— *Homework I* —

Computational Complexity

Hugo SALOU

1 Palindromes.

Question 1.0.1. *Let $y, z \in \mathrm{PaddedPal}_k$ and assume for some $i, j \in [\![k+1, 2k]\!]$, $C_i(y) = C_j(z)$. Show that $y = z$.*

As $y, z \in \mathrm{PaddedPal}_k$, we can write $y = u0^k\bar{u}$ and $z = v0^k\bar{v}$ with $u, v \in \Sigma^k$. Consider an accepting execution r for y, and s for z.

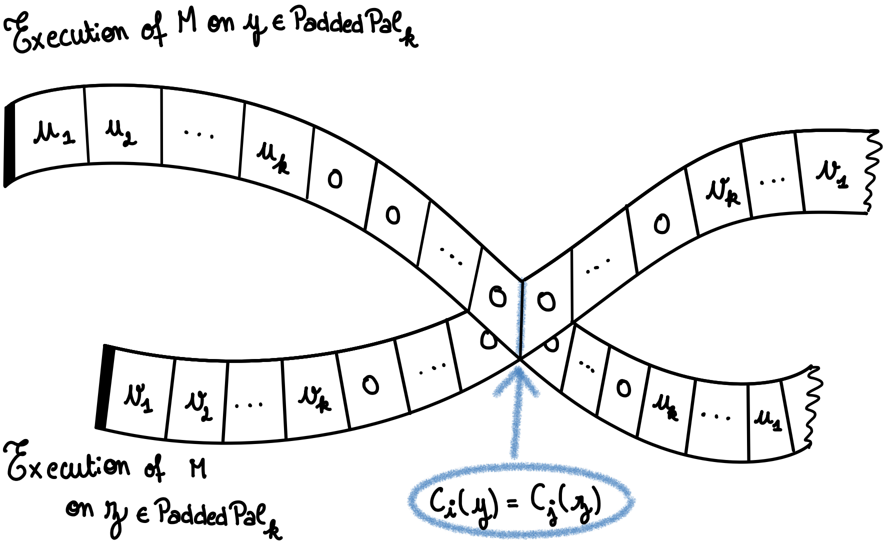

Figure 1 | *Intersection of two crossing sequences*

Note: Here, it is shown for the initial tape, but we have a similar situation throughout the execution.

We can assume that r will accept while its head is before i, or s after j (if that is not the case, swap y and z).

We can create an accepting execution t for $w := u0^\ell\bar{v}$ where $\ell = (i-k) + (2k-j) = k+i-j$: start like r, and whenever you try to cross from cell i to cell $i+1$, jump to cell $j+1$ and act like s (we have the same state as the crossing sequences are equal), and whenever you try to cross from $j+1$ to j, jump to i act like s again. This execution is, indeed, accepting, as either r will accept while its head is before i (as, in this case, the execution t behaves like r), either t will accept while its head is after j (as, in this case, t behaves like s). Thus w is a palindrome, and so $u = v$, and we can finally conclude $y = z$.

Question 1.0.2. *Show that for any $y \in \mathsf{PaddedPal}_k$, there is an $i \in [\![k+1, 2k]\!]$ such that $|C_i(y)| \leq T(3k)/k$.*

As each element of some $C_i(y)$ is a step during the execution of M on a y whose length is $3k$ (and we don't have duplicates when considering different is), we know that

$$\sum_{i=k+1}^{2k} |C_i(y)| \leq T(3k),$$

and if we take the average, we know at least one element will be less than or equal to the average: there is some $i \in [\![k+1, 2k]\!]$ such that

$$|C_i(y)| \leq T(3k)/k.$$

Question 1.0.3. *Conclude by observing that for each $y \in \mathsf{PaddedPal}_k$, we can associate a distinct sequence of states of size at most $T(3k)/k$.*

By question 2, we can associate some $i = i(y)$ such that $|C_{i(y)}(y)| \leq T(3k)/k$ to every $y \in \mathsf{PaddedPal}_k$. Then, the mapping $y \mapsto C_{i(y)}(y)$ is injective by question 1, thus giving a distinct sequence of states of size at most $T(3k)/k$ to every $y \in \mathsf{PaddedPal}_k$.

