Theories of Computation: Summative Assignment 2

To be handed in on Canvas before **Thursday 31st March, 5pm GMT**

Exercise 1 *Let $\Sigma = \{a, b\}$ and $u, v \in \Sigma^+$, where Σ^+ is the set of* nonempty *words over Σ. We say that u is* present *in v if u can be obtained by deleting letters from v. For example, abbba is present in aabbababa. We write $|u|$ to denote the* length *of word u, i.e., the number of letters in u. For example, $|abbba| = 5$.*

The goal of this exercise is to show that the following decision problem is in NP.

> ***Input:*** *$w_0\#w_1\#...\#w_k$ such that $w_i \in \Sigma^+$ for $0 \leq i \leq k$.*
>
> ***Problem:*** *is there a word $x \in \Sigma^+$ such that $|x| = |w_0|$ and x is present in each w_i for $0 < i \leq k$?*

We will decompose the task into several steps.

1. *Let us consider a two-tape deterministic Turing machine $\mathcal{M}_1$ on the input alphabet $I = \{a, b, \#\}$ with initial state 0, tape alphabet $T = \{a, b, \#, ␣\}$, return values $V = \{\texttt{True}, \texttt{False}\}$, and whose transition function is represented as the diagram in Figure 1 below.*

   Initially,

   - *the* Main *tape contains a non-empty block of as and bs (representing a word $w \in \Sigma^+$) in between a $\#$ on the left, on which the head is positioned, and a $\#$ or a ␣ on the right. (Outside of these $\#$s and/or ␣s there could be any symbol in T.)*
   - *the* Aux *tape contains a non-empty block of as and bs (representing a word $x \in \Sigma^+$) and is otherwise blank, and the head is located on the ␣ immediately to the left.*

   For example,

Main	$\overset{\bullet}{\#}$	a	a	b	b	a	b	a	b	a	$\#$
Aux	␣	␣	$\overset{\bullet}{␣}$	a	b	b	b	a	␣	␣	␣

   In the case where the input block on the Main *tape forms the word $w = a^n$ and the input block on the* Aux *tape forms the word $x = a^m$ for $m, n > 0$, how many steps does the machine $\mathcal{M}_1$ goes through until it returns a value in V? (Returning counts as a step.)* **[3 marks]**

This is in fact the worst case complexity as a function of $n = |w|$ and $m = |x|$ and you can use this fact in the remainder of the exercise.

2. *Design a two-tape nondeterministic Turing machine $\mathcal{M}_2$ that takes as an input a word $w \in \Sigma^+$ and can generate any word $x \in \Sigma^+$ that has the same length as w.*

   Formally, the start configuration is:

   - *the* Main *tape contains a nonempty block of as and bs (representing a word $w \in \Sigma^+$), with a ␣ to the left on which the head is placed and a $\#$ to the right, the rest of the tape is blank to the left and can contain any symbol in T beyond $\#$ on the right;*
   - *the* Aux *tape is blank.*

   For example,

Main	$\overset{\bullet}{␣}$	a	a	b	b	a	b	a	b	a	$\#$
Aux	$\overset{\bullet}{␣}$	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣

   The machine $\mathcal{M}_2$ should stop when reaching a configuration where:

