Tutorial 13

Exercise 1 (compulsory)

Show that the class PSPACE is closed under union, intersection, concatenation, Kleene star and complement. **Hint:** You can conveniently use the fact that PSPACE=NPSPACE.

Solution:
Use the same constructions like in Exercise 1, Tutorial 12. It is now easy to argue that these constructions are implementable in polynomial space, provided that the machines M_1 and M_2 use only polynomial space. The construction for complement was presented in Lecture 13, slide 11.

Exercise 2 (compulsory)

Prove that co-NP $\subseteq$ PSPACE.

Solution:
Let $L \in$ co-NP. By definition $\overline{L}$ belongs to NP and so there is a polynomial time nondeterministic TM deciding $\overline{L}$. Clearly, polynomial time TM cannot scan more then polynomially many tape cells, which implies that $\overline{L}$ is decidable in nondeterministic polynomial space. By Savitch's theorem, $\overline{L}$ is decidable in polynomial space also on a deterministic TM. By swapping the accept and reject states on that deterministic machine, we get a polynomial space deterministic decider for L. This means by definition that $L \in$ PSPACE.

Exercise 3 (compulsory)

Let f be a function such that $f(n) \geq n$. Which of the following claims are true?

1. *CLIQUE*$\in$ PSPACE
2. *VERTEX-COVER* $\notin$ PSPACE
3. *CLIQUE*$\in$ SPACE(n)
4. *CLIQUE*$\notin$ NSPACE(n)
5. SPACE$(f(n)) \subseteq$ NSPACE$(f(n))$
6. NSPACE$(f(n)) \subseteq$ SPACE$(f^2(n))$
7. NSPACE$(f(n)) \subseteq$ SPACE$(f^3(n))$
8. SPACE$(f(n)) \subseteq$ NSPACE$(f^2(n))$

Solution:

1. True.
2. False
3. True.
4. False.
5. True (trivial).

6. True (Savitch's Theorem).
7. True (weaker statement than Savitch's Theorem).
8. True (trivial).

Exercise 4 (compulsory)

Assume a deterministic decider M with space complexity n^3. How many steps does the machine M at most perform on an input w of length n?

Solution:
If the number of scanned cells (and hence possible positions of the head) is bounded by n^3, then there are at most $qn^3g^{n^3}$ different configurations where q is the number of states, and g is the number of tape symbols. Because q and g are constants, we get that there are at most $2^{O(n^3)}$ different configurations. A deterministic decider cannot enter the same configuration twice (otherwise it would start looping), which means that $2^{O(n^3)}$ provides also an upper bound on the number of computation steps.

Exercise 5 (compulsory)

Argue that given any TM M, we can without loss of generality assume that M satisfies the unique-accept-configuration condition:

> whenever M enters an accepting configuration, then the configuration is exactly this one: $q_{accept}\sqcup$.

In other words, provide a polynomial time reduction from A_{TM} to $A_{TM,unique}$, where

$A_{TM,unique} \stackrel{\text{def}}{=}$
$\{\langle M, w\rangle \mid$ M is a TM which accepts w and M satisfies the unique-accept-configuration condition $\}$.

Solution:
Assume a TM M and its input w. We will in polynomial time construct a TM M' which satisfies the unique-accept-configuration condition and such that M accepts w if and only if M' accepts w.

In order to construct M' we simply modify the machine M in such a way that any transition in M which enters the q_{accept} state will first enter a newly added control state in which the machine M' is going to completely clean the tape (on every used cell write the symbol $\sqcup$) and only then enter the q_{accept} state. Such machine M' clearly satisfies our requirement.

Exercise 6 (optional)

Let L be a language that can be decided by a deterministic Turing machine in space $f(n)$ where $f(n) \geq n$ for all n. Prove that for any real number c, $0 < c < 1$, L can be decided by a deterministic Turing machine M_c with space complexity $cf(n)$. **Note:** Here we consider the exact space complexity, not an approximation using O-notation, so you cannot simply write that O-notation allows us to disregard constant factors.

