Midterm

Wednesday, 4 March 2020
9:10–10:00 AM

There are 5 problems each worth 6 points for a total of 30 points. Show all your work, partial credit will be awarded. Space is provided on the test for your work; if you use a blue book for additional workspace, sign it and return it with the test. No notes, no collaboration.

Name: ______________________________

Problem	Credit
1	
2	
3	
4	
5	
Total	

1. Give an NFA that accepts exactly the strings over the alphabet $\{0, 1\}$ such that the number of 01 substrings equals the number of 10 substrings. (Exactly means, those string and only those strings. The empty string happens to be such a string, by the way.)

   Next, give a machine with the fewest number of states. *Do not worry if you believe your first answer had the minimum number of states. This is just a problem to come back to later, to see if you can improve your otherwise correct solution.*

   Scoring

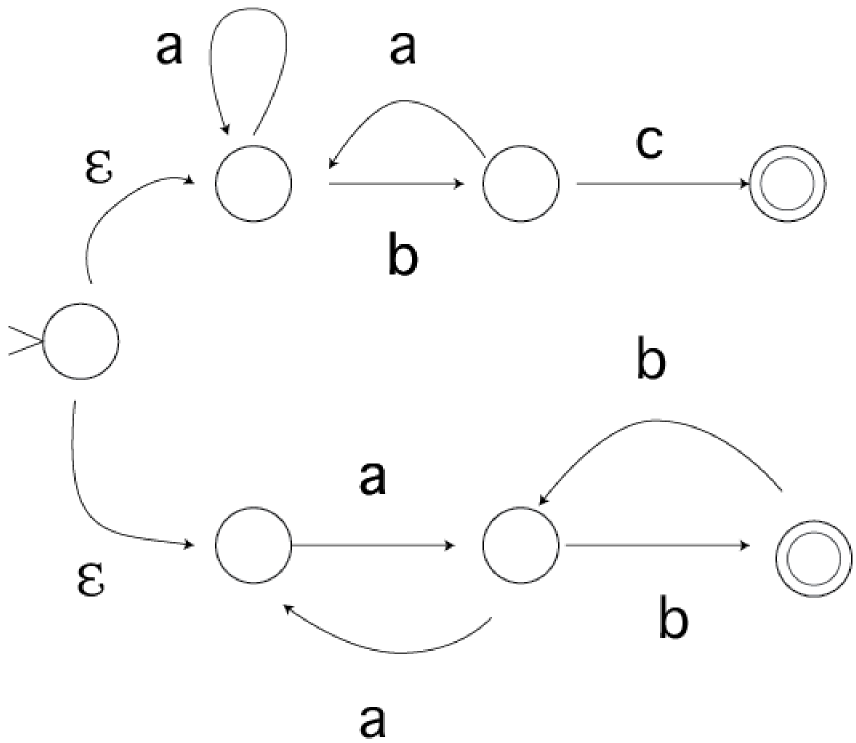
   One point and stop: if in essence the solution accepts $\{0, 1\}^*$, $\emptyset$, or just a small finite set of strings.

   (a) Accepts ϵ,
   (b) Accepts 0^+ and 1^+,
   (c) Accepts any string that begins and ends on x with at least one $1 - x$ between, $\forall x$.
   (d) Rejects all strings that begin and end on different values.
   (e) Correctly accepts or rejects a custom challenge string.

   If passes above, $4/x$ points if solution has x states:

```
start: S
final: S
    R
state: S
    0 A
    1 B
    0 R
    1 R
state: A
    0 A
    1 A
    0 R
state: B
    0 B
    1 B
    1 R
```

2. Write a Regular Expression that expresses the same language as the following FSA.



Scoring

- Top path and bottom path scored individually, with 3 points each.
- Generally speaking, a point if some correct string is in the set generated by the R.E.
- Correct is 3 points, with the following list of correct answers.
  - top path:
    - $a^*b(a^+b)^*c$,
    - or $a^*(ba^+)^*bc$,
    - or $(a \mid ba)^*bc$,
    - or $a^*(ba^+)^*bc$.

- bottom path:
  - $a(aa)^*b(b(aa)^*b)^*$,
  - or $a(aa \mid bb)^*b$.

- Rarely 2 points, for an almost correct solution.

Note that many bottom paths are subsets of a^*b^*. While this accepts many correct strings, it does not accept any string such as $abbaab$.

3. Show that the language

$$\{\, a^i \# b^j \# c^k \# d^n \mid \text{ where } i,j,k \geq 0 \text{ and } i+j+k=n \,\}$$

is not regular.

Discussion

The use of the PL to show $\mathcal{A}$ not Regular being: *there exists* a collection of increasing length strings $\{\, s_i \mid s_i \in \mathcal{A}, |s_i| = p_i\}$ such that *for all* dissections $s_i = xyz$ of s_i satisfying "certain" restrictions, *there exists* and i which pumps the string out of $\mathcal{A}$, $xy^iz \notin \mathcal{A}$.

