb$ via $S \rightarrow aSb \rightarrow aaSbb \rightarrow aabb$.

(ii) Which words does L_{13} contain?
Solution:

- λ, because of the rule $S \rightarrow \lambda$
- words of the shape $a^n S b^n$, for $n \in \mathbb{N}$, produced by the rule $S \rightarrow aSb$
- the nonterminal A produces words of the shape $a^m b^{2m}$ with $m \in \mathbb{N}$ and $m \geq 1$: abb, $aabbbb$, $aaabbbbbb$, ... This is easily seen, because the production rules containing A only use A itself, and don't recurse back to S.
- the rule $S \rightarrow A$ allows one to replace S by something that A produces (words of the form $a^m b^{2m}$). So then $a^n S b^n$ produces $a^n a^m b^{2m} b^n$ with $n, m \in \mathbb{N}$ and $m \geq 1$. This can be simplified to $a^{n+m} b^{n+2m}$ with $n, m \in \mathbb{N}$ and $m \geq 1$.

Taken together, we get:

$$L_{13} = \{a^{n+m} b^{n+2m} \mid n, m \in \mathbb{N}\}$$

The condition that $m \geq 1$ is eliminated because you can also apply the rule $S \rightarrow A$ zero times.
This can also be simplified to

$$L_{13} := \{a^k b^{k+l} \mid k \geq 0 \text{ and } k \geq l \geq 0\}.$$

Or even to

$$L_{13} := \{a^k b^{k+l} \mid k \geq l \geq 0\}.$$

Exercise 4.I
We take another look at the grammar G_2 from Exercise 4.G. It was then already noted that $bbaa \notin \mathcal{L}(G_2)$. But can we prove that using an invariant? Let us give it a try with the predicate

$$P(w) := [\, w \text{ does not contain } ba \text{ as sub-word} \,]$$

At first this may seem like an invariant, but it isn't. Note that $P(bA)$ holds, because bA does not contain ba as sub-word. And note that applying the rule $A \rightarrow aA$ we get the production $bA \rightarrow baA$ and it is clear that $P(baA)$ does not hold. So P is not an invariant. But maybe we can try to fix this by putting more requirements in our predicate P...

(i) Is

$$P(w) := [\, w \text{ does neither contain } ba, \text{ nor } bA, \text{ nor } Sa \text{ as sub-word} \,]$$

an invariant for G_2 that proves that $bbaa \notin \mathcal{L}(G_2)$?
Solution: No. The counterexample of the previous item doesn't work anymore, because $P(bA)$ doesn't hold, but there is another counterexample: we now have that $P(bS)$ holds, but we don't have that $P(bAS)$ holds, although we can apply the production rule $S \rightarrow AS$.

(ii) Is

$$P(w) := [\, w \text{ does neither contain } ba, \text{ nor } bA, \text{ nor } Sa, \text{ nor } bS \text{ as sub-word} \,]$$

an invariant for G_2 that proves that $bbaa \notin \mathcal{L}(G_2)$?

Solution: No. There are still more counterexamples:

- Let $v = SS$ and $v' = SbS$. Then clearly $P(SS)$ holds and $SS \to SbS$ holds because of the rule $S \to Sb$, but $P(SbS)$ does not hold.
- Let $v = SA$ and $v' = SaA$. Then clearly $P(SA)$ holds and $SA \to SaA$ holds because of the rule $A \to aA$, but $P(SaA)$ does not hold.

Exercise 4.J

Use invariants to prove that:

(i) $bba \notin L_{13}$.

Solution: We have the invariant

$$P(w) := [\, w = \lambda, \text{ or } w = S, \text{ or } w \text{ begins with an } A \text{ or an } a \,]$$

We will give a very explicit proof, to demonstrate how to go about such a proof. First, let us number the production rules:

$$\begin{aligned} S &\overset{1}{\to} aSb \\ S &\overset{2}{\to} A \\ S &\overset{3}{\to} \lambda \\ A &\overset{4}{\to} aAbb \\ A &\overset{5}{\to} abb \end{aligned}$$

It should be clear that $P(S)$ holds. So it remains to be proved that the property is indeed invariant under the production rules. That is, if $P(v)$ holds for some v and we have $v \to v'$, then we must also have $P(v')$. Because our invariant has the shape "w has one of these four different forms," this means we have to make a case distinction.

- $v = \lambda$. For this case, there is no corresponding production rule, so there is nothing to check. Specifically, for all v' (of which there are none), we have $P(v')$.
- $v = S$. For this case, we have to check the first three production rules. It is indeed easily seen that for these three production rules, we end up with v' being either aSb, or A, or λ, and for any of these we have $P(v')$. So we are done with this case as well.
- v begins with an A. Now, we have to make even more case distinctions, because v may contain more nonterminals. We write all these sub-cases out below, followed by their possible next step productions. By $v = A\ldots$, we denote the case that v begins with an A, and in the rest of v there are no nonterminals. By $v = A\ldots S\ldots$, we denote the case that v starts with an A, and the rest of v contains as least one

nonterminal S.

$$\begin{aligned} A\ldots &\xrightarrow{4} aAbb\ldots \\ A\ldots &\xrightarrow{5} abb \\ A\ldots S\ldots &\xrightarrow{1} A\ldots aSb\ldots \\ A\ldots S\ldots &\xrightarrow{2} A\ldots A\ldots \\ A\ldots S\ldots &\xrightarrow{3} A\ldots \lambda\ldots \\ A\ldots S\ldots &\xrightarrow{4} aAbb\ldots S\ldots \\ A\ldots S\ldots &\xrightarrow{5} abb\ldots S\ldots \\ A\ldots A\ldots &\xrightarrow{4} aAbb\ldots A\ldots \\ A\ldots A\ldots &\xrightarrow{4} A\ldots aAbb\ldots \\ A\ldots A\ldots &\xrightarrow{5} abb\ldots A\ldots \\ A\ldots A\ldots &\xrightarrow{5} A\ldots abb\ldots \end{aligned}$$

Now we can easily see that indeed any possible resulting v' either begins with an a, or an A, and thus we always have $P(v')$.

- v begins with an a. We make a similar further case distinctions as before:

$$\begin{aligned} a\ldots S\ldots &\xrightarrow{1} a\ldots aSb\ldots \\ a\ldots S\ldots &\xrightarrow{2} a\ldots A\ldots \\ a\ldots S\ldots &\xrightarrow{3} a\ldots \lambda\ldots \\ a\ldots A\ldots &\xrightarrow{4} a\ldots aAbb\ldots \\ a\ldots A\ldots &\xrightarrow{5} a\ldots abb\ldots \end{aligned}$$

Here, too, we see that any resulting v' begins with an a and therefore $P(v')$.

So now, we have proved that P is indeed an invariant for the grammar! And because $P(bba)$ does not hold, it automatically follows that $bba \notin L_{13}$.
As stated above, this is a rather explicitly written and lengthy proof. This is for demonstration purposes, and of course shorter proofs will often suffice. For the case that v begins with an a, you may for example simply state something like: "there is no production rule that changes the first letter, and so v' automatically also begins with an a."

(ii) *aabbb* is not produced by the grammar of L_3 that you constructed in Exercise 4.F.
Solution: An invariant here would be

$$P(w) := [\, w \text{ contains an } S \text{ or a } c\,]$$

Again, it is clear that $P(S)$ holds. Furthermore, P is indeed invariant under the production rules. Indeed, the rules for A and B don't change the number of S's and don't decrease the number of c's. And the rules for S either preserve the number of S's, or a c is added. Also, it is clear that *aabbb* does neither contain an S nor a c. So we have the falsity of $P(aabbb)$ and therefore $aabbb \notin L_3$.

(iii) *aabbb* is not produced by the grammar for L_4 that you constructed in Exercise 4.F.
Solution: An obvious invariant would be

$$P(w) := [\,\text{the first symbol of } w \text{ is the same as the last symbol of } w\,]$$

Indeed $P(S)$. However, contradictory to expectation, this property is not preserved by the production rules. Suppose for example $v = SS$. Then $P(v)$ holds, but we have the production $SS \to Sc$ and it is not true that $P(Sc)$ holds. So here, we note that the

invariant really must be preserved for all words v, and in particular, also for words such as $v = SS$ that won't even ever be generated by the grammar!
The essential idea of the proposed invariant isn't bad, so we only have to extend it to avoid problems as the one mentioned. Take

$$P(w) \quad := \quad \left[\begin{array}{c} \text{the first symbol of } w \text{ is the same as the last symbol of } w, \\ \text{and } w \text{ contains at most one } S \end{array} \right]$$

It is clear that $P(S)$ holds. For the production rules, we make the following case distinction.

- v does not contain an S. Simple: there are no applicable production rules, so vacuously we have that $P(v')$ for all v' for which $v \to v'$.
- v contains exactly one S in the middle of v. So then $v = xySzx$, with $x \in \{a, b, c\}$ and $y, z \in \{a, b, c\}^*$. Now the following productions are possible:

$$\begin{array}{rcl} xySzx & \to & xyaSazx \\ xySzx & \to & xybSbzx \\ xySzx & \to & xycSczx \\ xySzx & \to & xyazx \\ xySzx & \to & xybzx \\ xySzx & \to & xyczx \end{array}$$

  It is easily seen that $P(v')$ holds for all these cases. So indeed P is an invariant of the grammar. Also, $P(aabbb)$ does not hold, and thus indeed $aabbb \notin L_4$.

Exercise 4.K

(i) Consider the following grammar over the alphabet $\{a, b, c\}$:

$$\begin{array}{rcl} S & \to & A \mid B \\ A & \to & abS \mid \lambda \\ B & \to & bcS \mid \lambda \end{array}$$

Check whether you can produce these words with the grammar: *abab*, *bcabbc*, *abba*. If you can, provide a production. If not, provide an invariant which you could use to prove that the word can not be produced.
Solution: *abab* and *bcabbc* can be produced:

$$S \to A \to abS \to abA \to ababS \to ababA \to abab$$
$$S \to B \to bcS \to bcA \to bcabS \to bcabB \to bcabbcS \to bcabbcB \to bcabbc$$

The word *abba* can not be produced.
Let us consider the property

$$P(w) := [\text{the last symbol of } w \text{ that is not a nonterminal, is not an } a]$$

This is an invariant. $P(S)$ is vacuously true as there is no last symbol that is not a nonterminal in S. Now suppose we have a v for which $P(v)$ holds. Then the last symbol that isn't a nonterminal, should be a b or a c. We make a case distinction:

- There are only nonterminals in v. Then we have to check two cases:
  - If for $v \to v'$ we apply any of the rules $S \to A$, $S \to B$, $A \to \lambda$, or $B \to \lambda$, then v' has still only nonterminals (or is equal to λ) and $P(v')$ holds.

  - And in case we apply the rule $A \to abS$ or $B \to bcS$ then there is a last symbol in v' which is not a nonterminal, respectively b and c. So in neither case it is an a. So $P(v')$ also holds.

  So in all cases $P(v')$ holds.

- There is indeed a last symbol that isn't a nonterminal. Call it x. Because $P(v)$ holds, $x = b$ or $x = c$. We then further distinguish according to the nonterminals possibly present in v and possible further productions.
  - If there is a nonterminal in front of x being replaced, then it doesn't matter which production rule is used, because x will remain to be the last symbol that is not a nonterminal.
  - If there is an A after x being replaced, then two productions are possible:

$$\begin{array}{lcl} \ldots x \ldots A \ldots & \to & \ldots x \ldots abS \ldots \\ \ldots x \ldots A \ldots & \to & \ldots x \ldots \lambda \ldots \end{array}$$

    Because the $\ldots$ after the x can contain only nonterminals, the first case makes the b right in front of the S the new last terminal, and indeed not an a. In the second case, the x remains to be the last terminal yet again. So in both cases we have $P(v')$.
  - A similar scenario with two cases occurs for the case that there is a B after x. In the first case, the last terminal becomes a c, and in the other it remains to be x. So here too, $P(v')$ holds in any case.
  - If there is an S after x, there are also two possible cases, but both only give more nonterminals, so $P(v')$ holds vacuously.

So P is indeed an invariant. However, the last terminal of $abba$ is an a and thus $P(abba)$ is not true; so indeed $abba$ cannot be produced.

Now consider a different property:

$$P'(w) := \text{any } a \text{ in } w \text{ is directly followed by a } b$$

This is also an invariant. Again, $P'(S)$ holds vacuously as S does not contain any a. Now suppose we have a v for which $P'(v)$ holds and a v' such that $v \to v'$. Then either v has no a at all, or v' contains at least one a and all a's in v are directly followed by a b. We make a case distinction:

- If v does not contain any a, then the only interesting production rule is the rule $A \to abS$. However, although this production introduces an a, it also arranges that this a is always directly followed by a b. So in this case $P'(v')$ holds. Because all other production rules do not introduce any a's, v' will also not have any a's and hence $P'(v')$ holds trivially.
- Now suppose v contains at least one a and all a's are immediately followed by a b. In general, there are two ways to prevent $P'(v')$ from holding: introduce a new a which is not directly followed by a b, or replace a non-terminal which is located directly after some a by something which is not a b. However, as by assumption each a is directly followed by a b, there are no non-terminals directly after any a, so this case does not occur. And the other case we have already discussed above: the only rule that introduces a new a automatically also introduces a new b directly after it. So also in these situations $P'(v')$ holds.

