CS5003 Final Exam

Foundations of C.S.

Summer, 2023
Good Luck!

You may use your textbook or anything appearing on our canvas pages. You may not collaborate or use other sources..

Do any five of the following eight problems. Write your answers clearly and neatly. Upload a pdf of your answers. It should be a single file, but may have several pages.

---

1. Let $\Sigma = \{a, b\}$ and let
$X = \{w \in \Sigma^* \mid n_a(w) + n_b(w) = 2k; k \in \mathbb{N}\}$, and
$Y = \{uv \mid u \in \Sigma^*; v \in \Sigma^*; \text{length}(u) = \text{length}(v)\}$ define two languages.
Prove using the double inclusion method that $X = Y$.
[It is important the your argument is complete, uses the double inclusion method, and also that you not make it any more complicated than necessary.]

2. Let $\Sigma = \{a, b\}$ and let the language $L \subseteq \Sigma^*$ be defined recursively by
BASIS: $a^3 \subseteq L$, and $b^3 \subseteq L$, and
RECURSIVE STEP: If $w = uaav \in L$, then $ubaaabv \in L$ and $uaabaav \in L$.

If $w = ubbv \in L$, then $ubbabbv \in L$ and $uabbbav \in L$.

CLOSURE: All elements of S are obtained from the basis after a finite number of applications of the recursive step.

a) List all the elements in S_0 and all the elements of S_1.

b) Prove carefully by induction that every string $w \in L$ has either aaa or bbb as a substring, that is, w can be written either as $w = paaaq$, or $w = pbbbq$ for some $p, q \in \Sigma^*$.

[Your proof must be an induction proof, and must be clear and carefully written.]

3. Let L begin the set of strings on $\Sigma = \{a, b, c\}$ which do contain abc but which do not contain bbb as substrings.

Design a regular expression for the language L.

Make sure that your expression matches every string in L and does not match any string not in the language.

Your design principles should make your expression readable and understandable.

4. Let $\Sigma = \{a, b\}$. Consider the grammar

$$\begin{aligned} G : S &\rightarrow A \mid B \mid C \\ A &\rightarrow aaB \mid aaC \mid aa \\ B &\rightarrow bbA \mid bbC \mid bb \\ C &\rightarrow ccA \mid ccB \mid cc \end{aligned}$$

a) Draw a derivation tree for $ccbbaa \in L(G)$.

b) Describe the language of this grammar in words, or set theoretically.

c) Design a regular expression for the language of the grammar. (It is much more important that the expression is correct and well designed than being short.)

5. Given the grammar

$$\begin{array}{rcl} G: S & \rightarrow & C \mid D \mid E \mid \lambda \mid ABCDEFGH \\ A & \rightarrow & aAA \mid AaA \mid AAa \\ B & \rightarrow & b \mid bb \mid bbb \mid F \\ C & \rightarrow & aD \mid \lambda \\ D & \rightarrow & bE \mid E \\ E & \rightarrow & cC \mid C \\ F & \rightarrow & b \mid bb \mid bbb \mid B \\ G & \rightarrow & aaa \mid H \\ H & \rightarrow & bbb \mid \lambda \end{array}$$

Compute REACH and TERM and use them to construct a new grammar G' with no useless symbols.

Compute NULL and CHAIN on G' to convert to and equivalent G'', and essentially non-contracting grammar.

6. a) Suppose you have a grammar with the following rules. Find equivalent rules with no left recursion.

$$\begin{aligned} A &\rightarrow a \mid aA \mid Aa \mid BaA \mid AaB \\ B &\rightarrow bbb \mid BBB \\ C &\rightarrow bbb \mid BBB \mid ccc \mid CCC \mid cba \mid CBA \end{aligned}$$

If you do not use the methods presented in the text or lectures, then you must justify your procedure.

b) Convert to the following grammar to an equivalent grammar in Chomsky normal Form

$$\begin{aligned} G: S &\rightarrow aa \mid P \\ P &\rightarrow aa \mid aQaQ \\ Q &\rightarrow bb \mid bPbP \\ R &\rightarrow cc \mid bPQPRPQb \mid aPRQRaPaQaR \mid abc \end{aligned}$$

If you do not use the methods presented in the text or lectures, then you must justify your procedure.

7. Design a Deterministic Finite Automaton for the language of all strings on {a,b,c} which contain the substring *aa* or the substring *bb* but not both.

Your design should be clear and explained.

8. Here is a graph of an automaton. Find a regular expression for the language, or explain why the regular expression cannot exist.