

Exercises for the lecture

Foundations of Complexity Theory

Solutions to Exercise Sheet 1

for the exercise on October 21th, 2026

Exercise 1.1

Formally define the function 'trim' (see Slide 24 of 1st lecture).
Discuss how to implement it on a Turing machine.

Solution: (for single tape)

$$\begin{aligned} \text{trim, r-trim}: \Gamma^+ \times \mathbb{N} &\rightarrow \Gamma^+ \times \mathbb{N} \\ \text{trim}(b, 1) &= \text{r-trim}(b, 1) = (b, 1) && \forall b \in \Gamma \\ \text{trim}(bB, j) &= \begin{cases} \text{r-trim}(bB, j) & \text{if } j = 1 \text{ or } b \neq \square \\ \text{trim}(B, j - 1) & \text{otherwise} \end{cases} && \forall b \in \Gamma, B \in \Gamma^+, j \in \mathbb{N} \\ \text{r-trim}(Bb, j) &= \begin{cases} (Bb, j) & \text{if } j = |B| + 1 \text{ or } b \neq \square \\ \text{r-trim}(B, j) & \text{otherwise} \end{cases} && \forall b \in \Gamma, B \in \Gamma^+, j \in \mathbb{N} \end{aligned}$$

For an implementation on a Turing machine the problem of how to determine a finite representation of a tape occurs. The tape of a Turing machine always seems to be infinite to the TM and, without special precautions, it cannot be sure that it reached the end of the "utilized" part of the tape to obtain a finite representation of the tape. This problem can be addressed by explicitly keeping and preserving left- and right-end markers on the working tapes. Indeed every TM can be adjusted to mark the used region of the working tapes and preserve these markings over its transitions. Thus, we can, in general, assume that a TM can read the "useful" contents of a tape.

Exercise 1.2

Let $f: \mathbb{N} \rightarrow \mathbb{N}$ total and monotonic increasing function.
Prove or disprove the following statements.

- (a) f -time-bounded TMs are also f -space-bounded.
- (b) f -space-bounded TMs are also f -time-bounded.

Solution: (a) Wrong, but almost true. Since the number of steps is at most $f(|w|)$ for each computation of M for input w , the space-consumption is at most $f(|w|) + 1$. TM that reads input left-to-right (at stops at end) and outputs symbol 'a' on 1st tape in each step requires $|w| + 2$ steps and achieves configuration $(q, a^{|w|+2}\square, |w| + 2, |w| + 3)$ of length $|w| + 3$.

(b) Wrong. TM that stays on input and all other tapes and only writes $\square$ on tapes is space-bounded by constant function $x \mapsto 1$, but not time-bounded.

Exercise 1.3

Let Σ be arbitrary alphabet with $\$ \notin \Sigma$. Show that $\{w\$w \mid w \in \Sigma^*\} \in \mathbf{L}$.
(Principal approach sufficient)

Solution: (In all other cases we reject.) We store positions i, j , and k in the input in binary on tapes 1, 2, and 3. We start with position $i = j = k = 1$ and copy this value to all 3 tapes. Then we move i steps to the right (decrementing i). If we reach the symbol $\$$, then we move j steps to the right and expect the delimiter $\neg$. Should this happen, we accept. Otherwise we reject (since the 2 subwords separated by $\$$ have different lengths). Should we encounter a symbol of Σ during the move right for i steps, then we remember the symbol and move right to the separator $\$$. Then we move right for j steps and check whether the remembered symbol is present. If successful, we move back to the start of the input and copy the contents k of the 3rd tape to tapes 1 and 2 and increase all tape contents by 1. Since we only store 3 positions in the input, the space complexity is logarithmic, so we obtain complexity class $\mathbf{L}$.

Here is a program for input w (without delimiters):

- Test whether $w \in \Sigma^*\$ \Sigma^*$ and $|w|$ odd
- Set $i = 1$ and while $w[i] \neq \$$ increase i
- Set $j = 1$ and while $j < i$ check whether $w[j] = w[i + j]$
- Check whether $i + j = |w|$

Exercise 1.4

Let L be a regular language.

- (a) Determine the complexity of the non-uniform membership problem for L .
(Determine an ideally minimal complexity class $\mathcal{C}$ such that $L \in \mathcal{C}$. In the non-uniform membership problem the language L is constant and we can thus assume any suitable representation of L irrespective of how large or complicated to obtain this representation is. The name "non-uniform" derives from the fact that L is hard-wired in our solution TM, so the resource bounds apply only to the length of w , for which we want to check membership in L .)
- (b) Determine the complexity of the uniform membership problem for L given by an NFA.
(In the uniform variant your Turing machine receives both w , for which membership in L is to be tested, as well as a finite representation of L as input. Specifically, we receive w and an NFA M for L as input and the resource bounds apply to the sum of

the sizes of these two inputs. This is called “uniform” because our TM needs to work for any input w as well as any suitable finite representation (for L or other languages). For example, for the uniform problem to be polynomial time, the solution would need to run in polynomial time in both the length of w as well as the size of the NFA M accepting L .)