So P' is indeed also an invariant. And it can be used to prove that $abba$ is not in the language, as $P'(abba)$ clearly doesn't hold since the last a is not directly followed by a b.

So in particular it is possible that there exist several invariants that can be used to prove that a certain word is not in the language.

(ii) Describe the regular language L_{15} that this grammar generates, with a regular expression.
Solution: L_{15} is also produced by the regular expression $((ab) \cup (bc))^*$, that is, $L_{15} = \mathcal{L}\big(\, ((ab) \cup (bc))^* \,\big)$.

(iii) Construct a right linear grammar for the language L_{16} consisting of all words of the shape $ab \ldots aba$ (that is, words with alternating a's and b's, starting and ending with an a; make sure to also include the word a).
Solution: A possible solution would be

$$S \rightarrow abS \mid a$$

Another possibility:

$$\begin{aligned} S &\rightarrow aB \\ B &\rightarrow bS \mid \lambda \end{aligned}$$

Exercise 4.L
Give right linear grammars for the languages of Exercise 4.D:

(i) $L_6 := \{w \in \{a, b\}^* \mid \text{every } a \text{ in } w \text{ is directly followed by a } b\}$,
Solution:

$$S \rightarrow abS \mid bS \mid \lambda$$

(ii) L_8 := the language of well-formed integer expressions. These consist of the symbols $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, but never start with a 0, except for the word 0 itself, and may be preceded by a $+$ or a $-$ sign.
Solution:

$$\begin{aligned} S &\rightarrow +T \mid -T \mid T \\ T &\rightarrow 0 \mid 1C \mid 2C \mid 3C \mid 4C \mid 5C \mid 6C \mid 7C \mid 8C \mid 9C \\ C &\rightarrow 0C \mid 1C \mid 2C \mid 3C \mid 4C \mid 5C \mid 6C \mid 7C \mid 8C \mid 9C \mid \lambda \end{aligned}$$

(iii) L_9 := the language of well-formed arithmetical expressions without parentheses. These consist of natural numbers, interspersed with the operators $+$, $-$, and $\times$, as in for example $7 + 3 \times 29 - 78$.
Solution:

$$\begin{aligned} S &\rightarrow 0 \mid 0A \mid 1C \mid 2C \mid 3C \mid 4C \mid 5C \mid 6C \mid 7C \mid 8C \mid 9C \\ C &\rightarrow 0C \mid 1C \mid 2C \mid 3C \mid 4C \mid 5C \mid 6C \mid 7C \mid 8C \mid 9C \mid \lambda \mid A \\ A &\rightarrow +S \mid -S \mid \times S \end{aligned}$$

Note that this solution also allows for simply natural numbers without any operations. Can you construct a grammar that does not allow this, so that at least one operator must be included in every word?

Exercise 4.M
Here we give a grammar for a small part of the English language.

$$
\begin{array}{rcl}
S \;=\; \langle\text{sentence}\rangle & \to & \langle\text{subjectpart}\rangle\langle\text{verbpart}\rangle\textbf{.} \\
\langle\text{sentence}\rangle & \to & \langle\text{subjectpart}\rangle\langle\text{verbpart}\rangle\langle\text{objectpart}\rangle\textbf{.} \\
\langle\text{subjectpart}\rangle & \to & \langle\text{name}\rangle \mid \langle\text{article}\rangle\langle\text{noun}\rangle \\
\langle\text{name}\rangle & \to & \textbf{John} \mid \textbf{Jill} \\
\langle\text{noun}\rangle & \to & \textbf{bicycle} \mid \textbf{mango} \\
\langle\text{article}\rangle & \to & \textbf{a} \mid \textbf{the} \\
\langle\text{verbpart}\rangle & \to & \langle\text{verb}\rangle \mid \langle\text{adverb}\rangle\langle\text{verb}\rangle \\
\langle\text{verb}\rangle & \to & \textbf{eats} \mid \textbf{rides} \\
\langle\text{adverb}\rangle & \to & \textbf{slowly} \mid \textbf{frequently} \\
\langle\text{adjectives}\rangle & \to & \langle\text{adjective}\rangle\langle\text{adjectives}\rangle \mid \lambda \\
\langle\text{adjective}\rangle & \to & \textbf{big} \mid \textbf{juicy} \mid \textbf{yellow} \\
\langle\text{objectpart}\rangle & \to & \langle\text{adjectives}\rangle\langle\text{name}\rangle \\
\langle\text{objectpart}\rangle & \to & \langle\text{article}\rangle\langle\text{adjectives}\rangle\langle\text{noun}\rangle
\end{array}
$$

(i) Is this grammar right linear?
Solution: The grammar is context-free, but not right linear. For example, the first production rule is not right linear.

(ii) Show how you produce the following sentence: *Jill frequently eats a big juicy yellow mango.*
Solution:

- <sentence>
- →<subjectpart><verbpart><objectpart>. (R2)
- →<name><verbpart><objectpart>. (R3.1)
- →Jill<verbpart><objectpart>. (R4.2)
- →Jill<adverb><verb><objectpart>. (R7.2)
- →Jill frequently <verb><objectpart>. (R9.2)
- →Jill frequently eats <objectpart>. (R8.1)
- →Jill frequently eats <article><adjectives><noun>. (R13)
- →Jill frequently eats a <adjectives><noun>. (R6.1)
- →Jill frequently eats a <adjective><adjectives><noun>. (R10.1)
- →Jill frequently eats a big <adjectives><noun>. (R11.1)
- →Jill frequently eats a big <adjective><adjectives><noun>. (R10.1)
- →Jill frequently eats a big juicy <adjectives><noun>. (R11.2)
- →Jill frequently eats a big juicy <adjective><adjectives><noun>. (R10.1)
- →Jill frequently eats a big juicy yellow <adjectives><noun>. (R11.3)
- →Jill frequently eats a big juicy yellow <noun>. (R10.2)
- →Jill frequently eats a big juicy yellow mango. (R5.2)

(iii) Make some more sentences.
Solution: Do this yourself.

Exercise 4.N
Consider the grammar G_5:

$$
\begin{array}{rcl}
S & \to & AB \mid BCS \\
A & \to & aA \mid C \\
B & \to & bbB \mid b \\
C & \to & cC \mid \lambda
\end{array}
$$

(i) Give all the words of $\mathcal{L}(G_5)$ that have a length not exceeding three.

Solution: These are productions that lead to words of length not exceeding three:

$$
\begin{array}{l}
S \to AB \to CB \to B \to b \\
S \to AB \to aAB \to aCB \to aB \to ab \\
S \to AB \to CB \to cCB \to cB \to cb \\
S \to AB \to aAB \to aaAB \to aaCB \to aaB \to aab \\
S \to AB \to aAB \to aCB \to acCB \to acB \to acb \\
S \to AB \to CB \to B \to bbB \to bbb \\
S \to AB \to CB \to cCB \to ccCB \to ccB \to ccb \\
S \to BCS \to BS \to bS \to bAB \to bCB \to bB \to bb \\
S \to BCS \to BS \to bS \to bAB \to baAB \to baCB \to baB \to bab \\
S \to BCS \to BS \to bS \to bAB \to bCB \to bcCB \to bcB \to bcb
\end{array}
$$

(ii) What are the nullable non-terminals in G_5?
Solution: It is not difficult to see that S and B are not nullable, but C and A are, as

$$
\begin{array}{l}
C \to \lambda; \\
A \to C \to \lambda.
\end{array}
$$

(iii) Provide an equivalent grammar G_5' that has no λ-rules.
Solution: Knowing that A and C are nullable, we have to add the following rules:

$$S \to B; \quad S \to BS; \quad A \to a; \quad A \to \lambda; \quad C \to c$$

And removing the λ-rules gives:

$$
\begin{array}{lcl}
S & \to & AB \mid BCS \mid B \mid BS \\
A & \to & aA \mid C \mid a \\
B & \to & bbB \mid b \\
C & \to & cC \mid c
\end{array}
$$

And this is just a check that we can still produce the words that we found earlier:

$$
\begin{array}{l}
S \to B \to b \\
S \to AB \to aB \to ab \\
S \to AB \to CB \to cB \to cb \\
S \to AB \to aAB \to aaB \to aab \\
S \to AB \to aAB \to aCB \to acB \to acb \\
S \to B \to bbB \to bbb \\
S \to AB \to CB \to cCB \to ccB \to ccb \\
S \to BS \to bS \to bB \to bB \to bb \\
S \to BS \to bS \to bAB \to baB \to bab \\
S \to BS \to bS \to bAB \to bCB \to bcB \to bcb
\end{array}
$$

So they are all still there and with shorter productions.

Exercise 4.O
Consider the grammar G_6:

$$
\begin{array}{lcl}
S & \to & AS \mid A \\
A & \to & aA \mid bB \mid C \\
B & \to & bB \mid b \\
C & \to & cC \mid B
\end{array}
$$

(i) Give productions for the words b, ab, bb, acb, bab, cab, or explain why such a production doesn't exist.
Solution: These are productions for the first five words:

$$S \rightarrow A \rightarrow C \rightarrow B \rightarrow b$$
$$S \rightarrow A \rightarrow aA \rightarrow aC \rightarrow aB \rightarrow ab$$
$$S \rightarrow A \rightarrow bB \rightarrow bb$$
$$S \rightarrow A \rightarrow aA \rightarrow aC \rightarrow acC \rightarrow acB \rightarrow acb$$
$$S \rightarrow AS \rightarrow CS \rightarrow BS \rightarrow bS \rightarrow bA \rightarrow baA \rightarrow baC \rightarrow baB \rightarrow bab$$

There is no production for cab, as the only way to get a c is by using $C \rightarrow cC$. However, if we expand this C, we either get ccC, or cB, which in turn can go to cbB or cb. In all of these situations, it is clear that the next terminal after c cannot be an a.

(ii) Provide an equivalent grammar G_6' that has no chain rules.
Solution: First we determine the chains. The chain of S is $\{S, A, C\}$, the chain of A is $\{A, C\}$, the chain of B is $\{B\}$, and the chain of C is $\{C\}$. After adding the new rules according to the algorithm, and removing the chain rules we get

$$\begin{aligned} S &\rightarrow AS \mid aA \mid bB \mid b \mid cC \\ A &\rightarrow aA \mid bB \mid b \mid cC \\ B &\rightarrow bB \mid b \\ C &\rightarrow cC \mid bB \mid b \end{aligned}$$

(iii) Check for the words in (i) that they have a production with grammar G_6 if and only if they have a production with grammar G_6'.
Solution: These are productions for the first five words:

$$S \rightarrow b$$
$$S \rightarrow A \rightarrow aA \rightarrow ab$$
$$S \rightarrow bB \rightarrow bb$$
$$S \rightarrow aA \rightarrow acC \rightarrow acb$$
$$S \rightarrow AS \rightarrow bS \rightarrow bA \rightarrow baA \rightarrow bab$$

There is still no production for cab, as the only way to get a c is by using $C \rightarrow cC$. However, if we expand this C, we either get ccC, cbB, or cb. In all of these situations, it is clear that the next terminal after c cannot be an a.

Exercise 4.P
Consider the grammar $G_8 = \langle \{a, b\}, \{S, A, B, C, D, E, F, G\}, R \rangle$ where R is given by:

$$\begin{aligned} S &\rightarrow aA \mid BD \\ A &\rightarrow aA \mid aAB \mid aD \\ B &\rightarrow aB \mid aC \mid BF \\ C &\rightarrow Bb \mid aAC \mid E \\ D &\rightarrow bD \mid bC \mid b \\ E &\rightarrow aB \mid bC \\ F &\rightarrow aF \mid aG \mid a \\ G &\rightarrow a \mid b \end{aligned}$$

(i) Give all six words of $\mathcal{L}(G_8)$ that have a length not exceeding five.

Solution: These are the productions for words having a length not exceeding five.

$$\begin{array}{l} S \to aA \to aaD \to aab \\ S \to aA \to aaA \to aaaD \to aaab \\ S \to aA \to aaD \to aabD \to aabb \\ S \to aA \to aaA \to aaaA \to aaaaD \to aaaab \\ S \to aA \to aaA \to aaaD \to aaabD \to aaabb \\ S \to aA \to aaD \to aabD \to aabbD \to aabbb \end{array}$$

(ii) List the potentially useful non-terminals of G_8.
Solution: In the first round, D, F, and G are potentially useful. In the next round A (because of aD) is added. In the next round S (because of aA) is added. And in the next round nothing is added, so the potentially useful non-terminals are S, A, D, F, and G.