Solution:
We introduce a new tape alphabet for M_c. The characters in the new tape alphabet are vectors of length k of symbols from the original tape alphabet. In this way we ensure that M_c will use at most $\lceil \frac{f(n)}{k} \rceil + 1$ tape cells during its computation, where M used $f(n)$ cells. Moreover we must mark where the head would

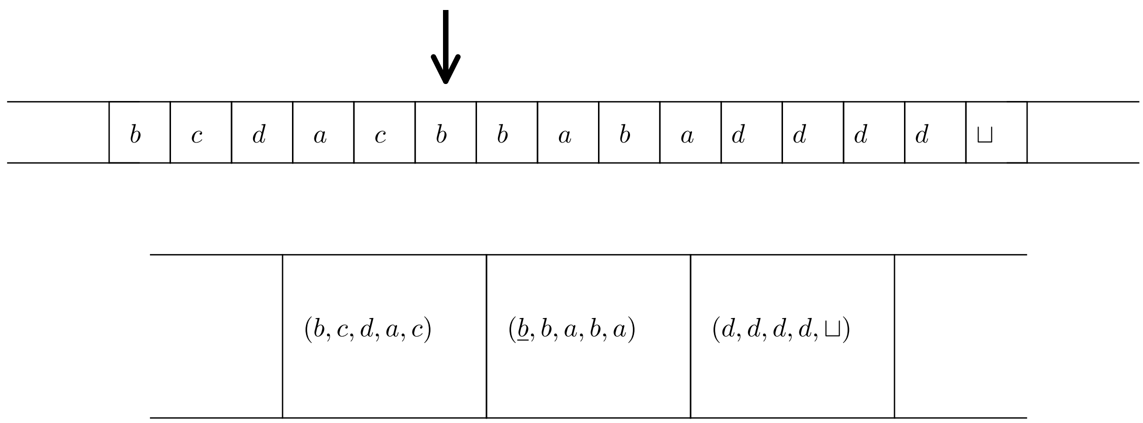

Figure 1: Original tape of M and its representation in M_c. The arrow indicates the head position.

have been on the original tape; we do this by overlining the symbol in question. Let M have tape alphabet Γ and let $\overline{\Gamma} = \{\overline{a} \mid a \in \Gamma\}$. We choose $k = \lceil \frac{1}{c} \rceil$, and the new tape alphabet is then $\Gamma \cup (\Gamma \cup \overline{\Gamma})^k$.

M_c starts its computation by rewriting the contents of the input tape so that it contains a string of k-vectors corresponding to the original input. Figure 1 shows an example of the resulting tape content. All we need to do now is to explain the subsequent operation of M_c as it simulates M.

Assume that we consider the tape cell containing the k-vector $(x_1, \ldots, x_k)$ and that M in state q would have its head on symbol x_i. Then we overline x_i. The simulation now depends on x_i:

- If x_i is the leftmost symbol, i.e. $i = 1$, and M would move its head to the left and change state to q'. Here M_c must update two cells: The current cell and the cell to its left. M_c must also move its head left, and the symbol to be overlined is now the rightmost symbol in the cell that the head is now pointing to. Finally, M_c must change its state to q'.

- If $i = k$, and M would move its head right, and change state to q', M_c must update two cells: the current cell and the cell to its right. M_c must also move its head right, and the symbol to be overlined is now the leftmost symbol in the cell that the head is now pointing to. Finally, M_c must change its state to q'.

- In all other cases M_c only needs to update the vector $(x_1, \ldots, x_k)$ by changing it and overlining a different symbol corresponding to the new head position of M. In this case M_c will mot move its head, only change state to q'.

When M_c moves its head to a blank tape cell, the blank symbol must be replaced by the k-vector$(\sqcup, \ldots, \sqcup)$.