CS 3100, Fall 2020

OPEN-BOOK Midterm-2 Exam

Given Saturday 10/31/20, 9am

Due Monday 11/02/20, 11:59pm

- The Exam has a Quiz part worth 25% and a Notebook part worth 75%
- The Quiz part is offered as a Canvas Quiz called MIDTERM-2 EXAM's Quiz Component. Canvas lists it as having 30 points (15 questions, 2 points each, 2 correct answers per question, 2 attempts of 60 mins each). This will be scaled to 25% of the MT-2 points.
- The notebook part is offered via Github in the directory **MIDTERM-2** as a single file called **u1234567-MT2-Notebook.ipynb** (`https://github.com/ganeshutah/Jove/tree/master/For_CS3100_Fall2020/MIDTERM-2`).
- Finish this notebook, upload it via **Assignment Midterm-2 Notebook Upload**. This upload portal is open from 10/31/20 9am till 11/02/20 11:59pm.

Questions in the Notebook, Expected Answers

Answers are expected around or after arrows I've placed in the notebook (for easy spotting). Points add up to 100% that will be scaled to 75%. Use this as a checklist for answering.

- Q1: Regular versus non-regular
  - Part Q1.1 (15%):
    - Mention which language is regular
    - Provide an RE for it
    - Display a minimal DFA for the RE using the provided code
  - Part Q1.2 (15%):
    - Mention which language is non-regular
    - Provide a PL proof thereof
- Q2: DFA minimization questions
  - Part Q2.1 (15%): Justify why a state-pair is 2-distinguishable
    - show what is 1-distinguishable
    - show what is 0-distinguishable
  - Part Q2.2 (10%): Given equivalence classes, merge states
    - Provide merged state-name 1
    - Provide merged state-name 2
    - ...more such entries (as many as needed, if needed)...

- Q3: CFG-related questions
  - Part Q3.1 (15%): Find one string w starting with a that can't be generated
    - Justify why w can't be generated, in a few clear steps
  - Part Q3.2 (15%): Implement a PDA for the CFG.
    - Show w isn't accepted
    - Present four strings starting with a that are accepted
    - Run and show using the PDA
- Q4: CFG design
  - Only one part: (15%): Design a CFG. Convert to a PDA. Run on given strings.