(iii) Which of the potentially useful non-terminals are not reachable from S?
Solution: If we restrict the grammar G_8 to the potentially useful non-terminals we get: $G_8'' = \langle \{a, b\}, \{S, A, D, F, G\}, R'' \rangle$ where R'' is given by:

$$\begin{array}{rcl} S & \to & aA \\ A & \to & aA \mid aD \\ D & \to & bD \mid b \\ F & \to & aF \mid aG \mid a \\ G & \to & a \mid b \end{array}$$

It is easy to see that from S we can only get to A via aA. And from A we can only get to A via aA and D via aD. And from D we can only get to D via bD. So F and G are not reachable.

(iv) Provide an equivalent grammar G_8' that has no useless symbols. Make sure to write the full triple!
Solution: The potentially useful non-terminals and the non-reachable non-terminals, lead us to grammar $G_8' = \langle \{a, b\}, \{S, A, D\}, R' \rangle$ where R' is given below, is an equivalent grammar not having useless symbols.

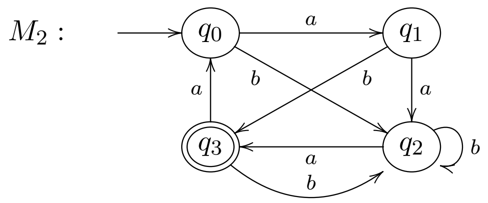
$$\begin{array}{rcl} S & \to & aA \\ A & \to & aA \mid aD \\ D & \to & bD \mid b \end{array}$$

(v) Can you now prove that $\mathcal{L}(G_8)$ is a regular language?
Solution: From the equivalent grammar G_8' it follows that $\mathcal{L}(G_8) = \mathcal{L}(aaa^*b^*b)$, so the language is generated by a regular expression and hence it is regular.

Chapter 5

Automata

Exercise 5.A
Consider the deterministic finite automaton M_2.



Here, $Q = \{q_0, q_1, q_2, q_3\}$, $F = \{q_3\}$, and $\Sigma = \{a, b\}$.

(i) Check whether these words are accepted or not: *abaab*, *aaaba*, *bab*, λ, and *aabbab*.
Solution: The words are processed as follows:

$$\begin{array}{rcl} abaab & : & q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_3 \xrightarrow{a} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_3 \\ aaaba & : & q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \xrightarrow{b} q_2 \xrightarrow{a} q_3 \\ bab & : & q_0 \xrightarrow{b} q_2 \xrightarrow{a} q_3 \xrightarrow{b} q_2 \\ \lambda & : & q_0 \\ aabbab & : & q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{b} q_2 \xrightarrow{b} q_2 \xrightarrow{a} q_3 \xrightarrow{b} q_2 \end{array}$$

Only the words *abaab* and *aaaba* halt in the final state q_3, and are accepted.

(ii) Are these statements true? Give a proof or counterexample.

(1) If w is accepted, then so is $wabba$.
Solution: True. Suppose w has been accepted. Then the automaton is in state q_3 after having read w. Subsequently consuming *abba* results in the computation: $q_3 \xrightarrow{a} q_0 \xrightarrow{b} q_2 \xrightarrow{b} q_2 \xrightarrow{a} q_3$, after which the automaton is yet again in state q_3, and thus it accepts $wabba$.

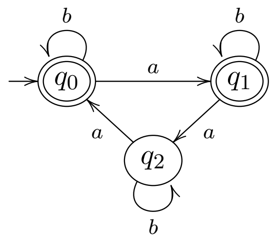
(2) If w is accepted, then wab is not accepted.
Solution: True. The proof is analogous to the previous item.

(3) If w is not accepted, then waa will not be accepted either.
Solution: Not true. Take for example $w = a$. Then w is not accepted, but waa is.

(4) If w is not accepted, then neither is wbb.
Solution: True. If the automaton halts after having consumed w and determines it is not accepted, it has halted in either q_0, q_1, or q_2. From each of these states, consuming two b's results in a non-final state yet again, and thus indeed wbb is not accepted either.

Exercise 5.B
Give all words of length three that are accepted by the following deterministic finite automaton

M_3:



Explain why these words are accepted.

Solution: We systematically list all words of length three and indicate in which state they end. Exactly those words that end in q_0 or q_1 are accepted.

aaa: q_0 accepted
aab: q_2 not accepted
aba: q_2 not accepted
abb: q_1 accepted
baa: q_2 not accepted
bab: q_1 accepted
bba: q_1 accepted
bbb: q_0 accepted

The pattern is that the words ending in the non-final state q_2 are exactly those words that contain precisely two a's. Using the modulo operator it can be expressed as: word w ends in state q_i if and only if the number of a's in w modulo 3 equals i.

Exercise 5.C

Conclude from the grammar G_{10} that

(i) $(aba)^k\, ab \in \mathcal{L}(G_{10})$ for all $k \geq 0$,
Solution: $S \to abaS \to \ldots \to (aba)^k\, S \to (aba)^k\, ab$ for all $k \geq 0$. That is, first traverse the rule $S \to abaS$, k times, and then finally traverse the rule $S \to ab$, once.

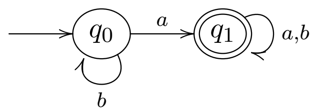
(ii) $aab^k a \in \mathcal{L}(G_{10})$ for all $k \geq 0$,
Solution: $S \to aaB \to aabB \to aabbB \to \ldots \to aab^k B \to aab^k a$ for all $k \geq 0$. That is, first traverse the rule $S \to aaB$, and then k times the rule $B \to bB$, and finally $B \to a$.

(iii) if $w \in \mathcal{L}(G_{10})$, then also $abaw \in \mathcal{L}(G_{10})$,
Solution: $S \to abaS$ and because w is accepted we know that there is an accepting computation $S \to \ldots \to w$. Prepending $S \to abaS$ to this computation gives us the accepting computation $S \to abaS \to \ldots \to abaw$.

(iv) if $w \in \mathcal{L}(G_{10})$, then also $aaaaw \in \mathcal{L}(G_{10})$,
Solution: Similar: we know that an accepting computation $S \to \ldots \to w$ exists, and prepending $S \to aaB \to aaaaS$, we then get the accepting computation $S \to aaB \to aaaaS \to \ldots \to aaaaw$.

Exercise 5.D

Consider the deterministic finite automaton M_4:



Which of the following regular expressions does *not* describe the language of this automaton? Choose one of the options and provide an explanation.

(i) $b^* a\,(a \cup b)^*$
(ii) $(a \cup b)^*\, a\,(a \cup b)^*$
(iii) $(a^* b^*)^*\, ab^*$
(iv) All of the above describe the language of the automaton.

Solution: All regular expressions describe the language of the automaton. The language accepted by this automaton M_4 is the language of all words that contain at least one a. Before we look at the specific regular expressions, note that $(a \cup b)^*$ and $(a^*b^*)^*$ are two ways of expressing any sequence of zero or more a's and b's.

Now let us look at the regular expressions:

(i) $b^*a\,(a \cup b)^*$
This expression states that any word starts with a possibly empty series of b's, then the obligatory a, followed by any sequence of zero or more a's and b's.

(ii) $(a \cup b)^*\,a\,(a \cup b)^*$
This expression states that there is an arbitrary a somewhere in the word. And there is a series of zero or more a's and b's before this obligatory a, as well as after it.

(iii) $(a^*b^*)^*\,ab^*$
This expression states that any word ends with a possibly empty series of b's, and just in front of that the obligatory a. And there is a sequence of zero or more a's and b's in front of this obligatory a.

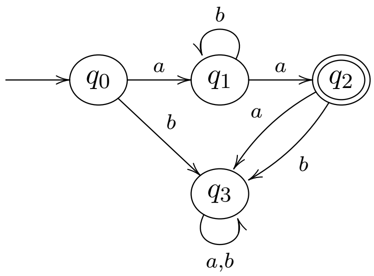
So all expressions indeed describe $\mathcal{L}(M_4)$.

Exercise 5.E
Give a deterministic finite automaton M_5 such that

$$\mathcal{L}(M_5) = \mathcal{L}(ab^*a)$$

Write the automaton by giving the tuple $M_5 = \langle Q, \Sigma, q_0, F, \delta \rangle$.

Solution: This DFA accepts the language $\mathcal{L}(ab^*a)$:



As a tuple it is given by

$$\begin{aligned} Q &:= \{q_0, q_1, q_2, q_3\} \\ \Sigma &:= \{a, b\} \\ q_0 &:= q_0 \\ F &:= \{q_2\} \end{aligned}$$

And the transition function δ is given by

$$\begin{array}{lllllll} \delta(q_0, a) & = & q_1 & \quad & \delta(q_0, b) & = & q_3 \\ \delta(q_1, a) & = & q_2 & & \delta(q_1, b) & = & q_1 \\ \delta(q_2, a) & = & q_3 & & \delta(q_2, b) & = & q_3 \\ \delta(q_3, a) & = & q_3 & & \delta(q_3, b) & = & q_3 \end{array}$$

Exercise 5.F
Construct a right linear grammar for the deterministic finite automaton M_1, similarly as one was constructed for M_2 above. After that, optimize the grammar by removing useless symbols as explained in Section 4.5.

Solution: Introduce S for q_0, A for q_1, B for q_2, and C for q_3.

$$\begin{aligned} S &\to aA \mid bC \\ A &\to bA \mid aB \\ B &\to aC \mid bB \mid \lambda \\ C &\to aC \mid bC \end{aligned}$$

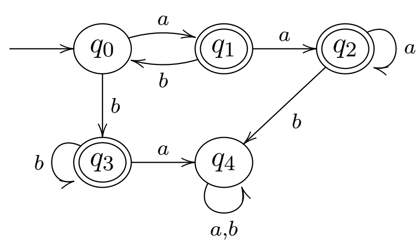
Because A, B, and C recursively rewrite to themselves, you cannot eliminate them by substitution. However, because every production that goes through C never leads to an accepting word (or, in the grammar, the production never stops), we can remove this nonterminal altogether without changing the grammar's language:

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

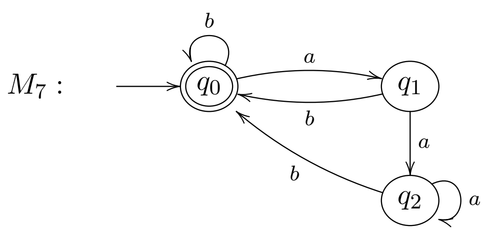
Exercise 5.G
Construct a deterministic finite automaton that recognizes the language of the following grammar G_{14}:

$$\begin{aligned} S &\rightarrow abS \mid aA \mid bB \\ A &\rightarrow aA \mid \lambda \\ B &\rightarrow bB \mid \lambda \end{aligned}$$

Solution: As the default procedure will lead to a state with two a-transitions, we forget about the specific grammar, but focus on the words that are generated by it. It turns out that the grammar produces the following words: a, aa, aaa, b, bb, bbb, aba, $abaa$, $abaaa$, abb, $abbb$, $abbb$, $ababa$, $ababaa$, $ababaaa$, $ababb$, $ababbb$, $ababbbb$,.... So it appears that the language we are dealing with is $\mathcal{L}\left((ab)^* \left(aa^* \cup bb^*\right)\right)$, and a corresponding deterministic finite automaton that accepts this language can be created from scratch:



Exercise 5.H
Consider the deterministic finite automaton M_7:



Construct a right linear grammar that generates $\mathcal{L}(M_7)$.

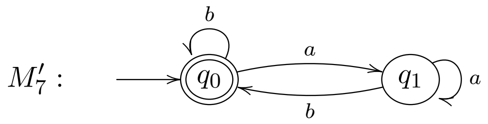
Solution: Again, we first take a look at the words accepted by this automaton: λ, b, bb, bbb, ab, $abab$, bab, $bbab$, aab, $aaab$, $aaaab$, ... This seems to be the language of all words that simply don't end with an a. However, the general procedure works. Introduce S for q_0, A for q_1, and B for q_2, and construct the grammar:

$$\begin{aligned} S &\rightarrow bS \mid aA \mid \lambda \\ A &\rightarrow bS \mid aB \\ B &\rightarrow bS \mid aB \end{aligned}$$

Note how the last two rules are more or less similar. If you think about this a bit, it becomes clear that you can reduce the grammar to:

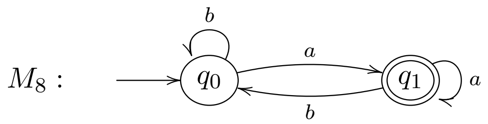
$$\begin{array}{rcl} S & \rightarrow & bS \mid aA \mid \lambda \\ A & \rightarrow & bS \mid aA \end{array}$$

This could have been seen from the automaton as well: the outgoing arrows of q_1 and q_2 are similar, and neither of the two states is a final state, so their behavior is the same, so they might as well be taken together, reducing the automaton to:



...recognizing the same language as M_7.