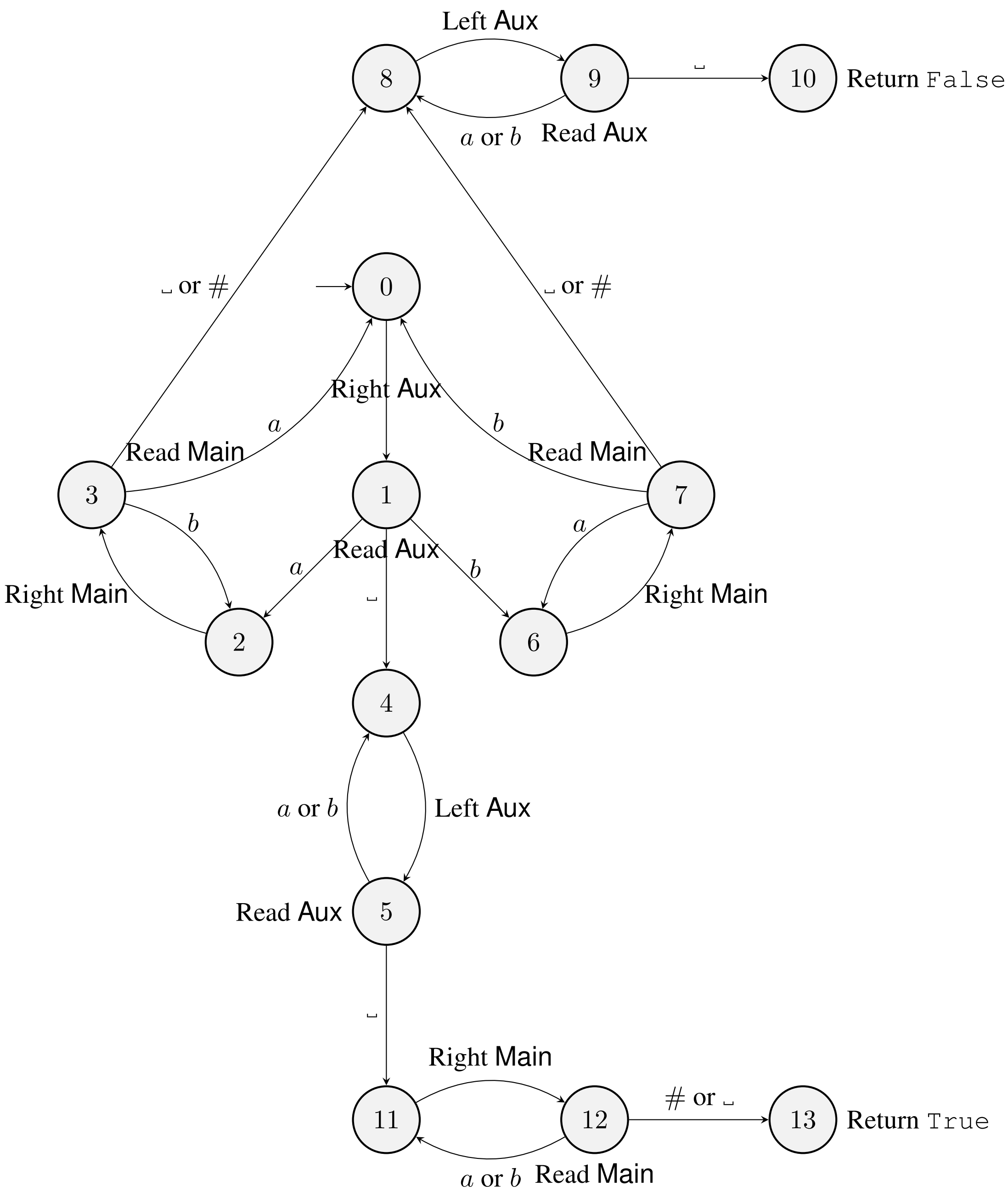


Figure 1: Transition diagram for machine $\mathcal{M}_1$

- *the Main tape is unchanged except for the head which should be placed on the $\#$ on the right*
- *the Aux tape contains an arbitrary block of as and bs (representing a word $x \in \Sigma^+$) of the same length as the input block on the Main tape and the head is on the first ␣ to the left.*

For example,

Main	␣	a	a	b	b	a	b	a	b	a	$\overset{\bullet}{\#}$
Aux	$\overset{\bullet}{␣}$	b	b	a	b	a	a	a	a	b	␣

Give the machine $\mathcal{M}_2$ and briefly explain your solution. (Do not use more than 10 states.) **[3 marks]**

3. *Using machines $\mathcal{M}_1$ and $\mathcal{M}_2$ as macros, design a two-tape nondeterministic Turing machine $\mathcal{M}_3$ for the above decision problem.*

   This means that the machine should start with

   - *a block of as, bs and $\#$s representing the input $w_0\#w_1\#...\#w_k$ on an otherwise blank Main tape with the head on the first ␣ to the left of w_0*

- *and a blank* Aux *tape.*

The tape contents and head positions at the end do not matter.

