

— Homework I —

Computational Complexity

Hugo SALOU

1 Palindromes.

Question 1.0.1. Let $y, z \in \text{PaddedPal}_k$ and assume for some $i, j \in \llbracket k+1, 2k \rrbracket$, $C_i(y) = C_j(z)$. Show that $y = z$.

As $y, z \in \text{PaddedPal}_k$, 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 .

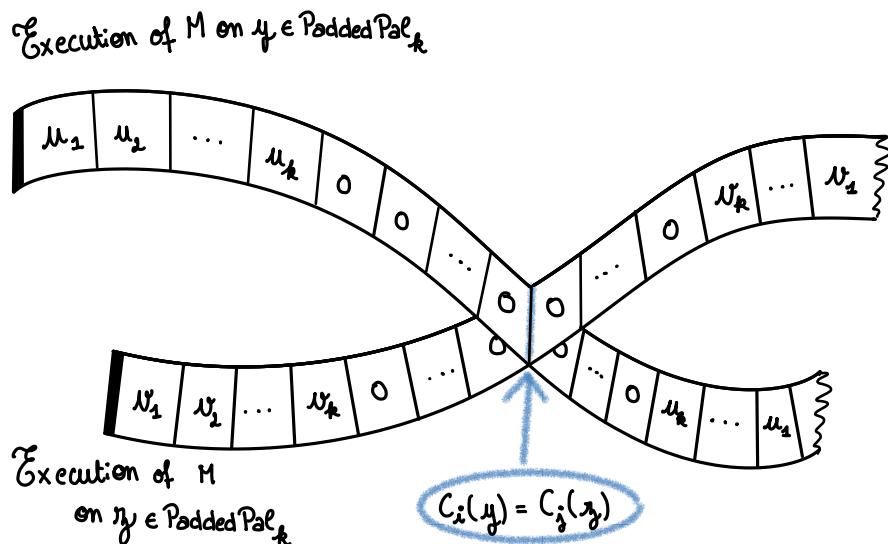


Figure 1 | Intersection of two crossing sequences

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 $y \in \text{PaddedPal}_k$, there is an $i \in \llbracket k+1, 2k \rrbracket$ 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 i s), 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 $i \in \llbracket k+1, 2k \rrbracket$ such that

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

Question 1.0.3. Conclude by observing that for each $y \in \text{PaddedPal}_k$, 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 $y \in \text{PaddedPal}_k$. 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 $y \in \text{PaddedPal}_k$.

We know that $|\text{PaddedPal}_k| = |\Sigma|^k$ 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

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

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 $\text{P} = \text{NP}$ then $\text{EXPTIME} = \text{NEXPTIME}$.^b

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

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

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 \square (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,

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

As $L' \in \text{P}$, 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 $\text{EXPTIME} = \bigcup_{k \in \mathbb{N}} \text{DTIME}(2^{n^k})$ and $\text{NEXPTIME} = \bigcup_{k \in \mathbb{N}} \text{NTIME}(2^{n^k})$.

^cWe 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, $L \in \text{EXPTIME}$.

We can conclude that $\text{NEXPTIME} \subseteq \text{EXPTIME}$.

2.2 Problem 2.

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

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

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

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

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

Thus we have that $A \in \text{NSPACE}(m^3)$ by the assumption that $\text{NSPACE}(m^4) \subseteq \text{NSPACE}(m^3)$. Also, the computation of $\lceil n^{5/4} \rceil - n$ can be done in space $O(n)$ (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 $O(n)$;
2. simulates P on \bar{w} in space $m^3 = O(n^{15/4}) = O(n^4)$,

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

We can conclude that

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

Question 2.2.2. Using Savitch's theorem and the previous result, show that $\text{NSPACE}(n^3) \subsetneq \text{NSPACE}(n^4)$ 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 $\text{NSPACE}(n^4) \subseteq \text{NSPACE}(n^3)$, then

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

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

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

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

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

Thus $\text{DSPACE}(n^6) = \text{DSPACE}(n^8)$ 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 $\text{NSPACE}(n^s) \subsetneq \text{NSPACE}(n^t)$ for all $0 \leq s < t, s, t \in \mathbb{N}$?

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

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

form a chain of equalities, we have that

$$\text{NSPACE}(n^{t+1}) = \text{NSPACE}(n^t).$$

Very similarly to the previous questions, we can show that

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

is in $\text{NSPACE}(m^{t+1})$ where M is a non-deterministic Turing machine that

uses at most $O(n^{t+1})$ space. Thus, we can show that $L(M) \in \text{NSPACE}(n^{t+1})$ as we can simulate M on \bar{w} using

$$O(m^{t+1}) = O((n^{(t+1)/(t+2)})^{t+1}) = O(n^{t+1})$$

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

$$\text{NSPACE}(n^t) = \text{NSPACE}(n^{t+1}) = \dots = \text{NSPACE}(n^{t+k}) = \dots, \quad (*)$$

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

We can conclude that

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

for all integers $s < t$.

Note. After (*), we could conclude that

$$\text{PSPACE} = \bigcup_{k \in \mathbb{N}} \text{DSPACE}(n^k) \subseteq \bigcup_{k \in \mathbb{N}} \text{NSPACE}(n^k) = \text{NSPACE}(n^t),$$

and thus we get the contradiction $\text{PSPACE} = \text{DSPACE}(n^t)$.