Exercise 5.I
Consider the deterministic finite automaton M_8:



(i) Construct a right linear grammar that generates $\mathcal{L}(M_8)$.
Solution: We apply the default procedure and get:

$$\begin{array}{rcl} S & \rightarrow & bS \mid aA \\ A & \rightarrow & aA \mid bS \mid \lambda \end{array}$$

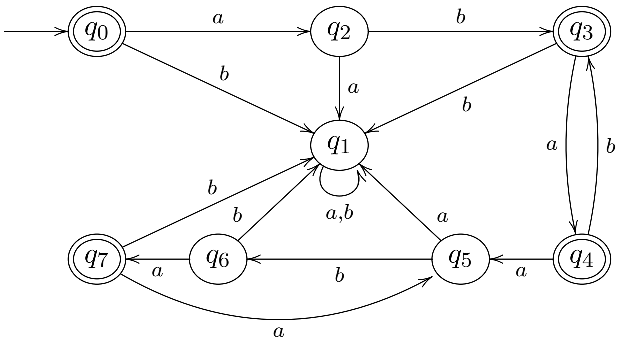
(ii) Provide a description of $\mathcal{L}(M_8)$. Try to make it as simple as possible.
Solution: Closely following the structure of the automaton, you can 'easily' come to the description: $\{b^k aa^l (bb^m aa^n)^p \mid k, l, m, n, p \geq 0\}$, which describes the same language as $\mathcal{L}(b^* aa^* (bb^* aa^*)^*)$. However, thinking a bit further, you will find that the language $\mathcal{L}(M_8)$ is actually fully characterized by the fact that the last symbol must be an a, meaning that we can also simply conclude: $\mathcal{L}(M_8) = \mathcal{L}((a \cup b)^* a)$. Properly stating in natural language that $\mathcal{L}(M_8)$ is containing exactly all words that end on an a, is also accepted as a simple description.

(iii) If all states are made into final states, which language does M_8 recognize?
Solution: For any deterministic finite automaton M where the set of final states F is equal to the set of states Q, we get that $\mathcal{L}(M) = \{a, b\}^*$, or in natural language, all words over $\{a, b\}$. So this also holds for $\mathcal{L}(M_8)$.

(iv) If we swap the final states with non-final states (every final state becomes a non-final state and the other way around), which language would M_8 recognize?
Solution: The *complement* of $\mathcal{L}(M_8)$, or in other words, all words over $\{a, b\}$ that are *not* in $\mathcal{L}(M_8)$. Closely following the structure of the automaton, this would result in the description: $\{b^k \left(aa^l bb^m\right)^p \mid k, l, m, p \geq 0\}$. In terms of a regular expression, that would then be: $\mathcal{L}(b^* (aa^* bb^*)^*)$. However, recalling the simpler characterization that $\mathcal{L}(M_8) = \mathcal{L}((a \cup b)^* a)$, the complement is simply all words not ending on an a, so all words ending on b, or λ. Hence, a simpler description of the complement of $\mathcal{L}(M_8)$ is $\mathcal{L}((a \cup b)^* b \cup \lambda)$.

Exercise 5.J
Define $L_{18} := \{(ab)^k (aba)^l \mid k, l \geq 0\}$ over the alphabet $\Sigma = \{a, b\}$. Construct a deterministic finite automaton that recognizes this language.

Solution: This exercise is often regarded as quite hard!



How do we find such an automaton? A combination of clever thinking and trying to minimize the amount of states necessary. A reconstruction of the thinking that went into making this automaton:

1. $\lambda \in L_{18}$, so the initial state q_0 must also be a final state.

2. $q_0 \overset{b}{\to} \ldots$? Note that the words of L_{18} never begin with a b, so that we can conclude that a sink q_1 and transition $q_0 \overset{b}{\to} q_1$ must be added.

3. $q_0 \overset{a}{\to} \ldots$? Because $a \notin L_{18}$, we need to add a non-final state q_2 with $q_0 \overset{a}{\to} q_2$.

4. $q_1 \overset{a}{\to} q_1$ and $q_1 \overset{b}{\to} q_1$ because q_1 is a sink.

5. $q_2 \overset{a}{\to} \ldots$? Because words never begin with aa, we need to send all computations that read a second a to the sink: $q_2 \overset{a}{\to} q_1$.

6. $q_2 \overset{b}{\to} \ldots$? Because $ab \in L_{18}$ we have to send this b to a final state: either q_0 or a new final state.

   - If we send $q_2 \overset{b}{\to} q_0$, then aba is not accepted, so that won't work.

   Thus, we need to add a new final state q_3 with: $q_2 \overset{b}{\to} q_3$.

7. $q_3 \overset{a}{\to} \ldots$? Because we have $aba \in L_{18}$, we need to send a to a final state: either q_0, or q_3, or a new final state.

   - If $q_3 \overset{a}{\to} q_0$, then $abaab$ is accepted, although $abaab \notin L_{18}$, so that doesn't work.
   - If $q_3 \overset{a}{\to} q_3$, then $abaa$ is accepted, although $abaa \notin L_{18}$, so that doesn't work either.

   Thus, a new final state q_4 must be added: $q_3 \overset{a}{\to} q_4$.

8. $q_3 \overset{b}{\to} \ldots$? Because words never begin with abb, we send all computations that read this b to the sink: $q_3 \overset{b}{\to} q_1$.

9. $q_4 \overset{a}{\to} \ldots$? Because we have $abaa \notin L_{18}$, this a may not be sent to a final state, and because $abaaba \in L_{18}$, it may not be sent to the sink either. So we send it to either to q_2 or to a new non-final state.

   - If $q_4 \overset{a}{\to} q_2$, then $abaab$ is accepted, although $abaab \notin L_{18}$, so that doesn't work.

   Thus, we add a new non-final state q_5 to the automaton: $q_4 \overset{a}{\to} q_5$.

10. $q_4 \overset{b}{\to} \ldots$? Because $abab \in L_{18}$, we must send b to a final state: either q_0, q_3, q_4, or a new one.

- If $q_4 \xrightarrow{b} q_0$, then *ababaaba* is sent to the sink, although $ababaaba \in L_{18}$, so that doesn't work.
- If $q_4 \xrightarrow{b} q_3$, no problems arise, so let's just try this and see if we can complete the automaton.

So we add $q_4 \xrightarrow{b} q_3$.

11. $q_5 \xrightarrow{a} \dots$? Because words never begin with *abaaa*, we must send this a to the sink: $q_5 \xrightarrow{a} q_1$.

12. $q_5 \xrightarrow{b} \dots$? Because $abaab \notin L_{18}$, we must send b to a non-final state. And because $abaaba \in L_{18}$, we can't send b to the sink. So our options are q_2, q_5, or a new non-final state.

    - If $q_5 \xrightarrow{b} q_2$, then *abaabb* is accepted, although $abaabb \notin L_{18}$, so that doesn't work.
    - If $q_5 \xrightarrow{b} q_5$, then *abaaba* is sent to the sink, although $abaaba \in L_{18}$, so that doesn't work either.

    Thus, we need to add a new non-final state q_6 to the automaton: $q_5 \xrightarrow{b} q_6$.

13. $q_6 \xrightarrow{a} \dots$? Because $abaaba \in L_{18}$, we must send a to a final state: either q_0, q_3, q_4, or a new one.

    - If $q_6 \xrightarrow{a} q_0$, then *abaabaab* is accepted, although $abaabaab \notin L_{18}$, so that won't work.
    - If $q_6 \xrightarrow{a} q_3$, then again *abaabaab* is accepted, although $abaabaab \notin L_{18}$, so that won't work either.
    - If $q_6 \xrightarrow{a} q_4$, then *abaababa* is accepted, although $abaababa \notin L_{18}$, so that won't work either.

    Thus, we add a new final state q_7 with: $q_6 \xrightarrow{a} q_7$.

14. $q_6 \xrightarrow{b} \dots$? Because words never begin with *abaabb*, we send b to the sink: $q_6 \xrightarrow{b} q_1$.

15. $q_7 \xrightarrow{a} \dots$? Because $abaabaa \notin L_{18}$, we must send a to a non-final state: either q_2, q_5, q_6, or a new non-final state.

    - If $q_7 \xrightarrow{a} q_2$, then *abaabaab* is accepted, although $abaabaab \notin L_{18}$, so that doesn't work.
    - If $q_7 \xrightarrow{a} q_5$, no problems arise. Also, we see that we have now added the *aba* loop in the regular expression.

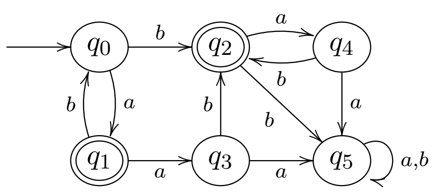
    So we add: $q_7 \xrightarrow{a} q_5$.

16. $q_7 \xrightarrow{b} \dots$? Because words never begin with *abaabab*, we send b to the sink. So: $q_7 \xrightarrow{b} q_1$.

And now, we have ended up with a valid automaton that accepts exactly the right words!

Exercise 5.K
Define $L_{19} := \{(ab)^k \, x \, (ab)^l \mid x \in \{a, b\}, k, l \geq 0\}$ over the alphabet $\Sigma = \{a, b\}$. Construct a deterministic finite automaton that recognizes this language.

Solution: This one is probably a bit easier than the previous one, but nevertheless still quite hard. (Check that indeed all words of the form $(ab)^k \, a \, (ab)^l$ and $(ab)^k \, b \, (ab)^l$ are accepted).



How to find such an automaton? This time, we will use a different method than we used in Exercise 5.J. In the reader, it is described how to construct an automaton from a right linear grammar, which is quite straightforward. So what we will do, is first make a right linear grammar that corresponds to the language in question, and then construct our automaton via that method.

$$\begin{aligned} S &\rightarrow abS \mid aA \mid bA \\ A &\rightarrow abA \mid \lambda \end{aligned}$$

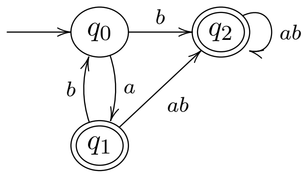
Unfortunately, the procedure outlined in the reader isn't directly of use, because there is overlap in the grammar: the a is the first letter of both the abS as well as the aA in the grammar. So, we change the grammar to get an equivalent grammar without this problem, by adding an extra nonterminal:

$$\begin{aligned} S &\rightarrow aP \mid bA \\ P &\rightarrow bS \mid A \\ A &\rightarrow abA \mid \lambda \end{aligned}$$

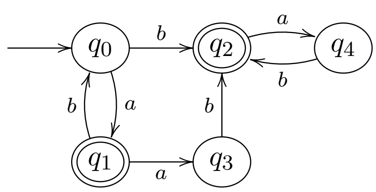
We still have a problem though, namely the newly introduced production rule $P \rightarrow A$. So we edit the grammar to this equivalent version:

$$\begin{aligned} S &\rightarrow aP \mid bA \\ P &\rightarrow bS \mid abA \mid \lambda \\ A &\rightarrow abA \mid \lambda \end{aligned}$$

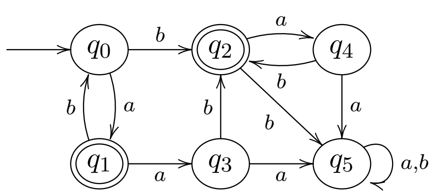
(Check for yourself that this grammar is indeed equivalent! The current grammar works, and so we can apply the method described in the reader, first introducing a state for each nonterminal ($S \rightarrow q_0$, $P \rightarrow q_1$ and $A \rightarrow q_2$) and adding the word transitions:



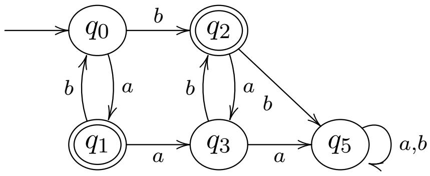
Then, we split up the word transitions into single letter transitions:



And finally, adding a sink and sink transitions, we get the same automaton as shown above:



Note that this automaton can still be optimized a bit, because the states q_3 and q_4 have the same outgoing arrows (an a to the sink and a b to q_2), and thus they have the same behavior and can be joined together without changing the recognized language of the automaton:

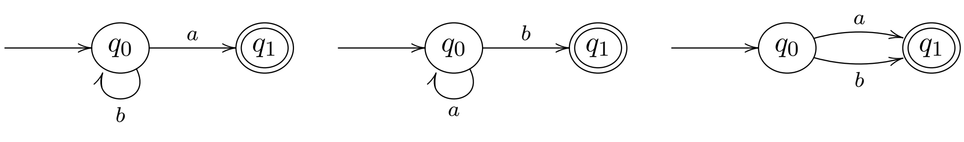


Note also, that if you had constructed the automaton for the language in the same way as in Exercise 5.J, you would have ended up with this last automaton in the first place.

