

11.1 Automata and regular languages

Name: _____ Class: _____ Date: _____

Total: 24 marks

KEY WORDS deterministic finite automaton

• pumping lemma

Objective

Trace DFAs exactly and use the pumping lemma as a ceiling.

1 Key knowledge and worked examples

■ The machine

States plus a transition table; acceptance only at end of input ✓. The ends-in-ab DFA: aabab accepts, aba rejects ✓.

■ Worked example: the ceiling

Pumping: any long accepted string has a pumpable y ; $a^n b^n$ breaks when pumped – not regular ✓.

2 Practice

2.1 In the ends-in-ab DFA, which state accepts? [3]

2.2 Why is aba rejected? [3]

2.3 One line: what does the pumping lemma forbid a regular language from doing? [3]

3 Exam-style questions

3.1 The ends-in-ab DFA over the alphabet a, b has these transitions: q0 on a goes to q1; q1 on a stays q1; q1 on b goes to q2; q2 on a goes to q1; q0 on b stays q0; q2 on b goes to q0. The accepting state is q2.

Give the full state trace of **babbab** and the verdict. Give the trace of **babba**. State which two positions of the input could be swapped to flip the verdict, and why. Draw (or describe) the missing 8th transition a complete table needs if any input letter were added. [8]

3.2 Use the pumping lemma to argue $L = \{a^n b^n\}$ is not regular with p as the pumping length.

Choose the string and state its constraint on n . Split it xyz and state what y must contain. Pump once and derive the contradiction. State what this ceiling means for a lexer that must match nested brackets. [7]

Solutions

2.1

- q_2 ✓.

2.2

- It ends in state q_1 – the final **a** leaves the pattern incomplete ✓.

2.3

- It forbids unbounded counting: every long string has a pumpable loop ✓.

3.1

- **babbab**: $q_0 \rightarrow q_0$ (b), $q_0 \rightarrow q_1$ (a), $q_1 \rightarrow q_2$ (b), $q_2 \rightarrow q_0$ (b), $q_0 \rightarrow q_1$ (a), $q_1 \rightarrow q_2$ (b) – accepted ✓.
- **babba**: q_0 , q_0 , q_1 , q_2 , q_1 , q_1 – rejected ✓.
- Swapping the last two letters flips it: **babbab** keeps ...ab; **babba**... variants end *ba* or *aa* – only strings ending *ab* accept, so the final pair decides ✓.
- A complete table over $\{a, b, c\}$ needs a dead state q_d : q_d loops on every letter, entered on *c* from anywhere ✓.

3.2

- Choose $a^p b^p$ ($n = p$ forced by the lemma's threshold) ✓.
- $|xy| \leq p$ keeps y inside the *a*-block: $y = a^k$, $k \geq 1$ ✓.
- Pump once: $xy^2z = a^{p+k}b^p$ – more *a*'s than *b*'s, not in L ✓.
- Ceiling for lexers: nested brackets need counting, so no regex (DFA) lexer can match them – the parser above must take over ✓.