Give the machine $\mathcal{M}_3$ and briefly explain your solution. (Do not use more than 5 states.) **[3 marks]**

4. *Explain briefly why the problem is in* NP. **[3 marks]**

   *(**Note:** You may assume that any polytime two-tape nondeterministic Turing machine can be converted into a polytime one-tape nondeterministic machine with the same language. Just as we learnt in lectures for deterministic machines.)*

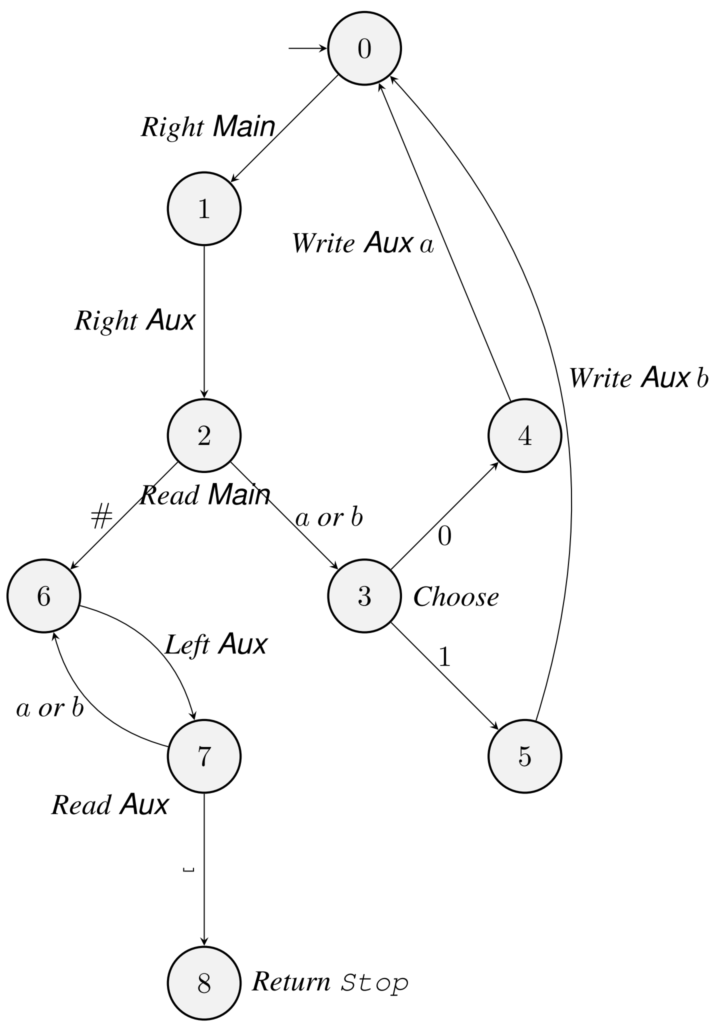
Solution 1 *Note that these solutions are more detailed than was required.*

1. *When $m \leq n$, the machine takes $4m + 2n + 7$ steps. When $m > n$, the machine takes $6n + 7$ steps.*

   Detailed explanation:

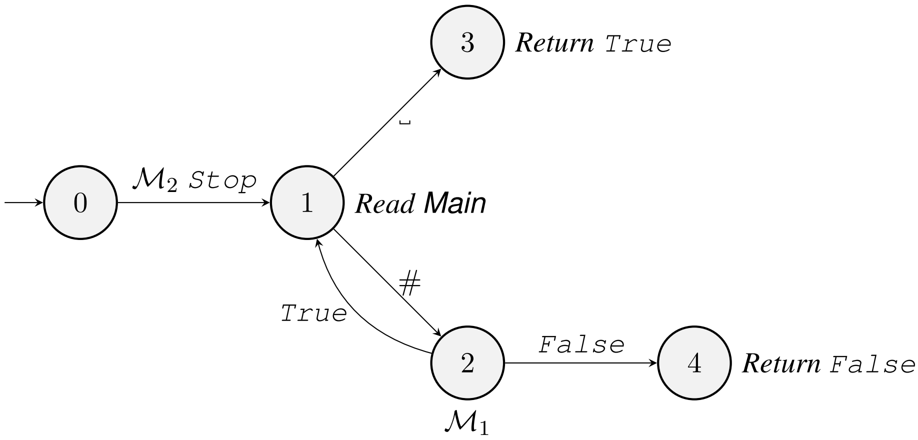
   - *If $m \leq n$: First, we go exactly m times through the (0-1-2-3-0) loop = $4m$ steps; until the last move right* Aux *gets us to a ␣ = 2 steps. At which point the auxiliary head is moved back to the ␣ to the left of the input block by going through it backwards via cycle (4-5-4) = $2m$ steps; until the last move left* Aux *gets us to a ␣ = 2 steps. Then, we finish reading up to the end of the block on the main tape via cycle (11-12-11) = $2(n-m)$ steps; before the last move right* Main *gets us to # or ␣ = 2 steps, and we return* `True` *= 1 step.*
     Total: $4m + 2 + 2m + 2 + 2(n - m) + 2 + 1 = 4m + 2n + 7$
   - *If $m > n$: First, we go exactly n times through the (0-1-2-3-0) loop = $4n$ steps; until the last move right* Main *gets us to a ␣ or # = 4 steps. At which point the auxiliary head is simply moved back on the ␣ to the left of the input block by going through it backwards = $2n$ steps; before the last move left* Aux *gets us to a ␣ = 2 steps and we return* `False` *= 1 step.*
     Total: $4n + 4 + 2n + 2 + 1 = 6n + 7$

2.



The machine uses the input on the main tape as a counter, going through each character until it reaches a $\#$ or a ␣. For each character that it reads on the main tape, it chooses nondeterministically to write an a or a b on the auxiliary tape. When it reaches the end of the input block on the main tape, it 'rewinds' the head on the auxiliary tape to the ␣ beginning the generated block of characters.

3.



Input: $w_0\#w_1\#...\#w_k$ such that $w_i \in \Sigma^+$ for $0 \leq i \leq k$ on Main tape with head on the ␣ to the left.

We start by running $\mathcal{M}_2$ on w_0 to generate a word x on the Aux tape (which is otherwise blank).

When it stops, it leaves the head on the ␣ (if $k = 0$) or the $\#$ (if $k \geq 1$) following w_0.

`For i` $= 1$ `to` k `repeat` $(*)$

We read the symbol under the head.

If it is ␣ we return `True`. If it is $\#$ we run $\mathcal{M}_1$ on word w_i and x.

If it returns `True`, then we go back to $(*)$.

If it returns `False`, then we return `False` as it means that x is not present in w_i.

4. Let us denote the size of the input by n which corresponds to $\sum_0^k |w_i| + (k-1)$ (when $k \geq 1$ or simply $|w_0|$ when $k = 0$).

Complexity of $\mathcal{M}_2$: $5|w_0| + 2 + 2|w_0| + 1 \leq 7n + 3$

Complexity of $\mathcal{M}_1$: In the worst case we will repeat the loop $(*)$ once for each word w_i for $1 \leq i \leq k$. That is, the total number of steps would be: $\sum_1^k (1 + 4|w_0| + 2|w_i| + 3) + 1 \leq (k-1)(6n+4) \leq 6n^2 + 4n$

Complexity of $\mathcal{M}_3$: $\leq 7n + 3 + 6n^2 + 4n + k$ (as we need to do the extra Read Main instruction k times) $\leq 6n^2 + 12n + 3$ steps

Since the machine $\mathcal{M}_3$ is polynomial time in the size n of the input, there is a polytime one-tape NDTM $\mathcal{M}_4$ with the same language.

A word of the given form has the required property iff it is possible for $\mathcal{M}_3$ to output some word in the first phase that passes all k tests in the second phase, i.e. if the word is acceptable to $\mathcal{M}_3$, or equivalently to $\mathcal{M}_4$. So the problem is in NP.