Exercise 5.L
In this exercise we consider the class of deterministic finite automata ${M_9}^i$ over alphabet $\Sigma = \{a, b\}$, where i is a natural number representing an index, such that

- ${M_9}^i$ has at most two states,
- $\lambda \notin \mathcal{L}\left({M_9}^i\right)$, and
- $\mathcal{L}\left({M_9}^i\right) \neq \varnothing$

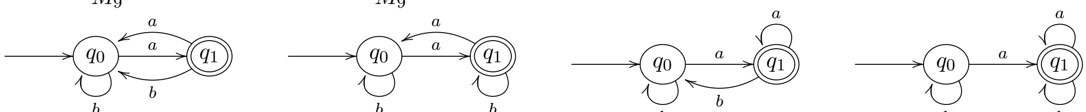
(i) How many different deterministic finite automata exist with these properties?
Solution: There are twelve DFAs that comply with these requirements.
The first claim is that from the last two requirements it follows that the automaton should have *exactly* two states. Because if it has only one state which is not a final state, then $\mathcal{L}\left({M_9}^i\right) = \varnothing$, which is not allowed. And if the only state is a final state, then $\lambda \in \mathcal{L}\left({M_9}^i\right)$ which is also not allowed.
Another claim is that the first state is not a final state, that there is a transition from the first to the second state and the second state is a final state. We have already seen that the first state cannot be a final state. And if no states are final, then $\mathcal{L}\left({M_9}^i\right) = \varnothing$ which is not allowed. So the second state must be a final state. And if there is no transition between this non-final first state and the final second state, then again $\mathcal{L}\left({M_9}^i\right) = \varnothing$ which is not allowed.
For this transition there are three different options. Either there is a single transition labeled with a (and because it is a DFA this implies that there must be a loop on the initial state with b), or there is a single transition labeled with b (and hence there must be a loop on the initial state with a, or there are two transitions labeled with a and b.
So based on the requirements, we know that there are three different automata when we leave out the transitions from the second state:



And for the transitions from the second state we have two options for a, namely going back to the first state or stay in the second state. Likewise, we have two options for b. So in total we have four options for the transitions from the second state.
In total this means that we have three times four different automata. So there are twelve of them.

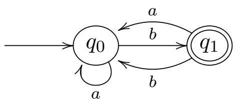
(ii) Draw all these automata.
Solution:

${M_9}^1$ ${M_9}^2$ ${M_9}^3$ ${M_9}^4$



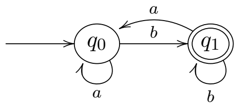
$b^*a\,((a \cup b)\,b^*a)^*$ $b^*a\,(b \cup ab^*a)^*$ $b^*a\,(a \cup bb^*a)^*$ $b^*a\,(a \cup b)^*$

$M_9{}^5$



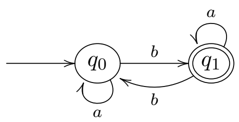
$a^*b\left((a\cup b)\,a^*b\right)^*$

$M_9{}^6$



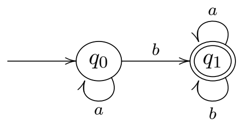
$a^*b\left(b\cup aa^*b\right)^*$

$M_9{}^7$



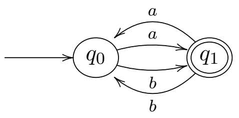
$a^*b\left(a\cup ba^*b\right)^*$

$M_9{}^8$



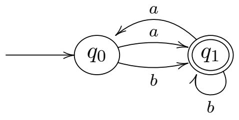
$a^*b\left(a\cup b\right)^*$

$M_9{}^9$



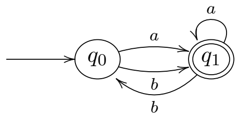
$(a\cup b)\left((a\cup b)\,(a\cup b)\right)^*$

$M_9{}^{10}$



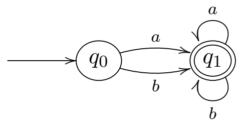
$(a\cup b)\left(b\cup a\,(a\cup b)\right)^*$

$M_9{}^{11}$



$(a\cup b)\left(a\cup b\,(a\cup b)\right)^*$

$M_9{}^{12}$



$(a\cup b)\,(a\cup b)^*$

(iii) For each of these automata $M_9{}^i$, provide a regular expression that generates the same language $\mathcal{L}\left(M_9{}^i\right)$.
Solution: See above.

(iv) How many different languages do these automata generate?
Solution: The claim is that each of the automata generates a different language, so based on the given requirements, twelve different languages can be generated.

- $\mathcal{L}\left(M_9{}^1\right)$ is different from
  - $\mathcal{L}\left(M_9{}^2\right)$ and $\mathcal{L}\left(M_9{}^4\right)$ as these include ab and $\mathcal{L}\left(M_9{}^1\right)$ does not.
  - $\mathcal{L}\left(M_9{}^3\right)$ as the latter includes aa and $\mathcal{L}\left(M_9{}^1\right)$ does not.
  - All 'higher' languages as these include b and $\mathcal{L}\left(M_9{}^1\right)$ does not. (A language $\mathcal{L}\left(M_9{}^j\right)$ is considered a 'higher' language than $\mathcal{L}\left(M_9{}^i\right)$ if $j > i$.)
- $\mathcal{L}\left(M_9{}^2\right)$ is different from
  - $\mathcal{L}\left(M_9{}^3\right)$ and $\mathcal{L}\left(M_9{}^4\right)$ as these include aa and $\mathcal{L}\left(M_9{}^2\right)$ does not.
  - All 'higher' languages as these include b and $\mathcal{L}\left(M_9{}^2\right)$ does not.
- $\mathcal{L}\left(M_9{}^3\right)$ is different from
  - $\mathcal{L}\left(M_9{}^4\right)$ as the latter includes ab and $\mathcal{L}\left(M_9{}^3\right)$ does not.
  - All 'higher' languages as these include b and $\mathcal{L}\left(M_9{}^3\right)$ does not.
- $\mathcal{L}\left(M_9{}^4\right)$ is different from
  - All 'higher' languages as these include b and $\mathcal{L}\left(M_9{}^4\right)$ does not.
- $\mathcal{L}\left(M_9{}^5\right)$ is different from
  - $\mathcal{L}\left(M_9{}^6\right)$ as the latter includes bb and $\mathcal{L}\left(M_9{}^5\right)$ does not.
  - $\mathcal{L}\left(M_9{}^7\right)$ and $\mathcal{L}\left(M_9{}^8\right)$ as these include ba and $\mathcal{L}\left(M_9{}^5\right)$ does not.
  - All 'higher' languages as these include a and $\mathcal{L}\left(M_9{}^5\right)$ does not.
- $\mathcal{L}\left(M_9{}^6\right)$ is different from
  - $\mathcal{L}\left(M_9{}^7\right)$ and $\mathcal{L}\left(M_9{}^8\right)$ as these include ba and $\mathcal{L}\left(M_9{}^6\right)$ does not.
  - All 'higher' languages as these include a and $\mathcal{L}\left(M_9{}^6\right)$ does not.
- $\mathcal{L}\left(M_9{}^7\right)$ is different from
  - $\mathcal{L}\left(M_9{}^8\right)$ as the latter includes bb and $\mathcal{L}\left(M_9{}^7\right)$ does not.
  - All 'higher' languages as these include a and $\mathcal{L}\left(M_9{}^7\right)$ does not.
- $\mathcal{L}\left(M_9{}^8\right)$ is different from
  - All 'higher' languages as these include a and $\mathcal{L}\left(M_9{}^8\right)$ does not.
- $\mathcal{L}\left(M_9{}^9\right)$ is different from
  - $\mathcal{L}\left(M_9{}^{10}\right)$ as the latter includes ab and $\mathcal{L}\left(M_9{}^9\right)$ does not.
  - All 'higher' languages as these include aa and $\mathcal{L}\left(M_9{}^9\right)$ does not.

- $\mathcal{L}\left(M_9{}^{10}\right)$ is different from
  - All 'higher' languages as these include aa and $\mathcal{L}\left(M_9{}^{10}\right)$ does not.
- $\mathcal{L}\left(M_9{}^{11}\right)$ is different from
  - $\mathcal{L}\left(M_9{}^{12}\right)$ as the latter includes ab and $\mathcal{L}\left(M_9{}^{11}\right)$ does not.

Note that in the list above we have shown that for all pairs (i, j) where $i < j$ it holds that $\mathcal{L}\left(M_9{}^i\right) \neq \mathcal{L}\left(M_9{}^j\right)$. Due to the symmetry of 'not being equal' that suffices to prove that indeed all languages are different.

Exercise 5.M

(i) Investigate in the non-deterministic finite automaton M_{10} that we defined before which computations exist with input *abaaa*, *ababa*, *ab* and *baaab*.

Solution:

$$\begin{array}{lcl} abaaa & : & q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \\ abaaa & : & q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_1 \\ abaaa & : & q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \\ abaaa & : & q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \\ abaaa & : & q_0 \xrightarrow{a} q_1 \end{array}$$

So *abaaa* can end in q_0, q_1, q_2 or q_3 or it gets stuck in a deadlock in q_1.

$$\begin{array}{lcl} ababa & : & q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \\ ababa & : & q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \\ ababa & : & q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \\ ababa & : & q_0 \xrightarrow{a} q_1 \end{array}$$

So *ababa* can end in q_0 or q_1 or it gets stuck in a deadlock in q_1.

$$\begin{array}{lcl} ab & : & q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \\ ab & : & q_0 \xrightarrow{a} q_1 \end{array}$$

So *ab* can end in q_0 or it gets stuck in a deadlock in q_1.

$$\begin{array}{lcl} baaab & : & q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \\ baaab & : & q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_1 \\ baaab & : & q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \\ baaab & : & q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \xrightarrow{b} q_3 \end{array}$$

So *baaab* can end in q_0 or q_3 or it gets stuck in a deadlock in q_1 or q_2.

(ii) Which of these words are accepted?

Solution: Only *abaaa* and *baaab* are accepted.

(iii) Describe the language that M_{10} accepts.

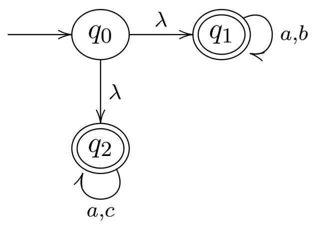
Solution: $\{w \in \{a, b\}^* \mid w \text{ contains } aaa\}$.

(iv) Adapt M_{10} in such a way that it accepts $\{w \mid w \text{ ends with } aaa\}$.

Solution: Remove the a, b loop at q_3. Actually, removing only the b loop already suffices to achieve this.

Exercise 5.N
Consider the non-deterministic finite automaton M_{11}.

M_{11} :



(i) Which computations are possible with input aba, cac, abc and λ?
Solution:

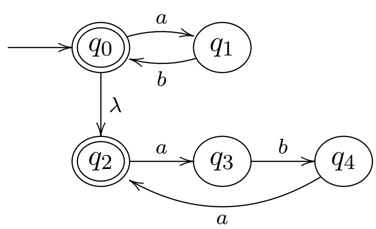
$$\begin{array}{rcl}
aba & : & q_0 \overset{\lambda}{\to} q_1 \overset{a}{\to} q_1 \overset{b}{\to} q_1 \overset{a}{\to} q_1 \\
aba & : & q_0 \overset{\lambda}{\to} q_2 \overset{a}{\to} q_2 \\
cac & : & q_0 \overset{\lambda}{\to} q_1 \\
cac & : & q_0 \overset{\lambda}{\to} q_2 \overset{c}{\to} q_2 \overset{a}{\to} q_2 \overset{c}{\to} q_2 \\
abc & : & q_0 \overset{\lambda}{\to} q_1 \overset{a}{\to} q_1 \overset{b}{\to} q_1 \\
abc & : & q_0 \overset{\lambda}{\to} q_2 \overset{a}{\to} q_2 \\
\lambda & : & q_0 \\
\lambda & : & q_0 \overset{\lambda}{\to} q_1 \\
\lambda & : & q_0 \overset{\lambda}{\to} q_2
\end{array}$$

(ii) Which of these words are accepted?
Solution: Only aba, cac and λ.

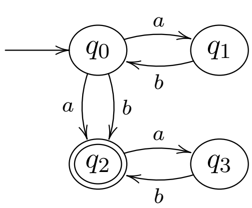
(iii) Describe the language that M_{11} accepts.
Solution: $\mathcal{L}(M_{11}) = \{w \in \{a,b,c\}^* \mid w \text{ contains no } c \text{ or } w \text{ contains no } b\}$. It is a regular language as $\mathcal{L}(M_{11}) == \mathcal{L}\left((a \cup b)^* \cup (a \cup c)^*\right)$.

