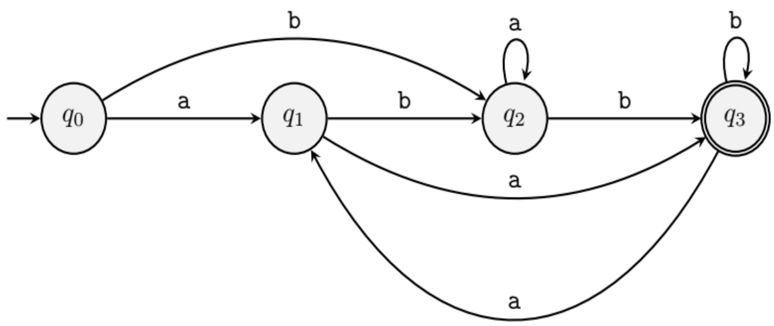
Final Exam
Total marks: 45
Duration: 100 minutes

CSE331

Solve problems 1 through 6. Problem 7 is optional.

Problem 1: Converting DFA to Regular Expressions (10 points)

Convert the following DFA into an equivalent regular expression using the state elimination method. First eliminate q_1, then q_2 and so on. You must show work.



Problem 2: Designing Context-Free Grammars (5 points)

Solve the following problems.

(a) Give a context-free grammar for the following language. (2 points)

$$A = \{w \in \{0,1\}^* : w \text{ contains at least two 0s}\}$$

(b) Give a context-free grammar for the following language. (3 points)

$$L = \{w \in \{0,1\}^* : w = 0^{3i} v 1^{2i} \text{ where } v \in A \text{ and } i \geq 0\}$$

Problem 3 : Derivations, Parse Trees, and Ambiguity (5 points)

$$S \to 0S1 \mid 1S0 \mid SS \mid 01 \mid 10$$

Take a look at the grammar above and solve the following problems.

(a) Show that the grammar above is ambiguous by demonstrating two different parse trees for 011010. (4 points)

(b) Find a string w of length six such that w has exactly one parse tree in the grammar above. (1 point)

Problem 4: Chomsky Normal Form (10 points)

Convert the following grammar into Chomsky Normal Form. You must show work.

$$\begin{aligned} S &\to \mathtt{a} \mid \mathtt{a}A \mid B \\ A &\to \mathtt{a}BB \mid \varepsilon \mid D \\ B &\to A\mathtt{a} \mid \mathtt{b} \\ D &\to \varepsilon \end{aligned}$$

Here a, and b are terminals and the rest are variables.

Problem 5: The CYK Algorithm (10 points)

Use the CYK algorithm to determine whether the string `abcccc` can be derived in the following grammar. You must show the entire CYK table.

$$
\begin{aligned}
S &\to AB \\
A &\to CD \mid CF \\
B &\to \texttt{c} \mid EB \\
C &\to \texttt{a} \\
D &\to \texttt{b} \\
E &\to \texttt{c} \\
F &\to AD
\end{aligned}
$$

Here, `a`, `b`, and `c` are terminals and the rest are variables.

Problem 6: Constructing Pushdown Automata (5 points)

Construct a pushdown automaton for the following language.

$$L = \{w\overline{w}^{\mathcal{R}} : w \in \{0,1\}^*\}$$

Here, $\overline{w}^{\mathcal{R}}$ denotes the reverse complement of the string w.

For example, $01001101 \in L$ because the second half of the string, 1101, is the reverse complement of the first half, 0100, i.e. $1101 = \overline{0100}^{\mathcal{R}}$.

Problem 7: (Bonus) Deja Vu (5 points)

(Note that this is a bonus problem. Attempt it only after you are done with everything else. Even if you do not attempt it, you can get a perfect score. So, do not worry if you find it too hard!)

Take a look at the following languages. They should seem familiar.

$$L_1 = \{w \in \{0,1\}^* : w = 0^m1^n \text{ where } m \text{ and } n \text{ are both even}\},$$

$$L_2 = \{w \in \{0,1\}^* : w = 0^m1^n \text{ where } m \text{ and } n \text{ are both odd}\}.$$

We will use these two languages to define the following language.

$$L = \{w \in \{0,1\}^* : w = uv, \text{ where } u \in L_1,\ v \in L_2, \text{ and } |u| = |v|\}$$

(a) Write down a string $w \in L$ such that w is of length eight. (0 point)

(b) Construct a pushdown automaton that recognizes L. Describe what your automaton is doing in two or three sentences. (5 points)