We know that $|\mathsf{PaddedPal}_k| = |\Sigma|^k$ and that the number of state sequences of length at most ℓ is less than $\sum_{i=0}^{\ell} |Q|^i \leq |Q|^{\ell+1}$. Therefore, with the association made previously, we know that $|\Sigma|^k \leq |Q|^{(T(3k)/k)+1}$, and thus

$$k \log|\Sigma| \leq \left(\frac{T(3k)}{k} + 1\right) \log|Q|, \qquad \text{so} \qquad k^2 \underbrace{\left(\frac{\log|\Sigma|}{\log|Q|} - \frac{1}{k}\right)}_{9\alpha^{\mathrm{a}}} \leq T(3k).$$

Writing $n = 3k$, we can finally conclude that $T(n) \geq \alpha n^2$ for infinitely many n.

2 ### The Padding Technique ###.

2.1 Problem 1.

Question 2.1.1. *Show that if* P = NP *then* EXPTIME = NEXPTIME.[b]

Let us assume P = NP. We already know EXPTIME ⊆ NEXPTIME, so let us show the converse. Consider $L \in$ NEXPTIME, and M a non-deterministic Turing machine recognizing M and that runs in time $C \cdot 2^{n^k}$ for some integer k and some constant C. Then, consider the following language on $\{\mathtt{0}, \mathtt{1}, \#\}$:[c]

$$L' := \left\{ w\#^{C \cdot 2^{|w|^k}} \;\middle|\; w \in L \right\}.$$

We use the #s as fuel for M to run in polynomial time in the length of

$$\bar{w} := w\#^{C \cdot 2^{|w|^k}}.$$

Define M' to be the non-deterministic machine that follows exactly M's behaviour except that it consider # as a blank character □ (so it'll only read the "w part" of $\bar{w}$). Therefore, the language recognized by M' is exactly L', and its runtime is in polynomial time (as the "extra length" of $\bar{w}$ allows the execution of M on w to happen in polynomial time in $|\bar{w}|$). Thus,

$$L' \in \mathsf{NP} = \mathsf{P}.$$

As $L' \in$ P, there exists some deterministic poly-time Turing machine N recognizing L'. Let us deduce a exponential-time deterministic Turing machine N' recognizing L: start by writing w on N's input tape, compute $C \cdot 2^{n^k}$ (which is time-constructible by a construction similar to Tutorial 3, Exercise 2) in unary (using the # character), adding that on the input tape after w, and finally start the execution of N.

[b] EXPTIME $= \bigcup_{k \in \mathbb{N}} \mathrm{DTIME}(2^{n^k})$ and NEXPTIME $= \bigcup_{k \in \mathbb{N}} \mathrm{NTIME}(2^{n^k})$.

[c] We can always encode such an alphabet with the binary alphabet using twice as many cells.

The input tape of N contains $\bar{w}$ and so we have that N will test whether $\bar{w} \in L'$, *i.e.* whether $w \in L$. Furthermore, N runs in exponential time and is deterministic. Therefore, $L \in$ EXPTIME.

We can conclude that NEXPTIME ⊆ EXPTIME.

2.2 Problem 2.

Question 2.2.1. *Suppose* $\mathrm{NSPACE}(n^4) \subseteq \mathrm{NSPACE}(n^3)$*. Let M be a Turing machine running in* $\mathrm{O}(n^5)$ *space. Show that there is a Turing machine N accepting the language* $L(M)$ *and running in* $\mathrm{O}(n^4)$ *space. Consequently,* $\mathrm{NSPACE}(n^5) \subseteq \mathrm{NSPACE}(n^4)$*.*

Let M be a non-deterministic Turing machine that runs in $\mathrm{O}(n^5)$ space. We define the language

$$A := \left\{ w\#^{\lceil n^{5/4} \rceil - n} \;\middle|\; w \in L(M), n = |w| \right\}.$$

In the following, we will write $\bar{w}$ the word associated with $w \in L(M)$ in language A, and $m = \mathrm{O}(n^{5/4})$ its length.

We have that $A \in \mathrm{NSPACE}(m^4)$ as we can simply ignore the # and run M on the input tape. The extra #s allows M to use more space, and $m^4 = \mathrm{O}(n^5)$.

Thus we have that $A \in \mathrm{NSPACE}(m^3)$ by the assumption that $\mathrm{NSPACE}(m^4) \subseteq \mathrm{NSPACE}(m^3)$. Also, the computation of $\lceil n^{5/4} \rceil - n$ can be done in space $\mathrm{O}(n)$ (non-output tapes only), and thus, so is the computation of $\bar{w}$. Therefore, considering N the non-deterministic Turing machine that:

1. computes $\bar{w}$ in space $\mathrm{O}(n)$;
2. simulates P on $\bar{w}$ in space $m^3 = \mathrm{O}(n^{15/4}) = \mathrm{O}(n^4)$,

where P is a non-deterministic Turing machine that recognizes A' in space $\mathrm{O}(m^3)$. We have that N runs in space $\mathrm{O}(n^4)$ and its language is exactly $L(M)$. So, $L(M) \in \mathrm{NSPACE}(n^4)$.