Exercise 5.O

(i) Construct a non-deterministic automaton with at most five states that accepts the language L_{18} from Exercise 5.J.
Solution: Take for instance:



(ii) Construct a non-deterministic automaton with at most four states that accepts the language L_{19} from Exercise 5.K.
Solution: Take for instance:

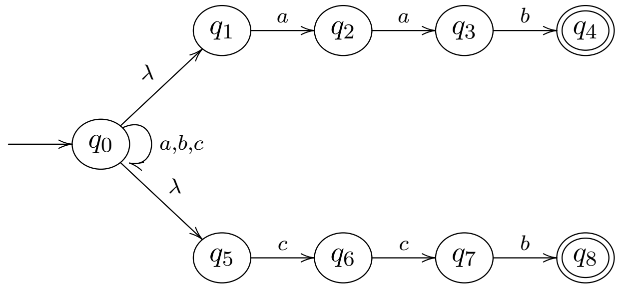


Exercise 5.P

(i) Construct a non-deterministic finite automaton for the language

$$L = \{w \in \{a, b, c\}^* \mid w \text{ ends with } aab \text{ or } w \text{ ends with } ccb\}$$

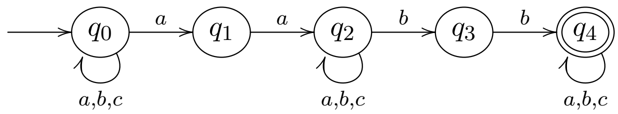
Solution:



(ii) Construct a non-deterministic finite automaton for the language

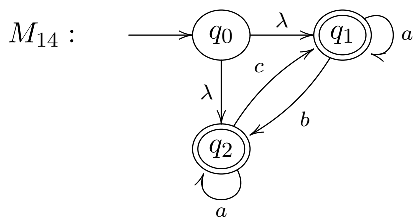
$$L' = \{w \in \{a, b, c\}^* \mid w = vu \text{ and } v \text{ contains } aa \text{ and } u \text{ contains } bb\}$$

Solution:



Exercise 5.Q

(i) Describe the language that automaton M_{14} accepts.



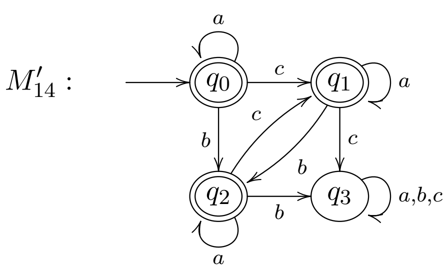
Solution: The words in the language consist of arbitrarily long sequences of alternating b's and c's, where between these arbitrarily long sequences of a's are allowed.
In other words:

$$\begin{aligned}
& \{a^{k_0}ba^{k_1}ca^{k_2}b \dots ba^{k_n}ca^{k_{n+1}} \mid k_0, k_1, k_2 \dots, k_{n+1} \geq 0\} \\
\cup \ & \{a^{k_0}ba^{k_1}ca^{k_2}b \dots ca^{k_n}ba^{k_{n+1}} \mid k_0, k_1, k_2 \dots, k_{n+1} \geq 0\} \\
\cup \ & \{a^{k_0}ca^{k_1}ba^{k_2}c \dots ba^{k_n}ca^{k_{n+1}} \mid k_0, k_1, k_2 \dots, k_{n+1} \geq 0\} \\
\cup \ & \{a^{k_0}ca^{k_1}ba^{k_2}c \dots ca^{k_n}ba^{k_{n+1}} \mid k_0, k_1, k_2 \dots, k_{n+1} \geq 0\}
\end{aligned}$$

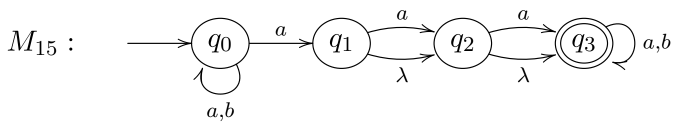
The language can also be described using a regular expression:

$$\begin{aligned}
& \mathcal{L}(M_{14}) \\
= \ & \mathcal{L}(a^* (((b \cup \lambda)(a^*ca^*b)^*) \cup ((c \cup \lambda)(a^*ba^*c)^*)) a^*) \\
= \ & \mathcal{L}(a^* \cup a^*b(a^*ca^*ba^*)^* \cup a^*b(a^*ca^*ba^*)^* ca^* \cup a^*c(a^*ba^*ca^*)^* \cup a^*c(a^*ba^*ca^*)^* ba^*)
\end{aligned}$$

(ii) Construct a deterministic finite automaton that accepts the language of M_{14}.
Solution:



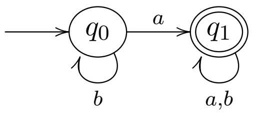
(iii) Describe the language that M_{15} accepts.



Solution: $\mathcal{L}(M_{15}) = \{w \in \{a,b\}^* \mid w \text{ contains an } a\} = \mathcal{L}(b^*a(a \cup b)^*)$.

(iv) Construct a deterministic finite automaton that accepts the language of M_{15}.

Solution:



Exercise 5.R

(i) Suppose M_1 is a finite automaton that accepts L_1 and that M_2 is a finite automaton that accepts L_2. Construct a non-deterministic finite automaton that accepts $L_1 \cup L_2$. [*Hint:* Look at example M_{11} above.]

Solution: Suppose: M_1 accepts L_1 and M_2 accepts L_2. Now define a new non-deterministic automaton M in the following way (by combining M_1 and M_2): add a new initial state q_0 and from there a λ arrow to the initial state of M_1 and a λ arrow to the initial state of M_2. So the initial states of M_1 and M_2 are no longer initial states in M. This non-deterministic automaton M accepts precisely $L_1 \cup L_2$, because:

- If $w \in L_1 \cup L_2$, then $w \in L_1$ or $w \in L_2$. In the first case w is accepted: take a λ step to the former initial state of M_1 and go through M_1 until you are in a final state. This is possible because $w \in L_1$. The second case is similar.
- If w is accepted by M, then there is a computation that ends in a final state. In that case there are two possibilities: either the first step is a λ step 'to' M_1, and then we have $w \in L_1$, or the first step is a λ step 'to' M_2, and then $w \in L_2$. In both cases: $w \in L_1 \cup L_2$.

(ii) Prove that the class of regular languages is closed under $\cup$, i.e., if L_1 and L_2 are regular, then $L_1 \cup L_2$ is also regular.

Solution: We give a proof based upon the theory of the last two chapters:

- Assume that L_1 and L_2 are regular.
- Then there are right linear grammars G_1 and G_2 that generate L_1, resp. L_2 according to Theorem 4.33.
- Therefore there are also finite automata M_1 and M_2 that accept L_1, resp. L_2 according to Theorem 5.9.
- And based upon Exercise (i) we know that there is also a non-deterministic finite automaton that accepts $L_1 \cup L_2$.
- But then there is also a deterministic finite automaton that accepts $L_1 \cup L_2$ according to Theorem 5.15.
- And therefore there is also a right linear grammar that generates $L_1 \cup L_2$ according to Theorem 5.6.
- But then $L_1 \cup L_2$ is regular according to Theorem 4.33.

Of course you can also prove this by just looking at the definition of regular expressions and not use any automata theory, but that answer fits less well with this part about automata.

(iii) Suppose M_1 is a finite automaton that accepts L_1 and M_2 is a finite automaton that accepts L_2. Define a non-deterministic finite automaton that accepts L_1L_2. Remember that L_1L_2 is the language that consists of first a word from L_1 and then a word from L_2, so $L_1L_2 = \{vw \mid v \in L_1, w \in L_2\}$.

Solution: Again we prove this using our knowledge about languages and automata:

- Assume that M_1 accepts L_1 and M_2 accepts L_2.
- Now make a new non-deterministic finite automaton M as follows (from M_1 and M_2):
  - take as initial state q_0 the initial state of M_1;
  - add from all final states of M_1 a λ arrow to the initial state of M_2;
  - so the final states of M_1 are no longer final states in M, and the initial state of M_2 is no longer an initial state.

This non-deterministic automaton M accepts exactly L_1L_2, because:

- If $w \in L_1L_2$, then $w = uv$ for certain $u \in L_1$ and $v \in L_2$. Execute now M_1 with u until you are in a final state of M_1. (This is possible because $u \in L_1$.) Subsequently take a λ step to the initial state of M_2 and execute now M_2 with v until you are in a final state. (This is possible because $v \in L_2$).
- If w is accepted by M, then there is a computation that ends in a final state. This is only reachable by first running M_1 until you are in an (old) final state of M_1 and then run M_2 until you are in a final state of M_2. In the first part a word $u \in L_1$ is consumed and afterwards in the second part a word $v \in L_2$. Therefore $w \in L_1L_2$.

(iv) Prove that the class of regular languages is closed under *concatenation*, i.e., if L_1 and L_2 are regular, then L_1L_2 is also regular.

Solution: The reasoning here is similar to the one of Exercise (ii):

- Assume that L_1 and L_2 are regular.
- Then there are right linear grammars G_1 and G_2 that generate L_1, resp. L_2 according to Theorem 4.33.
- Therefore there are also finite automata M_1 and M_2 that accept L_1, resp. L_2 according to Theorem 5.9.
- And based upon Exercise (ii) we know that there is also a non-deterministic finite automaton that accepts L_1L_2.
- But then there is also a deterministic finite automaton that accepts L_1L_2 according to Theorem 5.15.
- And therefore there is also a right linear grammar that generates L_1L_2 according to Theorem 5.6.
- But then L_1L_2 is regular according to Theorem 4.33.

Chapter 6

Modal logic

Exercise 6.A
So we see that the formula $\Diamond R$ means the same as $\neg\Box\neg R$. Try to find a formula without the symbol '$\Box$' that means the same as $\Box R$. Then, translate both $\Box R$ and the formula you found to ordinary English.

Solution: We have $\Box R \equiv \neg\Diamond\neg R$. The formula $\Box R$ means 'It is necessarily true that it is raining.' And the formula $\neg\Diamond\neg R$ means 'It is not possible for it not to be raining.'

Exercise 6.B
In modal logic the statement 'U is necessarily true' is symbolically represented as $\Box U$, and 'U is impossible' as $\Box\neg U$, or alternatively, $\neg\Diamond U$. Give two different formulas in modal logic which express the statement that 'U is contingent.'

Solution: $\neg\Box U \wedge \Diamond U$ and $\Diamond U \wedge \Diamond\neg U$.

Exercise 6.C
Use the following dictionary

M	I have money
B	I am buying something

to translate these English sentences to formulas of modal logic:

(i) *It is possible for me to buy something without having money.*
Solution: $\Diamond(B \wedge \neg M)$. The sentence seems impossible. (Unless of course we consider using a credit card or other such methods of payment.) An alternative translation would be $(\Diamond B) \wedge \neg M$, which is also impossible.

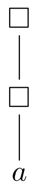
(ii) *It is necessarily true, if I am buying something, for me to have money.*
Solution: $\Box(B \to M)$. This sentence indeed seems to be necessarily true.

(iii) *It is possible that if I buy something I don't have any money.*
Solution: This can depend on your interpretation of the sentence. Writing $\Diamond(B \to \neg M)$, we have a necessarily true statement. For instance, $B \to \neg M$ is true when not buying anything at all, regardless of the amount of money in possession. Therefore, $\Diamond(B \to \neg M)$ is necessarily true. However, if you choose for the alternative of $\Diamond(B \wedge \neg M)$, we have already seen its impossibility in the above.

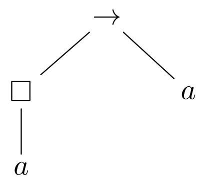
Which of *these* sentences seem true? Explain why.

Exercise 6.D
For each of the formulas below, give the 'official' form according to the grammar of modal logic, and also draw a tree according to its structure.

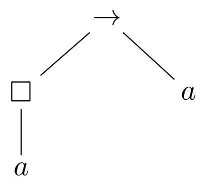
(i) $\Box(\Box a)$
Solution: $\Box\Box a$;



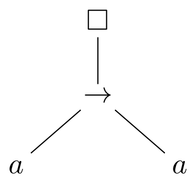
(ii) $\Box a \to a$
Solution: $(\Box a \to a)$;



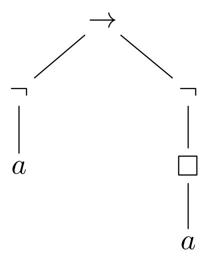
(iii) $(\Box a) \to a$
Solution: $(\Box a \to a)$;



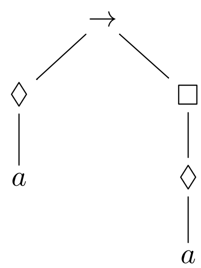
(iv) $\Box(a \to a)$
Solution: $\Box(a \to a)$;



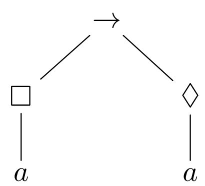
(v) $\neg a \to \neg\Box a$
Solution: $(\neg a \to \neg\Box a)$;



(vi) $\Diamond a \to \Box\Diamond a$
Solution: $(\Diamond a \to \Box\Diamond a)$;

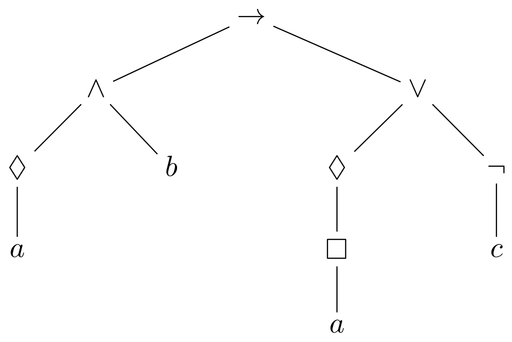


(vii) $\Box a \to \Diamond a$
Solution: $(\Box a \to \Diamond a)$;



(viii) $\Diamond a \wedge b \to \Diamond\Box a \vee \neg c$

Solution: $((\Diamond a \wedge b) \to (\Diamond\Box a \vee \neg c))$;



Exercise 6.E

Use the following dictionary:

R	I am ready
H	I am going home

For each of the logics listed in the table above (except program-logic), give an English translation of the formula:

$$\neg R \to \neg\Diamond H$$

Solution:

(i) Modal: If I am not ready, it is not possible for me to go home.
(ii) Epistemic: If I am not ready, it contradicts my knowledge that I am going home.
(iii) Doxastic: If I am not ready, it contradicts my beliefs that I am going home.
(iv) Temporal: If I am not ready, then it is not the case that I sometimes go home. Or rephrased: If I am not ready, I never go home.
(v) Deontic: If I am not ready, I am not allowed to go home.