(a) Make sure the Prosecutor is an "exists" player. That is, s is sufficiently specfified.

(b) Make sure the Advocate is a "for all player". That is, no lazy Advocates. All possible y must be considered hoping to get an acquittal.

(c) Contempt of court: y must be properly from s, and the conclusion given the choice if i must be properly stated.

The best answer has $s = a^p \# b \# c \# d^{p+2}$ or similar, because the Prosecutor wants to tie the Advocates hands. A good lawyer does not ask a witness a question unless she already knows the answer. In this case, she knows it will be $y = a^k$, and her counter argument, $i = 0$ (for instance) is ready.

In considering what it means for the Advocate to be the "for all player", consider these languages,

$$\{\, a^i b^j c^k \mid \text{ where } i,j,k \geq 0 \text{ and } i+j \geq k \,\}$$

or

$$\{\, a^i b^j c^k \mid \text{ where } i,j,k \geq 0 \text{ and } k \geq \max(i,j) \,\}$$

The danger is that the Prosecutor selects $s_i = ab^{p_i}c^{p_i}$, and the Advocate responds $y = a$.

For the first example, the Prosecutor must select something like, $s_i = a^{p_i}bc^{p_i+1}$, and for the second example, the Prosecutor must selection something like $s_i = a^{p_i}bc^{p_i}$. Then the Advocate has no choice but $y = a^+$, and the Prosecutor wins by pumping down.

Scoring

Generally,

(a) Full credit for a correct answer.

(b) For a lazy Advocate, 4 points.

(c) Similar to 4 points, but an error in the Prosecutor's string.

(d) For a problem with the math, such that what is stated does not make sense, 2 points.

(e) No credible work, 0 points.

4. Give a Context Free Grammar for the language,

$$\{\, a^i b^j c^k \mid i = j \text{ or } i = k \,\}$$

Then show that the CFG is ambiguous by giving two parse trees in you grammar of the string $aaabbbccc$.

Scoring

$$\begin{array}{rcl} S & \longrightarrow & J\,C \mid K \\ C & \longrightarrow & \epsilon \mid c\,C \\ J & \longrightarrow & \epsilon \mid a\,J\,b \\ K & \longrightarrow & B \mid a\,K\,c \\ B & \longrightarrow & \epsilon \mid b\,B \end{array}$$

- One point for some CFG like syntax, but either results in a finite language or uses variables such as an expression $a^i b^i$, for an integer i.
- For three points, the CFG must express $\mathcal{L} \subset a^* b^* c^*$ yet not be $a^* b^* c^*$ or similar.
- Above that solutions were fully correct for 6 points, or had a deficiency for 5 points.

5. Give a Context Free Grammar for the Regular Expression:

$$ab^*(a|b)(c(a|b))^*$$

Give a Regular Expression for the following Context Free Grammar, or give a proof or a concise logical argument why an equivalent Regular Expression does not exist,

$$\begin{aligned} S &\longrightarrow AX \\ A &\longrightarrow aA \,|\, a \\ X &\longrightarrow \epsilon \,|\, abXc \end{aligned}$$

Solution

$$\begin{aligned} S &\longrightarrow a\,BXY \\ B &\longrightarrow \epsilon \,|\, b\,B \\ X &\longrightarrow a \,|\, b \\ Y &\longrightarrow \epsilon \,|\, c\,XY \end{aligned}$$

or

$$\begin{aligned} S &\longrightarrow aX \\ X &\longrightarrow b\,X \,|\, YZ \\ Y &\longrightarrow a \,|\, b \\ Z &\longrightarrow \epsilon \,|\, c\,YZ \end{aligned}$$

The CFG cannot have an equivalent RE, as the CFG is not regular.

It the CFG, call it $\mathcal{A}$, were regular, so would be $\mathcal{A} \cap \big((a\,|\,b)^* c^*\big)$. However each string in the family $a(ab)^p c^p$ of strings in that intersection language can be pumped out.

Because y in the decomposition cannot contain any c's, by size restriction on xy, and pumping y will imbalance either the number of a's or the number of b's (or both) compared to the number of c's.

Scoring

(a) The first problem was fully right, 6 points.

(b) 5 points that were close to fully right.

(c) From there there was a gap where solutions were wrong but on the right track. These were 3 points.

(d) Then 2 or 1 points for grammars that had substantial and obvious errors, but were some sort of CFG. Many suggested grammars here were actually finite.

The second problem did not contribute any points, as it did not suggest additional degrees of correctness to the overall question.

One answer attempted a proof using the false theorem that for L_1 regular and L_2 non-regular but CFL, then $L_1 \circ L_2$ is not regular. However, $L_1 = \{\, a^i b^j \,|\, i \geq j \,\}$ and $L_2 = b^*$ gives a counter-example.

Another counter-example is $L_1 = \{\, a^i \,|\, i \text{ a square} \,\}$ and $L_2 = a^*$.