

This homework covers sections 4.6 and 5.1. It is due in class Monday, May 4. Hand in a hardcopy of your solutions.

*While you may discuss problems with other students, you should always make the first attempt on a problem yourself and **you must write up your own solutions in your own words**. You may not collaboratively write solutions or copy a solution that one person in the group writes up.*

1. Two grammars for $L = \{ a^n b^n c^n \mid n \in \mathbb{N} \}$ are shown below. (The one on the right was discussed on class.)

$S \rightarrow XTZ$	$S \rightarrow SABC$
$T \rightarrow AbCT$	$S \rightarrow X$
$T \rightarrow \epsilon$	$BA \rightarrow AB$
$bA \rightarrow Ab$	$CA \rightarrow AC$
$CA \rightarrow AC$	$CB \rightarrow BC$
$Cb \rightarrow bC$	$XA \rightarrow aX$
$XA \rightarrow aX$	$X \rightarrow Y$
$CZ \rightarrow Zc$	$YB \rightarrow bY$
$X \rightarrow \epsilon$	$Y \rightarrow Z$
$Z \rightarrow \epsilon$	$ZC \rightarrow cZ$
	$Z \rightarrow \epsilon$

- (a) Give a derivation for the string $aabbcc$ using the grammar on the left.
- (b) Explain how the grammar on the left works, and compare it to the grammar on the right. (See the posted derivations examples from 4/13 for a discussion of how the grammar on the right works.) You can explain how the grammar works by giving comments on the rules — identify what purpose they serve.

2. Two grammars for $L = \{ ww \mid w \in \{a, b\}^* \}$ are shown below. (The one on the right was partially discussed on class.)

$S \longrightarrow HTE$	$S \longrightarrow RT$
$T \longrightarrow aAT$	$R \longrightarrow aRa$
$T \longrightarrow bBT$	$R \longrightarrow bRb$
$Aa \longrightarrow aA$	$R \longrightarrow D$
$Ab \longrightarrow bA$	$D \longrightarrow DP$
$Ba \longrightarrow aB$	$Paa \longrightarrow aPa$
$Bb \longrightarrow bB$	$Pab \longrightarrow bPa$
$AE \longrightarrow Ea$	$Pba \longrightarrow aPb$
$BE \longrightarrow Eb$	$Pbb \longrightarrow bPb$
$Ha \longrightarrow aH$	$PaT \longrightarrow Ta$
$Hb \longrightarrow bH$	$PbT \longrightarrow Tb$
$HE \longrightarrow \epsilon$	$DT \longrightarrow \epsilon$
$T \longrightarrow \epsilon$	

- (a) Give a derivation for the string $aabbaabb$ using the grammar on the left.
- (b) Explain how the grammar on the left works, and compare it to the grammar on the right. (See the posted creating a grammar examples from 4/13 for a discussion of how the grammar on the right works.) You can explain how the grammar works by giving comments on the rules — identify what purpose they serve.
3. Create a general grammar for the language $L = \{ a^n b a^n b a^n \mid n \in \mathbb{N} \}$ and explain how the grammar works. You can explain how the grammar works by giving comments on the rules — identify what purpose they serve.
4. Create a general grammar for the language $L = \{ a^n b^m b c^{nm} \mid n, m \in \mathbb{N} \}$ and explain how the grammar works. You can explain how the grammar works by giving comments on the rules — identify what purpose they serve.

Use the Turing Machine web app at <https://math.hws.edu/eck/js/turing-machine/TM.html> for the following exercises. Be sure to review the instructions at <https://math.hws.edu/eck/js/turing-machine/TM-info.html> (or click on the link at the top of the simulator page).

Name your Turing machines TM5, TM6, and TM7 according to the problem and fill in the description with a brief description of what the machine does. If you don't do this when you create a new Turing machine, click "Show Edit/Import/Export", then "Grab Current Example". Edit the name and description and click "Apply".

Save the current Turing machine by clicking the "Save File" button. This will create a `.json` file in your browser's Downloads folder.

Hand in your solutions electronically by uploading the JSON files to Canvas — look for "HW 10 Turing machine handin" in the Assignments section. Hand in all of your files at once — subsequent submissions overwrite previous ones rather than adding files.

5. Create a Turing machine which decides the language

$$L = \{w \in \{a, b\}^* \mid n_a(w) \text{ is a multiple of } 3 \}$$

The input is a string of *as* and *bs* with the machine positioned at the left end of the string. The output of the computation should be 1 if the number of *as* is a multiple of 3 and 0 otherwise. Note that the *only* thing on the tape at the end should be 0 or 1, and the machine should be positioned on that value.

Hint: Consider a DFA that accepts this language; the Turing machine will have states that fulfill a similar role.

6. Create a Turing machine which accepts the language

$$L = \{a^n b^m \mid n \geq m \}$$

The input *w* is a string of *n as* followed by *m bs* with the machine positioned at the left end of the string. The machine should halt if and only if $w \in L$.

Hint: there are several ways to create a machine that doesn't halt. The simplest is to use S ("stay") instead of L or R, but the machine can also just go left (or right) forever or alternate left and right moves (requires two states).

7. Create a Turing machine which converts unary numbers to binary. The input is a string of *as* with the machine positioned at the left end of the string. The output of the computation should be the binary number equal to the original number of *as*. Note that the *only* thing on the tape at the end should be the binary number, and the machine should be positioned at the left end of the number.

For a complete answer, your machine should also work for empty input — write a 0 in that case.

Hint: Create the binary number to the left of the string of as . Start by writing a 0. Then repeatedly erase the rightmost a , incrementing the binary number each time. Note that adding 1 to a binary number amounts to finding the rightmost 0 in the number, changing it to a 1, then changing all of the 1s to the right of that new 1 to 0. If there is no rightmost 0, add a new 1 on the left and do the same thing (change all of the 1s to the right of that new 1 to 0). For example:

$$\begin{aligned} 0 &\rightarrow 1 \\ 1 &\rightarrow 10 \\ 10 &\rightarrow 11 \\ 11 &\rightarrow 100 \\ 100 &\rightarrow 101 \\ 10111 &\rightarrow 11000 \end{aligned}$$