Exercise 6.F

Construct a matrix that indicates which axiom schemes holds in which logic. Let the rows denote the logics listed on page 110, and the columns the axiom schemes. The matrix will then have a total of 36 cells. Then write a '+' or '−' in each cell, according to whether you think the axiom scheme is always true in the logic, or you think that it need not always hold.

Solution:

	distr.	refl.	sym.	trans.	Eucl.	ser.
	$\Box(f \to g) \to (\Box f \to \Box g)$	$\Box f \to f$	$f \to \Box\Diamond f$	$\Box f \to \Box\Box f$	$\Diamond f \to \Box\Diamond f$	$\Box f \to \Diamond f$
	K	T	B	4	5	D
Modal	+[1]	+[2]	+[3]	+[4]	+[5]	+[6]
Epistemic	+[7]	+[8]	+[9]	+[10]	−/+[11]	+[12]
Doxastic	+[13]	−[14]	+[15]	+[16]	−/+[17]	−[18]
Temporal	+[19]	+[20]	−[21]	+[22]	−[23]	+[24]
Deontic	+[25]	−[26]	−[27]	−/+[28]	−/+[29]	−/+[30]
Program	+[31]	−[32]	−[33]	−[34]	−[35]	+[36]

Below we explain our solutions to this problem. Note how some combinations of axiom schemes and logics are not entirely clear, and leave room for discussion. This means that our solutions need not be the true ones at all. If you think our reasoning is wrong or incomplete, tell us! Also, (because of this) it is not useful to memorize these answers at all—at the test, if we ask a question like this, you should be able to answer with arguments of your own.

Modal logic

1. If g necessarily follows from f and f is necessarily true, then g is necessarily true as well.
2. If f is necessarily true, then in particular, f is true.
3. If f is true, then it must also necessarily be the case that f is possible.
4. If f is necessarily true, then it only seems reasonable for $\Box f$ to be necessarily true too.
5. If f is possible, then it would be weird if $\Diamond f$ is not necessarily true.
6. Every reflexive system is serial as well.

Epistemic logic

7. If we know that g follows from f and we know f is true, then we can deduce (and thus know) that g is true.
8. Only truths f can be known, right?
9. If f is true, then indeed you know that it is not contradictory with your knowledge.
10. You know what you know.
11. This translates to something like: you know what you don't know. If something could be known, then you know that it could be known. Not everybody thinks this is true.
12. Every reflexive system is serial too.

Doxastic logic

13. If you believe that g follows from f, and you believe that f holds, it is only logical to believe g too.
14. Not everything you believe is true.
15. If something is true, do you then believe it to be not contradictory to your belief?
16. If you believe something, then surely you believe that you believe it?
17. If something is not contradictory to your beliefs, is that then a reason to believe it?
18. People can believe in contradictory things.

Temporal logic

19. If it is the case that f implies g, and also that f always holds, well then g will also always hold.
20. If f always holds, then in particular it must hold at this moment. (Note: we are using the LTL interpretation of Section 6.6; but other temporal logics exist as well.)
21. Given the LTL interpretation that tells us that the current moment in time is also a part of the future, does it follow from the fact that f holds currently, that f will always hold eventually? No, for instance, take a property f that holds only today. Then, today, $\Diamond f$ indeed holds. But tomorrow, not any more. And thus $\Box\Diamond f$ does not hold. (If you think about it, $\Box\Diamond f$ says that f will hold an infinite amount of times: at each point in time, there is a later point in time at which f will hold.)
22. If something holds forever, from this point on, then at any later point in time, it will hold forever too.

23. If something will eventually be true, then it doesn't have to always be eventually true. See also the counterexample against symmetry in 21.

24. If something is always true, then in particular it is eventually true.

Deontic logic

25. If you ought to do g in the case of f, and you ought to do f, then you ought to do g as well.

26. That you ought to do f doesn't necessarily mean you actually will do f.

27. You might do f even though it was not ought to be permissible.

28. If you are ought to do f, are you then ought to ought to do it?

29. If you may do f, is it then ought to be permissible?

30. It doesn't seem unreasonable to assume that if you ought to do something, it should be permissible as well. On the other hand, you might ought to do something because you made up your own mind about it, although someone else does not find this permissible. Also, this axiom is equivalent to $\Box f \wedge \Box \neg f$. This kind of 'double bind' indeed sometimes happens, for instance when reading a sign stating 'dismiss this sign'.

Program logic

31. If after any execution of the program, g follows from f, and f holds, then indeed g holds as well.

32. The fact that f must hold after any execution of a program does not imply that f holds now.

33. If f holds now, it need not be that after any execution f holds after a certain execution.

34. If f holds after any execution, it need to hold after a second execution as well.

35. If f holds after a certain execution, it need not be that after any execution f holds after a certain execution.

36. If f holds after any execution, then of course a certain execution exists after which f holds.

Exercise 6.G

For each formula f listed below, check in which worlds x of our Kripke model in Example 6.6 it holds, that is, for which x we have $x \Vdash f$. In addition, check whether f holds in the model, that is, check whether $M_1 \models f$.

(i) a

Solution: For $x \in \{x_0, x_1, x_3\}$ we have $a \in V(x)$ and thus $x \Vdash a$. But because $x \notin V(x_2)$ we have $x_2 \nVdash a$. And thus $M_1 \not\models a$.

(ii) $\Box a$

Solution: First we check x_0, for which we have $R(x_0) = \{x_1\}$. Because $x_1 \Vdash a$, we indeed have $x_0 \Vdash \Box a$. Then for x_1, for which $R(x_1) = \{x_0, x_1\}$, we verify that $x_0 \Vdash a$ and $x_1 \Vdash a$, and thus $x_1 \Vdash \Box a$. And for x_2, we have $R(x_2) = \{x_0\}$ and $x_0 \Vdash a$, so $x_2 \Vdash \Box a$ as well. Finally, $R(x_3) = \varnothing$ and thus $x_3 \Vdash \Box a$ holds vacuously, because there are no accessible worlds for which we have to check a's truth. This all shows us that indeed $M_1 \models \Box a$.

(iii) $\Diamond a$
Solution: $R(x_0) = \{x_1\}$ and $x_1 \Vdash a$, so $x_0 \Vdash \Diamond a$. $R(x_1) = \{x_0, x_1\}$ and $x_0 \Vdash a$, so $x_1 \Vdash \Diamond a$. $R(x_2) = \{x_0\}$ and $x_0 \Vdash a$, so $x_2 \Vdash \Diamond a$. $R(x_3) = \varnothing$ so vacuously $x_3 \nVdash \Diamond a$, because $\Diamond a$ requires a world successive to x_3. Hence $M_1 \not\models \Diamond a$.

(iv) $\Box a \to a$
Solution: We have already seen that $x_0 \Vdash a$, $x_1 \Vdash a$, and $x_3 \Vdash a$. Simply from the truth table of implication we then have $x_0 \Vdash \Box a \to a$, $x_1 \Vdash \Box a \to a$, and $x_3 \Vdash \Box a \to a$. Furthermore, we know that $x_2 \Vdash \Box a$ but $x_2 \nVdash a$. Thus, $x_2 \nVdash \Box a \to a$. And thus, $M_1 \not\models \Box a \to a$.

(v) $\Box a \to \Box\Box a$
Solution: $R(x_0) = \{x_1\}$ and $x_1 \Vdash \Box a$, so $x_0 \Vdash \Box\Box a$. And thus $x_0 \Vdash \Box a \to \Box\Box a$. $R(x_1) = \{x_0, x_1\}$, $x_0 \Vdash \Box a$ and $x_1 \Vdash \Box a$, so $x_1 \Vdash \Box\Box a$. And thus $x_1 \Vdash \Box a \to \Box\Box a$. $R(x_2) = \{x_0\}$ and $x_0 \Vdash \Box a$, so $x_2 \Vdash \Box\Box a$. And thus $x_2 \Vdash \Box a \to \Box\Box a$. $R(x_3) = \varnothing$ so that vacuously $x_3 \Vdash \Box\Box a$. And thus $x_3 \Vdash \Box a \to \Box\Box a$. And finally, $M_1 \models \Box a \to \Box\Box a$.

(vi) $a \to \Box\Diamond a$
Solution: $R(x_0) = \{x_1\}$. We have already seen that $x_1 \Vdash \Diamond a$. Thus $x_0 \Vdash \Box\Diamond a$. And also $x_0 \Vdash a \to \Box\Diamond a$. $R(x_1) = \{x_0, x_1\}$. We already know that $x_0 \Vdash \Diamond a$ and $x_1 \Vdash \Diamond a$. Thus $x_1 \Vdash \Box\Diamond a$. And also $x_1 \Vdash a \to \Box\Diamond a$. Because $x_2 \nVdash a$ we automatically have $x_2 \Vdash a \to \Box\Diamond a$. $R(x_3) = \varnothing$. And thus it also directly follows that $x_3 \Vdash \Box\Diamond a$. And thus $x_3 \Vdash a \to \Box\Diamond a$. And finally, $M_1 \models a \to \Box\Diamond a$.

Solution: We can also describe the situation using a so-called *satisfiability table.* This is a table that looks like a truth table, but it is slightly different. In the rows it has all the worlds and as columns it has formulas, typically starting with simple formulas on the left, building up to ever more complex formulas on the right. And a one in a certain column and row means that the formula of that column holds in the world of that row. And a zero means that that formula doesn't hold in that world. For filling in the next column, if the operator at hand is a normal propositional operator, then we can simply look at the corresponding columns in the same row, just as we did with our truth tables. However, if the operator at hand is a modal operator, so either a $\Box$ or a $\Diamond$, we have to look at the accessibility relation and see whether the formula behind the modal operator, is true in respectively all accessible worlds, which means rows here, or in at least one accessible world (row here). The benefit of such a table is that it save a lot on the natural language that was used above. Of course, for drawing a conclusion from such a table, one should still use natural language, just as with truth tables.

$\Vdash$	a	$\Box a$	$\Diamond a$	$\Box a \to a$	$\Box\Box a$	$\Box a \to \Box\Box a$	$\Box\Diamond a$	$a \to \Box\Diamond a$
x_0	1	1	1	1	1	1	1	1
x_1	1	1	1	1	1	1	1	1
x_2	0	1	1	0	1	1	1	1
x_3	1	1	0	1	1	1	1	1

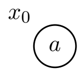
The formulas f for which $M_1 \models f$ are exactly these formulas that have four ones in their columns. So here this holds for $\Box a$, $\Box\Box a$, $\Box a \to \Box\Box a$, $\Box\Diamond a$, and $a \to \Box\Diamond a$.

Exercise 6.H

(i) Can you find a Kripke model M_2 for which $M_2 \models a$ and at the same time $M_2 \not\models \Box a$? If so, provide such a model; otherwise, explain why such a model cannot exist.
Solution: No, such a model would be impossible. Suppose, for contradiction, we would find a model $M_2 = \langle W, R, V \rangle$ in which $M_2 \models a$ holds, as well as $M_2 \not\models \Box a$. Then we have a world $x \in W$ for which $x \Vdash a$ and $x \nVdash \Box a$. And then there are two possible cases for $R(x)$: either it is empty, or not. If $R(x) = \varnothing$ we automatically have $x \Vdash \Box a$ and thus we have our contradiction. So it must be that $R(x) \neq \varnothing$. But for all $y \in R(x)$ we have $y \Vdash a$ because $R(x) \subseteq W$ and $M_2 \models a$ means that $y \Vdash a$ holds for all $y \in W$.

But then we have $x \Vdash \Box a$ and yet again we have a contradiction. So we see that such a model cannot exist.

