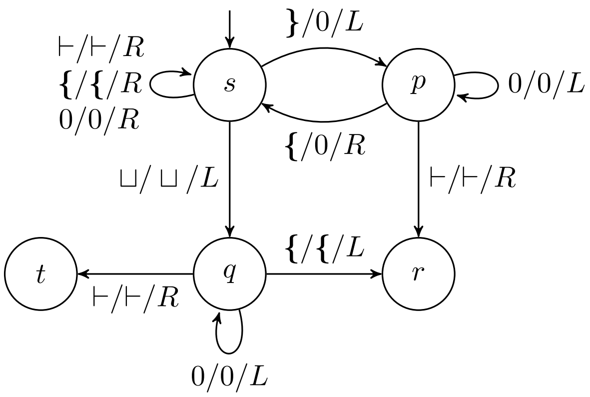
1) Specify a Turing machine that checks for balanced brackets. We only consider the brackets "**{**" and "**}**". The Turing machine should accept if the brackets are balanced e.g. **{{}{}}{}** or reject otherwise e.g. **{{}**.

Solution:
Consider the Turing machine $M = (\{s, p, q, t, r\}, \{\mathbf{\{}, \mathbf{\}}\}, \{\vdash, \sqcup, \mathbf{\{}, \mathbf{\}}, 0\}, \vdash, \sqcup, \delta, s, t, r)$ where δ is specified by following transition table:

	$\vdash$	**{**	**}**	0	$\sqcup$
s	$(s, \vdash, R)$	$(s, \mathbf{\{}, R)$	$(p, 0, L)$	$(s, 0, R)$	$(q, \sqcup, L)$
p	$(r, \vdash, R)$	$(s, 0, R)$	$\cdot$	$(p, 0, L)$	$\cdot$
q	$(t, \vdash, R)$	$(r, \mathbf{\{}, L)$	$\cdot$	$(q, 0, L)$	$\cdot$

The following shows a graphical representation of the same automaton:



Remark: Irrelevant transitions (e.g. those that will never be reached, or the loops of the accepting/rejecting states) have been omitted for better readability.

Informal explanation: In state s the read/write-head moves right until we find a closing bracket "}". Then we overwrite it with 0 and change to state p. In state p the read/write-head moves left and searches for the matching opening bracket "{". If we find it, we also overwrite with 0 and change to state s again.
If we went through the whole input we switch to state q that checks if there are other symbols than 0 on the tape. If so we reject, otherwise we accept.

2) Given a partial order $\leq$ on a set A we say that $x \in A$ is a minimal element of $\leq$ if there is no $y \in A$ such that $y < x$. Analogously, $x \in A$ is a maximal element if there is no $y \in A$ such that $y > x$.

List the minimal and maximal elements of the following partial orders:

i) the natural order ($\leq$) on $\mathbb{N}$

ii) the divisibility order ($|$) on $\mathbb{N}$

iii) the divisibility order ($|$) on $\mathbb{N} \setminus \{0, 1\}$

iv) the subset order ($\subseteq$) on the subsets of $\{1,2,3,4\}$ that do not contain both 2 and 4

Hint: It may help to sketch the graph of the relations.

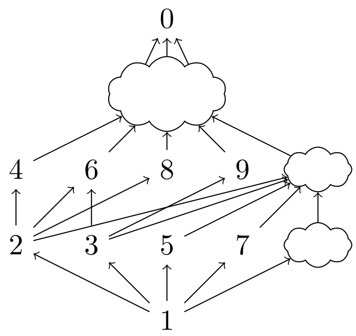
Solution:

i) minimal element: 0, maximal elements: none

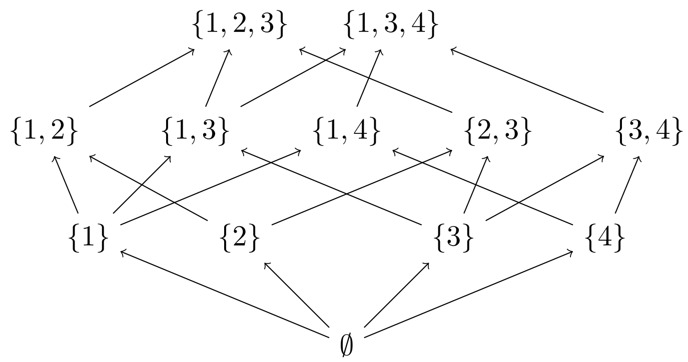
$$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots \longrightarrow n \longrightarrow n+1 \longrightarrow \cdots$$