Solution: (a) For the non-uniform solution we can simply take a DFA for L , which is also a special TM that does not utilize any working tape. It is known that for every regular language L there exists a DFA that recognizes L and with the help of this DFA we obtain that $L \in \text{DSPACE}(1) \subseteq \text{DSPACE}(\log x) = \mathbf{L}$. Indeed we even have $\mathbf{REG} = \text{DSPACE}(1)$, where $\mathbf{REG}$ is the class of regular languages.

Clearly, the DFA runs in linear time as it advances to the next symbol in every transition and immediately accepts or rejects at the end of the input, so we also obtain $L \in \text{DTIME}(x) = \mathbf{LinTime}$. This time this result does not yield a characterization for all of the regular languages as $\mathbf{REG} \subsetneq \text{DTIME}(x)$. Let L' be the language mentioned in Exercise 1.3. Indeed L' is a good example for the strictness; i.e., $L' \in \text{DTIME}(x)$ but $L' \notin \mathbf{REG}$. In summary we obtain

$$L \in \text{DSPACE}(1) \cap \text{DTIME}(x) .$$

(b) In the uniform variant, the NFA M is now part of the input. We first claim that we can solve this problem in $\mathbf{NL}$. To make the analysis simpler, we assume that each state of the NFA M has at least 1 transition, in which it occurs. Without going into details of a specific encoding (and size measure) of M , this assumption yields $|M| \geq |Q|$, where Q is the set of states of M . We can easily store the current state in space $\log |Q| \leq \log |M|$. We also need to store the current position in the input word (as we need to keep the representation of the NFA in the input) and a counter that we use to approach the position indicated by the current position in the input word. These are positions in the input and can thus be stored in logarithmic space. Let us assume that the reading head on the input is on the first symbol. We first advance the reading head to the stored position in the input word, by copying the stored position to our counter and advancing the reading head while decreasing the counter. At this point we are aware of the input symbol to be processed and the current state, which is also stored on a tape. With this information we now advance the reading head to the NFA part and read its transitions while matching state and input symbol. We nondeterministically select a transition that can be applied in the current state while reading the symbol and update our state on the tape according to the target state of that transition. We also increase the stored position in the input and rewind the input tape. Once the current input symbol is the separator (between input word and NFA representation), there is no more input to process and decide based on the current state whether to accept or reject. To this end, we advance to the NFA part of the input and compare our stored current state against any final state. If we find a match, then we accept and otherwise we reject. This solution utilizes nondeterminism, but never uses more than logarithmic space, so we obtain that the uniform membership problem for NFA is in $\mathbf{NL}$.

For nondeterministic time the approach sketched above (maybe with the NFA copied to a working tape to avoid the moves between input word and NFA representation) yields

that the uniform membership problem for $w \in L(M)$ can be solved nondeterministically in time $O(|w| \cdot |M|)$ as we essentially scan the whole NFA M for a suitable transition to execute for each input symbol of w . There is no additional charge for the comparison of states as this can be done directly while reading the transition (against the stored current state) and the rewind on the stored current state takes the same time, so disappears in the O -notation. Thus the nondeterministic time complexity is linear in the length $|w|$ of the input word w and linear in the size $|M|$ of the input NFA M . However, our input size is $x = |w| + 1 + |M| + 2$ (final +2 for the end markers), but let us stick to $x = |w| + |M|$ for simplicity. Thus we obtain the nondeterministic time complexity $\text{NTIME}(x^2)$ as

$$O(x^2) = O((|w| + |M|)^2) = O(|w|^2 + 2|w| \cdot |M| + |M|^2) \supseteq O(|w| \cdot |M|) .$$

Let $\text{NTIME}(O(x^2)) = \bigcup_{f \in O(x^2)} \text{NTIME}(f)$. The result $\text{NTIME}(x^2) = \text{NTIME}(O(x^2))$ will be proved in the lectures. Overall we obtained that the uniform membership problem is in

$$\mathbf{NL} \cap \text{NTIME}(x^2) .$$

If we want to determine deterministic complexity classes, then we can adjust the approach detailed above to keep a set of states (like in the powerset construction) instead of just a single state as current state set. Since we can store a single state in space $\log |Q| \leq \log |M|$, we can store such a set in space proportional to

$$|Q| \cdot \log |Q| \leq |M| \cdot \log |M|$$

as such a set can contain at most $|Q|$ states. While scanning the NFA for suitable transitions we resolve the nondeterminism in the selection by collecting all target states that can be reached via suitable transitions (instead of just a single one from the nondeterministically selected transition). This requires another set of states (on a separate tape) and matching a state in the transition against our current set of states, which requires no additional space but time proportional to $|M| \cdot \log |M|$ (for a simple direct check) for each input symbol. A similar effort, again without effect for the space complexity, is necessary to check whether a newly identified suitable target state is also already in our collection of target states. Thus rough approximates for the space and time complexity are $O(|M| \cdot \log |M|)$ and $O(|w| \cdot |M|^2 \cdot \log |M|)$, respectively. For the time complexity observe that for every input word position and every transition in the NFA we need to check whether the origin state of that transition is inside our set of current states of size $|M| \cdot \log |M|$. In particular, this proves that the deterministic time complexity is polynomial, so the uniform membership problem for NFA is in

$$\mathbf{NL} \cap \text{NTIME}(x^2) \cap \text{DSPACE}(x \log x) \cap \mathbf{P} .$$

These are not always the best known bounds and with the help of some results of the lectures we will be able to improve some bounds.