(ii) Provide a Kripke model M_3 for which $M_3 \models a$ although $M_3 \not\models \Diamond a$. Or, if no such model can exist, explain why.
Solution: Take $M_3 = \langle W, R, V \rangle$ with $W = \{x_0\}$, $R(x_0) = \varnothing$, and $V(x_0) = \{a\}$. In a diagram we can draw this as:

M_3 : 

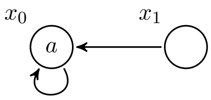
Because $x_0 \Vdash a$ we have $M_3 \models a$. But because $R(x_0) = \varnothing$ we have $x_0 \nVdash \Diamond a$ and thus $M_3 \not\models \Diamond a$.

Exercise 6.I

(i) Does a Kripke model M_4 exist that is serial but not reflexive? If so, provide an example, and otherwise, explain why this is not possible.
Solution: ***Note:*** a model is only serial if it follows axiom D *for all possible formulas* f, and similarly for reflexivity. Therefore, it is not enough to construct a model with just some propositional letter a and then show that the axioms D and T hold in the case $f = a$. The model may still be non-serial or non-reflexive because of other, composite, formulas f! In general it is in fact quite difficult to prove a model serial or reflexive. Instead, it is best to use the characterization given in Section 6.5:

- A model is serial if and only if all worlds have at least one outgoing arrow.
- A model is reflexive if an only if every world has an outgoing arrow leading back to itself.

Back to the exercise. The answer is: yes, such a model can be constructed. Take for instance $M_4 = \langle W, R, V \rangle$ with $W = \{x_0, x_1\}$, $R(x_0) = \{x_0\}$, $R(x_1) = \{x_0\}$, $V(x_0) = \{a\}$, and $V(x_1) = \varnothing$. In a diagram:

M_4 : 

We see that $R(x_0) = R(x_1) = \{x_0\}$, so that every world has an outgoing arrow, and thus the model is serial. However, the model is not reflexive: from x_1 there is no arrow back to x_1 itself. Because we constructed a model that is *not* reflexive, we can demonstrate a specific formula f for which $M_4 \not\models \Box f \to f$. For instance, $f = a$ suffices. Because $R(x_1) = \{x_0\}$ and $a \in V(x_0)$ we have $x_1 \Vdash \Box a$. However, because $a \notin V(x_1)$ we do not have $x_1 \Vdash a$. And thus, $x_1 \nVdash \Box a \to a$. And thus, $M_4 \not\models \Box a \to a$. Hence M_4 is indeed not reflexive.

(ii) Does a Kripke model M_5 exist that is reflexive but not serial? If yes, provide an example, and otherwise, explain why this is not possible.
Solution: No, such a model cannot exist. For M_5 to be reflexive means that $x \in R(x)$ for all $x \in W$: each world has at least an outgoing arrow leading back to itself. But then in particular, at least one outgoing arrow exists from each world, and thus M_5 is automatically serial. Hence M_5 cannot be both reflexive and not serial at the same time.

Exercise 6.J

Express the following sentences in LTL. You may use all operators of LTL in your answers.

(i) *There exists a moment after which the formula a will always be true.*
Solution: $\mathcal{FG}a$ or $\mathcal{FXG}a$. The second version stresses the fact that 'after which' means that a need not hold at the specified moment yet. What do you think, is the first or second a better translation? And what do you think about the formula $\mathcal{XFG}a$? How would you translate that formula back into a sentence?

(ii) *The statements a and b are alternatingly true.*
Solution: The formula

$$\mathcal{G}\left(\left(a \rightarrow \mathcal{X}\left(\neg a \wedge b\right)\right) \wedge \left(b \rightarrow \mathcal{X}\left(\neg b \wedge a\right)\right)\right)$$

would seem OK, but is not. In a situation where $\neg a \wedge \neg b$ is always true, the formula in fact holds. (And of course a and b are then not alternatingly true.) But, with a small addition, we can change the formula to demand that at all times either a or b must hold as well:

$$\mathcal{G}\left(\left(a \vee b\right) \wedge \left(a \rightarrow \mathcal{X}\left(\neg a \wedge b\right)\right) \wedge \left(b \rightarrow \mathcal{X}\left(\neg b \wedge a\right)\right)\right)$$

(iii) *Every time a holds, b holds after a while as well.*
Solution: $\mathcal{G}\left(a \rightarrow \mathcal{X}\,\mathcal{F}b\right)$. Note that we added the $\mathcal{X}$ to express the idea that 'after a while' typically doesn't include the current moment. Another formula that comes close is $\mathcal{G}\left(a\,\mathcal{U}\,\mathcal{X}b\right)$. However, this second one is stronger, because it demands that a stays true until the moment that b holds (that moment not included, by the way). But this is more than our English sentence required.

Exercise 6.K
Define the operator $\mathcal{U}$ in terms of $\mathcal{R}$.

Solution: $f\,\mathcal{U}\,g \equiv \neg\left(\neg f\,\mathcal{R}\,\neg g\right)$. We will prove this. We already know that $\neg\left(\neg f\,\mathcal{U}\,\neg g\right) \equiv f\,\mathcal{R}\,g$. But then $\neg f\,\mathcal{U}\,\neg g \equiv \neg\left(f\,\mathcal{R}\,g\right)$. This holds for all f and g, so in particular it also holds for $\neg f$ and $\neg g$, and thus we get $\neg\neg f\,\mathcal{U}\,\neg\neg g \equiv \neg\left(\neg f\,\mathcal{R}\,\neg g\right)$. And now we are done, because this is the same as: $f\,\mathcal{U}\,g \equiv \neg\left(\neg f\,\mathcal{R}\,\neg g\right)$.

Exercise 6.L
Define the operators $\mathcal{G}$ and $\mathcal{F}$ in terms of $\mathcal{U}$ and $\mathcal{R}$. You may use the propositions $\top$ and $\bot$, which are always true, respectively always false.

Solution: $\mathcal{G}f \equiv \bot\,\mathcal{R}\,f$ and $\mathcal{F}f \equiv \top\,\mathcal{U}\,f$. For the first equality, note that $\bot\,\mathcal{R}\,f$ expresses that f holds until $\bot$ holds. And because $\bot$ never holds, this means that f will always hold. For the second equality, note that $\top\,\mathcal{U}\,f$ expresses two things: first, that f will eventually hold, which is already equivalent to $\mathcal{F}\,f$. And second, that until f holds, $\top$ must hold. But this extra requirement is not actually a requirement at all, because $\top$ always holds already.

Exercise 6.M
Show that all LTL operators (except $\mathcal{X}$) can be defined in terms of the $\mathcal{W}$ operator. Again, you may use the propositions $\top$ and $\bot$.

Solution:

$$\begin{aligned}
f\,\mathcal{R}\,g &\equiv g\,\mathcal{W}\left(f \wedge g\right) \\
\mathcal{G}\,f &\equiv \bot\,\mathcal{R}\,f \\
&\equiv f\,\mathcal{W}\left(\bot \wedge f\right) \\
&\equiv f\,\mathcal{W}\,\bot \\
\mathcal{F}\,f &\equiv \neg\,\mathcal{G}\,\neg f \\
&\equiv \neg\left(\neg f\,\mathcal{W}\,\bot\right) \\
f\,\mathcal{U}\,g &\equiv f\,\mathcal{W}\,g \wedge \mathcal{F}\,g \\
&\equiv f\,\mathcal{W}\,g \wedge \neg\left(\neg g\,\mathcal{W}\,\bot\right)
\end{aligned}$$

Exercise 6.N
Consider the LTL Kripke model $M_7 = \langle W, R, V\rangle$. So we know that $W = \{x_i \mid i \in \mathbb{N}\}$ and $x_j \in R\left(x_i\right)$ if $i \leq j$. Now define V as follows:

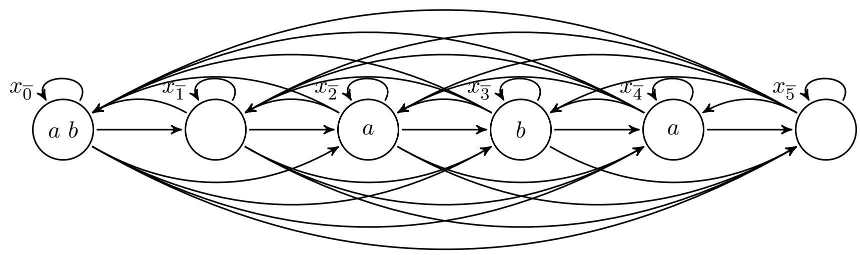
$$\begin{aligned}
a \in V\left(x_i\right) &\quad \text{iff} \quad i \text{ is a multiple of two} \\
b \in V\left(x_i\right) &\quad \text{iff} \quad i \text{ is a multiple of three}
\end{aligned}$$

Check whether the following properties hold:

(i) $x_0 \Vdash \mathcal{F}(a \wedge b)$
Solution: Because $x_6 \in R(x_0)$ and $x_6 \Vdash a$ and $x_6 \Vdash b$, it follows that $x_0 \Vdash \mathcal{F}(a \wedge b)$.

(ii) $x_6 \Vdash \mathcal{G}(a \vee b)$
Solution: Because $x_7 \in R(x_6)$ and $x_7 \nVdash a$ and $x_7 \nVdash b$, it follows that $x_7 \nVdash a \vee b$. And hence $x_6 \nVdash \mathcal{G}(a \vee b)$.

(iii) $M_7 \vDash \mathcal{G}\mathcal{F}(a\,\mathcal{U}\,b)$
Solution: Let $x_i \in W$. Because of $\mathcal{G}$ we have to show that for all $j \geq i$ it holds that $x_j \Vdash \mathcal{F}(a\,\mathcal{U}\,b)$. And because of $\mathcal{F}$ this means that we have to show that for all $j \geq i$ there exists $k \geq j$ such that $x_k \Vdash a\,\mathcal{U}\,b$. Hence because of $\mathcal{U}$ we have to show that for all $j \geq i$ there exists $k \geq j$ and there exists $l \geq k$ such that $x_l \Vdash b$ and for all m such that $k \leq m < l$ it holds that $x_m \Vdash a$. Note that it doesn't matter what the specific value of i is. All that matters is that we find and algorithm to find k and l given any j and i where $j \geq i$. We do this by doing our computations modulo 6. This leads to the following Kripke model:



We now make a case distinction on the remainder of j modulo 6. We write $a \equiv_6 b$ to indicate that $a \mod 6 = b$.

- Case $j \equiv_6 0$. Take $k = j + 2$ and $l = j + 3$. Because all m such that $k \leq m < l$ is nothing but $m = k$ it follows that $x_l \Vdash b$ and for all m such that $k \leq m < l$ it follows that $x_m \Vdash a$. So we have that $x_k \Vdash a\,\mathcal{U}\,b$. And in particular we have that $x_j \Vdash \mathcal{F}(a\,\mathcal{U}\,b)$.
- Case $j \equiv_6 1$. Take $k = j + 1$ and $l = j + 2$.
- Case $j \equiv_6 2$. Take $k = j$ and $l = j + 1$.
- Case $j \equiv_6 3$. Take $k = j + 5$ and $l = j + 6$.
- Case $j \equiv_6 4$. Take $k = j + 4$ and $l = j + 5$.
- Case $j \equiv_6 5$. Take $k = j + 3$ and $l = j + 4$.

Exercise 6.O
Which of the axiom schemes in the table on page 112 hold in LTL?
Solution:

- The axiom for the 'distributive property' holds. If we know that it is globally true that $f \to g$ holds, then we know that in each world from now on the formula $f \to g$ holds. And if we know in addition that f is globally true, then we know that in each world from now of the formula f holds. But combined this means that in each world from now on the formula g holds. But this means that g is globally true.

- The axiom for the 'reflexive property' holds. Because if we know that f is globally true, we know that in each world from now on, including now, formula f holds. So in particular formula f holds now.

- The axiom for the 'symmetry property' does not hold. The axiom states that if the formula f holds now, it will globally hold that sometimes this formula f will hold again. This actually means that formula f holds infinitely often. In particular we can use this counterexample: Assume $a \in V(x_0)$ and $a \notin V(x_i)$ where $i \geq 1$. So $x_0 \Vdash a$. Obviously $x_1 \nVdash \mathcal{F}\, a$. But then $x_0 \nVdash \mathcal{G}\, \mathcal{F}\, a$. And hence $x_0 \nVdash a \rightarrow \mathcal{G}\, \mathcal{F}\, a$.

- The axiom for the 'transitive property' holds. If we have that f holds globally, then $x_i \Vdash f$ for all $i \geq 0$. But then for all $j \geq 0$ we have that $x_i \Vdash f$ for all $i \geq j$. So in each x_j we have that $\mathcal{G}\, f$ holds. But this means that $\mathcal{G}\, \mathcal{G}\, f$ holds in each world x_i with $i \geq 0$. Hence it is globally true that f is globally true.

- The axiom for the 'Euclidean property' does not hold. We can use the same counter example as for the 'symmetric property'.

- The axiom for the 'serial property' holds. Due to the specific Kripke model we know that each world x_i has infinitely many outgoing arrows to the worlds x_j where $j \geq i$.