ii) minimal element: 1, maximal element: 0 (1 divides everything, everything divides 0, in particular we assumed that 0 divides 0 based on the definition given in the slides)

iii) minimal element: the prime numbers, maximal element: none (prime numbers are minimal in $\mathbb{N} \setminus \{1\}$ by definition, no number n is maximal since $n \mid 2n$ but $2n \nmid n$)



iv) minimal element: $\emptyset$, maximal elements: $\{1,2,3\}$, $\{1,3,4\}$



3) Recall that a function $f : A \to B$ is

- "injective" if $\forall x, y \in A.\ f(x) = f(y) \longrightarrow x = y$
- "surjective" if $\forall y \in B.\ \exists x \in A.\ f(x) = y$
- "bijective" if it is injective and surjective

a) For a function $f : \mathbb{R} \to \mathbb{R}$, what does "injective" and "surjective" mean geometrically? How can you tell if f is injective/surjective by looking at the graph of f?

b) Determine for each of the following functions which of these three properties they have:

i) $f_1 : \mathbb{N} \to \mathbb{N},\ f_1(x) = x^2$

ii) $f_2 : \mathbb{Z} \to \mathbb{Z},\ f_2(x) = x^2$

iii) $f_3 : \mathbb{N} \to \mathbb{N},\ f_3(x) = x^3$

iv) $f_4 : \mathbb{R} \to \mathbb{R}_{\geq 0},\ f_4(x) = x^2$

v) $f_5 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0},\ f_5(x) = x^2$

Note: $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers.

c) Show that the function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{Z},\ f(a, b) = a - b$ is surjective.

d) Show that the function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}_{>0},\ f(a, b) = (2a + 1)2^b$ is bijective.

Solution:

a) When looking at the graph of the function, surjective means that every node has at least one incoming edge and injective means at most one incoming edge. When looking at the plot of the function, f is surjective iff every horizontal line intersects with the graph of f at least once. f is injective iff every horizontal line intersects the graph of f at most once.

b) i) injective but not surjective (e.g. there is no $x \in \mathbb{N}$ such that $x^2 = 2$)

ii) neither injective ($f_2(1) = f_2(-1) = 1$) nor surjective (again no x such that $x^2 = 2$)

iii) injective but not surjective (no x such that $x^3 = 2$)

iv) not injective ($f_4(1) = f_4(-1) = 1$), but surjective ($\sqrt{x}^2 = x$ for any real $x \geq 0$)

v) bijective ($x \mapsto \sqrt{x}$ serves as an inverse function)

c) Let $c \in \mathbb{Z}$. If $c \geq 0$, we have $(c, 0) \in \mathbb{N}^2$ and $f(c, 0) = c$. If $c < 0$, we have $(0, -c) \in \mathbb{N}^2$ and $f(0, -c) = c$.

d) We first show surjectivity. Let $c \in \mathbb{N}_{>0}$. Let b be the largest natural number such that 2^b divides c. Let $a' := c/2^b$. We clearly have $a' \in \mathbb{N}$ and a' is an odd number (otherwise 2^{b+1} would also divide c). We can thus let $a := (a' - 1)/2 \in \mathbb{N}$ and obtain

$$f(a, b) = (2a' + 1)2^b = a2^b = \frac{c}{2^b} \cdot 2^b = c\ .$$

For injectivity, suppose we had $a, a', b, b' \in \mathbb{N}$ such that $(2a + 1)2^b = (2a' + 1)2^{b'}$. The largest n such that 2^n divides the left-hand side is then the same as the largest number n such that 2^n divides the right-hand side, but the former is clearly b and the latter is clearly b'. So we have $b = b'$. Cancelling 2^b from both sides of the equation, we get $2a + 1 = 2a' + 1$, and simplification yields $a = a'$.

Remark: If we denote by $\mathrm{ord}_2(c)$ the largest n such that 2^n divides c, the function $g(c) = ((c/2^{\mathrm{ord}}_2(c) - 1)/2, \mathrm{ord}_2(c))$ is the *inverse function* to f, i.e. $g : \mathbb{N}_{>0} \to \mathbb{N} \times \mathbb{N}$ and $g(f(a, b)) = (a, b)$ and $f(g(c)) = c$ for all a, b, c. The existence of an inverse function implies bijectivity and is often an easy way to show bijectivity.

