Tutorial 03 – Time and space

Exercise 1. *Tally Language (Berman's Theorem, 1974)*

A language is said to be *tally* (or unary), if it is included in a unary alphabet $\{a\}^*$ for a fixed symbol a.

Definition (Subset-Sum)**.** Given n numbers $v_1, \dots, v_n \in \mathbb{Z}$, and a *target* number $T \in \mathbb{Z}$, we need to decide whether there exists a nonempty subset $S \subseteq [\![1, n]\!]$ such that $\sum_{i \in S} v_i = T$. The problem size is $|T|_2 + \sum_{i=1}^{n} |v_i|_2$.

1. Prove that Subset-Sum is NP-complete by reduction from 3-SAT
2. Let Unary-Subset-Sum be the tally variant of Subset-Sum where all numbers are represented by their unary representation. Show that Unary-Subset-Sum is in P.
3. Show that there exist undecidable tally languages.
4. Show that if there exists an NP-hard tally language, then $\mathsf{P} = \mathsf{NP}$.

Exercise 2. *Universal Turing machines*

We want to study the space complexity of universal Turing machines.

1. Prove that a TM on alphabet Γ that has k work tapes and uses at most $s(|x|)$ work cells on input x can be simulated by a TM machine that has one work tape, is on alphabet $\{0, 1, , \triangleright\}$, and uses $O\big(s(|x|)\big)$ work cells for every input x.
   Hint: Look at the construction done in TD 01
2. Prove that there exists a universal Turing machine $\mathcal{U}(x, \alpha)$ such that if the machine M_α on input x uses at most $s(|x|)$ work cells, then $\mathcal{U}(x, \alpha)$ uses at most $C_\alpha s(|x|)$ work cells, where $C_\alpha > 0$ depends on α but not on x.
   Hint: Look at the construction done in TD 01 for 2-tape to 1-tape TMs

Exercise 3.

Let $x, y, z \in \{0, 1\}^n$ be two binary numbers. Show that we can arrange them on a tape in such a way that we can check if $x + y = z$ in $O(1)$ space.

Exercise 4.

Let $x, y \in \{0, 1\}^n$ be two binary numbers. Show that we can compute their product xy in $O(\log n)$ space.

Exercise 5. *Time(space)'s up*

We recall that a function $f \colon \mathbb{N} \to \mathbb{N}$ is *space-constructible* if there exists a Turing machine that given 1^n computes $1^{f(n)}$ and uses $O(f(n))$ space. Similarly, a function $f \colon \mathbb{N} \to \mathbb{N}$ is *time-constructible* if there exists a Turing machine that given 1^n computes $1^{f(n)}$ in time $O(f(n))$.

1. Show that we can convert between the binary and unary representation of $n \in \mathbb{N}$ in $O(n)$ time and $O(\log n)$ space. Deduce that if $f(n) \geq cn$ for all n, then we can replace the unary representation by the binary representation in the definition of time-constructible functions.

2. Show that $n \mapsto n$, $n \mapsto n^2$, $n \mapsto 2^n$, $n \mapsto n\lfloor \log n \rfloor$ are time-constructible functions.

3. Show that $n \mapsto \lfloor \log n \rfloor$ is space-constructible.

Exercise 6. L, NL

Let $\mathsf{L} := \mathsf{DSPACE}(\log n)$ and $\mathsf{NL} := \mathsf{NSPACE}(\log n)$.

1. Show that $\textsc{Even} = \{x \mid x \text{ has an even number of 1s}\}$ is in L.

2. Show that the language of balanced parentheses (with only one kind of parenthesis) is in L.

3. Show that $\textsc{Path} = \{\langle G = (V, E), x \in V, y \in V\rangle \mid \text{There exists a path between } x \text{ and } y \text{ in } G\}$ is in NL.

Exercise 7. *Little space is no space at all!*

We are going to show that a language that can be recognized in $o(\log \log n)$ space can in fact be recognized in $O(1)$ space. This implies, e.g., that $\mathsf{DSPACE}(\sqrt{\log\log n}) = \mathsf{DSPACE}(1)$.

1. Let $\mathcal{L} \notin \mathsf{DSPACE}(1)$ and suppose that M is a Turing machine that recognizes $\mathcal{L}$ in space $o(\log \log n)$. For every $k \in \mathbb{N}$ show that there exists $x \in \mathcal{L}$ such that $M(x)$ uses at least k cells on the work tape during the computation. Furthermore, if $x_k \in \mathcal{L}$ denotes the shortest word with this property, then $\lim_{k\to\infty} |x_k| \to \infty$.

2. At every step of the computation of $M(x)$ we define the current *internal configuration* of M as the tuple (q, y), where q is the current state of M and y encodes the current contents of the work tapes of M and the positions of the work heads. Show that M has $o(\log |x|)$ different internal configurations during the computation of $M(x)$.

3. We define the ith *crossing sequence* $\mathcal{C}_i(x)$ of a M on an input x, $1 \leq i \leq |x| - 1$, as a vector $(c_1, c_2, \ldots, c_m)$, where c_1 is the internal configuration of M when the input head first crossed from a cell i to a cell $i + 1$ on the input tape, c_2 is the internal configuration of M when the input head first crossed from a cell $i + 1$ to i, etc. (so that the odd elements denote configurations of M after crossing ith cell left-to-right, and the even ones – after crossing right-to-left from $(i + 1)$th cell to ith). Show that there are $o(|x|)$ different crossing-sequences that appear on an input x.

4. Show that the crossing sequences of M on x_k are pairwise different. Conclude.

Exercise 8. *Tally Languages Acceptable with Sublogarithmic Space*

Let

$$C = \{a^n | F(n) \text{ is a power of } 2\}$$

where

$$F(n) = \min\{i | i \text{ does not divide } n\}$$

Then we want to prove that $C \in DSPACE(\log\log n)$

1. Let $G(k) = \operatorname{lcm}\{j | j \leqslant k\}$. Show that for $k \geqslant 2$ there exists constants $c_1, c_2 > 1$, such that

$$c_1^k \leqslant G(k) \leqslant c_2^k$$

Hint: Use without proving the fact that $G(k) = \prod_{\textit{all primes } p} p^{\lfloor \log_p k \rfloor}$. *You can also use the prime number theorem.*

2. Now show that

$$F(k) \leqslant c \log k \text{ for some constant } c \text{ and } k \geqslant 2$$

3. Use the above bound on $F(k)$ to give a $O(\log \log n)$ space algorithm.

4. [Bonus question:] Show that C is non-regular
Hint: Observe that F(G(k)) > k. Choose n carefully such that $F(n) = 2^r$. Then use pumping lemma.

Why important? Interesting question in homework. $DSPACE(1) = REG$ where REG is the class of regular languages. This along with Exercise 7 shows that $\Omega(\log \log n)$ space is required to recognize any non-regular language. This shows that for Exercise 8, you cannot do better.