We can conclude that

$$\mathrm{NSPACE}(n^5) \subseteq \mathrm{NSPACE}(n^4).$$

Question 2.2.2. *Using Savitch's theorem and the previous result, show that* $\mathrm{NSPACE}(n^3) \subsetneq \mathrm{NSPACE}(n^4)$ *has to be true.*

By induction on $k \in \mathbb{N}$, and with a proof very similar to the one in the previous question, we can show that if $\mathrm{NSPACE}(n^4) \subseteq \mathrm{NSPACE}(n^3)$, then

$$\mathrm{NSPACE}(n^{k+4}) \subseteq \mathrm{NSPACE}(n^{k+3}) \subseteq \cdots \subseteq \mathrm{NSPACE}(n^3),$$

and the reverse inclusions are clearly true, we have the equality of all these classes of languages. Also, by Savitch's theorem (for the right inclusion), we have that

$$\mathrm{NSPACE}(n^3) \subseteq \mathrm{DSPACE}(n^6) \subseteq \mathrm{NSPACE}(n^6) = \mathrm{NSPACE}(n^3),$$

thus $\mathrm{DSPACE}(n^6) = \mathrm{NSPACE}(n^6)$. Similarly, we have that $\mathrm{DSPACE}(n^8) = \mathrm{NSPACE}(n^8)$ as

$$\mathrm{NSPACE}(n^4) \subseteq \mathrm{DSPACE}(n^8) \subseteq \mathrm{NSPACE}(n^8) = \mathrm{NSPACE}(n^4).$$

Thus $\mathrm{DSPACE}(n^6) = \mathrm{DSPACE}(n^8)$ which is absurd as $n^6 = o(n^8)$ (and n^8 is space-constructible) using the space hierarchy.

Question 2.2.3. *Can you now show that* $\mathrm{NSPACE}(n^s) \subsetneq \mathrm{NSPACE}(n^t)$ *for all* $0 \leq s < t, s, t \in \mathbb{N}$*?*

Assume, by contradiction, that $\mathrm{NSPACE}(n^s) = \mathrm{NSPACE}(n^t)$ for some integers $s < t$. As

$$\mathrm{NSPACE}(n^t) = \mathrm{NSPACE}(n^s) \subseteq \mathrm{NSPACE}(n^{s+1}) \subseteq \cdots \subseteq \mathrm{NSPACE}(n^t)$$

form a chain of equalities, we have that

$$\mathrm{NSPACE}(n^{s+1}) = \mathrm{NSPACE}(n^s).$$

Very similarly to the previous questions, we can show that

$$A := \left\{ w\#^{n^{(s+2)/(s+1)}-n} \;\middle|\; w \in L(M), n = |w| \right\}$$

is in $\mathrm{NSPACE}(m^{s+1})$ where M is a non-deterministic Turing machine that uses at most $\mathrm{O}(n^{s+1})$ space. Thus, we can show that $L(M) \in \mathrm{NSPACE}(n^{s+1})$ as we can simulate M on $\bar{w}$ using

$$\mathrm{O}(m^{s+1}) = \mathrm{O}\big((n^{(s+1)/(s+2)})^{s+1}\big) = \mathrm{O}(n^{s+1})$$

space. Doing this for $t = s+1, s+2, s+3, \ldots$, we obtain that

$$\mathrm{NSPACE}(n^s) = \mathrm{NSPACE}(n^{s+1}) = \cdots = \mathrm{NSPACE}(n^{s+k}) = \cdots, \tag{$*$}$$

and thus $\mathrm{DSPACE}(n^{2s}) = \mathrm{DSPACE}(n^{2(s+1)})$ which contradicts the space hierarchy ($n^{2(s+1)}$ is space-contractible and $n^{2s} = \mathrm{o}(n^{2(s+1)})$).

We can conclude that

$$\mathrm{NSPACE}(n^s) \subsetneq \mathrm{NSPACE}(n^t),$$

for all integers $s < t$.

> **Note.** After $(*)$, we could conclude that
>
> $$\mathrm{PSPACE} = \bigcup_{k\in\mathbb{N}} \mathrm{DSPACE}(n^k) \subseteq \bigcup_{k\in\mathbb{N}} \mathrm{NSPACE}(n^k) = \mathrm{NSPACE}(n^s),$$
>
> and thus we get the contradiction $\mathrm{PSPACE} = \mathrm{DSPACE}(n^{2s})$.