CS4120/4121/5120/5121—Spring 2018
Homework 1
Lexical Analysis
Due: Thursday, February 1, 11:59PM

0 Updates

- None yet; watch this space.

1 Instructions

1.1 Partners

You may discuss the homework with other students but you must acknowledge the contributions of others. Remember the course staff is happy to help with problems you run into. Use Piazza for questions, attend office hours, or set up meetings with any course staff member for help.

1.2 Homework structure

There are two parts of the homework. The first part is required of all students. The second part is required of students taking CS5120, but those enrolled in CS4120 are welcome to try it for good KARMA.

1.3 Tips

You may find the Dot and Graphviz packages helpful for generating drawings of DFAs and other graphs. You can get these packages for multiple OSes from the Graphviz download page. An example of a DFA drawn using Graphviz may be useful.

2 Problems

1. Design of finite automata

Living cells on earth encode their genetic information in a chemical code made from repetitions of four basic compounds abbreviated as `A`, `C`, `G`, and `U`. These constituents are arranged in long linear sequences of triplets, e.g., `GCU` and `UGA`, referred to as *codons*.

The meaning of a short subsequence of a genetic encoding is often not immediately clear, since there are different *reading frames* in which to interpret the information. For instance, the partial sequence `CUUCUCCGCAAUUUAC` might specify any of the following codons, where question marks indicate missing data:

- `CUU CUC CGC AAU UUA C??`
- `?CU UCU CCG CAA UUU AC?`

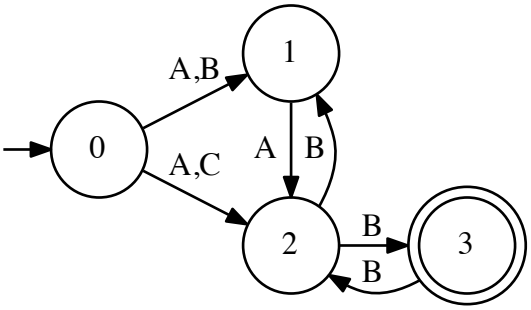
- `??C UUC UCC GCA AUU UAC`

Certain codons have special meanings that resolve the reading frame ambiguity. The *start codon* `GUG` denotes the beginning of a gene-encoding region. One of the three *termination codons* `UAG`, `UAA`, and `UGA` marks the end of such a region.

A commonly asked question is whether a partial sequence acquired from a biological sample contains a complete gene-coding region under some reading frame. Design a nondeterministic finite automaton that accepts such a subsequence. Ensure that the start and termination codons are correctly aligned.

2. Determinizing an automaton

Construct a deterministic version of the following nondeterministic finite automaton. Make sure to indicate the initial and terminal states. Label each DFA state with the set of NFA states to which it corresponds.



3. Simulation

Use the construction scheme given in class to build an NFA for the following regular expression:

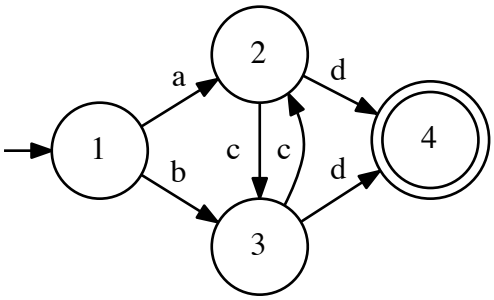
```
(abc*)* | abc
```

Simulate your constructed NFA on the input string "`abcccabc`" and show the set of states reachable at each step of input processing.

3 Problem for CS5120

4. DFA simplification

Simplify the following DFA using the method presented in class. Show the minimized version, as well as any intermediate steps you took.



4 Submission

Submit your solution as a PDF file on CMS. This file should contain your name, your NetID, all known issues you have with your solution, and the names of anyone you have discussed the homework with.