4*) Show that the following modified notions of Turing Machines are equivalent to the ones you know from the lecture:

a) Turing Machines that can also make transitions without moving their read/write head.

Formally: the transition function now has the form

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$$

with the step function extended in the following way:

$$(p, y\sqcup^\infty, n) \to (q, y_0 \dots b \dots y_{m-1}\sqcup^\infty, n) \quad \text{if } \delta(p, y_n) = (q, b, N)$$

b) Turing Machines whose tape is infinitely long in *both* directions (not only to the right) and without a start marker $\vdash$, i.e. the initial tape content for input $x \in \Sigma^*$ is $\sqcup^\infty x \sqcup^\infty$ with the read/write head at the first character of x (or a $\sqcup$ if x is empty).

Formally: the tape content is a function $y : \mathbb{Z} \to \Gamma$ which is initially

$$y_i = \begin{cases} x_i & \text{if } 0 \le i < |x| \\ \sqcup & \text{otherwise} \end{cases}$$

and the step function is

$$(p, y, n) \xrightarrow[M]{1} \begin{cases} (q, y_{n:=b}, n-1) & \text{if } \delta(p, y_n) = (q, b, \mathrm{L}) \\ (q, y_{n:=b}, n+1) & \text{if } \delta(p, y_n) = (q, b, \mathrm{R}) \end{cases}$$

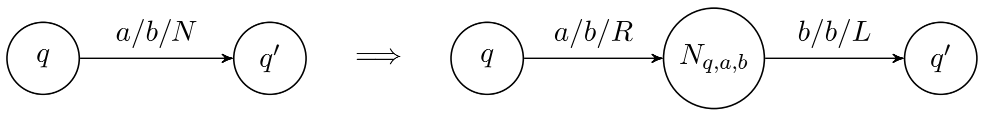
$$\text{where } (y_{n:=b})_i = \begin{cases} b & \text{if } i = n \\ y_i & \text{otherwise} \end{cases}$$

Hint: Show that two notions of Turing Machines are equivalent by transforming a Turing Machine according to one notion into one of the other notion that has the same input–output behaviour. An informal description of this process is sufficient and you do not have to prove that your process really does preserve input–output behaviour.

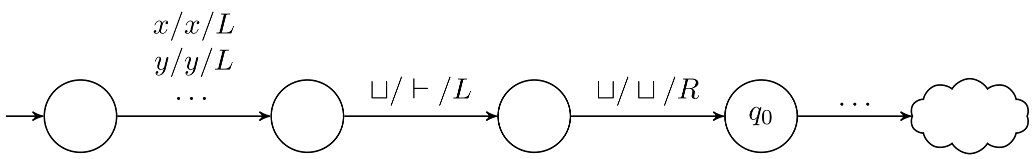
Solution:

a) It is clear that every Turing Machine according to the lecture definition is also one according to the new definition (they just don't use the "N" transitions). We therefore only need to show that every TM using "N" transitions can be transformed into an equivalent one that does not use them.

To do this, we simply replace every "N" transition by a "R" transition to a new state, immediately followed by a "L" transitions like this:



b) **Direction 1:** We can turn a "lecture-style" TM into a "doubly-infinite" TM.
Take a "lecture-style" TM and replace the initial state q_0 by a sequence of states and transitions that go to the left by one step, write a $\vdash$, and then go to q_0, e.g.



Then the "old" TM runs just as before.

Direction 2: We can turn a "doubly-infinite" TM into a "lecture-style" TM.
The only problem is that a transition to the left might encounter the $\vdash$ symbol. We therefore replace every transition of the form $a/b/L$ by a sequence of transitions that checks whether we have reached $\vdash$ and then:

- moves the entire tape content to the right by 1
- moves all the way to the left again
- writes b to the right of the $\vdash$

One subtle issue here is that in order to move the entire tape content to the right, we need to know where the right end of the tape is. This can be done in several way, e.g.

- by introducing a "right-end" marker $\dashv$ and moving it further to the right whenever it is encountered
- or by never writing a $\sqcup$ symbol back to the type but instead a modified symbol $\tilde{\sqcup}$ so that we know which parts of the tape we have already touched.