# CS143 Midterm Spring 2021 

- Please read all instructions (including these) carefully.
- There are 5 questions on the exam, some with multiple parts. You have 95 minutes to both finish the exam and upload it. After 95 minutes, gradescope will close your exam and automatically submit it, so make sure submit what you have by then.
- The exam is open note. You may use laptops, phones, e-readers, and the internet, but you may not consult anyone.
- You must upload the answer to each question as an image file or a PDF. You may submit images of hand-written answers taken with your phone (but allow yourself time to send the image to your computer, so you can upload it to gradescope). It is your responsibility, however, to ensure they are legible. Computer typed answers as a PDF are also permitted.
- Solutions will be graded on correctness and clarity. Each problem has a relatively simple and straightforward solution. You may get as few as 0 points for a question if your solution is far more complicated than necessary. Partial solutions will be graded for partial credit.

| Problem | Max points | Points |
| :---: | :---: | :---: |
| 1 | 15 |  |
| 2 | 15 |  |
| 3 | 20 |  |
| 4 | 20 |  |
| 5 | 30 |  |
| TOTAL | 100 |  |

In accordance with both the letter and spirit of the Honor Code, I have neither given nor received assistance on this examination.

Type your name as a signature below:

## 1. Sizes of NFAs and DFAs

A regular expression can be converted to both an NFA and a DFA. (Remember that NFAs and DFAs must have a starting state and accepting states and that each state of a DFA must have an outgoing transition for every symbol in the alphabet.) Given the alphabet $\{0,1\}$, provide

- a regular expression,
- a corresponding NFA, and
- a corresponding DFA
where:
(a) The NFA has fewer states than the DFA.


## Answer:

(b) The NFA has the same number of states as the DFA.

Answer:
(c) The NFA has more states than the DFA.

Answer:

## 2. CFG Transformations

Left factor and remove immediate left recursion in the following grammar:

$$
\begin{aligned}
S & \rightarrow c A|c B e| c B d|S c| S A \\
A & \rightarrow e \\
B & \rightarrow f
\end{aligned}
$$

## Answer:

## 3. Syntax-Directed Translation

Given the following grammar, add semantic actions that computes the number of sequences of successive $a$ 's. For example, both $a c a b$ and $a a c a b b$ have two sequences of $a$ 's each, while abacabcb have three such sequences.
You must use the following attributes: int val, bool left, and bool right. Use bison syntax: \$i.val refers to the val attribute of the $i$ th symbol of the production and $\$ \$$.val refers to the val attribute of the production's result. You may not use global variables.

$$
\begin{array}{ccl}
S & \rightarrow & T S \\
& \mid & \epsilon \\
T & \rightarrow & a T b \\
& \mid & b T c \\
& & c T a \\
& \mid & \epsilon
\end{array}
$$

## Answer:

## 4. Top-Down Parsers

In this question we will explore top-down parsers and $\mathrm{LL}(\mathrm{k})$ grammars.
(a) Construct the $\mathrm{LL}(1)$ top-down parser table for the following grammar.

$$
\begin{array}{rll}
S & \rightarrow & a S A \\
& \mid & \epsilon \\
A \rightarrow & \rightarrow b S \\
& \mid & c
\end{array}
$$

## Answer:

(b) Show that the following grammar is not LL(1) by pointing out where a conflict occurs when constructing its $\mathrm{LL}(1)$ top-down parser table.

$$
\begin{aligned}
S & \rightarrow a S A \\
& \mid \epsilon \\
A \rightarrow & \rightarrow a b S \\
& \mid c
\end{aligned}
$$

## Answer:

## 5. Bottom-Up Parsers

Consider the following grammar:

$$
\begin{aligned}
& S \rightarrow A e|A B| d B \\
& A \rightarrow d \mid f \\
& B \rightarrow g
\end{aligned}
$$

where the terminals are $\{d, e, f, g\}$.
(a) List the first and follow sets of the non-terminals.

Answer:
(b) Draw the LR(0) DFA. (In this question, you need only provide accepting states, including their items, and transitions between accepting states.)
Answer:
(c) Is the grammar SLR(1)? Justify your answer.

Answer:

