

CSC173 FLAT Module Exam With FFQ Comments

Chris Brown

Sept 24, 2009

Please write your name on the bluebook. This is a closed-book Exam. There are 75 possible points (one per minute). Best not to spend more minutes on a question than it's worth. Stay cool and please write neatly.

1 Regular Expression to Minimized DFA (30 min.)

In Pascal and other languages, an identifier must be “a letter followed by zero or more letters or digits”, so A and ABC and Z92A are fine but 12 and 5MYVAR are not.

1. (7 min.) Using “l” (lower case 'ell') for letter and “d” (lower case 'dee') for digit, give the regular expression defining legal identifiers meeting the above rule. (Hint: Don't get this wrong!).
2. (8 min.) Create an equivalent nondeterministic finite automaton by our “mini-machine” method (Thompson's method).
3. (8 min.) Use the subset construction to convert the NFA to a DFA.
4. (7 min.) Use the equivalence class argument to minimize the DFA.

People did fine on this.

2 A Promotion Opportunity? (10 min.)

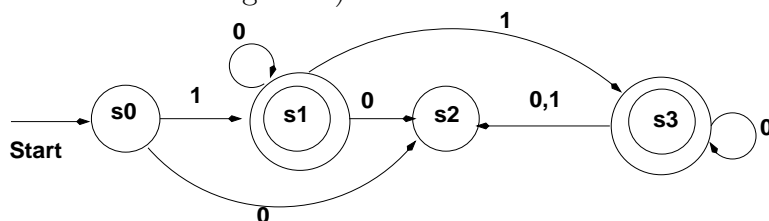
Your new boss at Facebook wants you to design a finite state machine that recognizes the set of all bit strings containing an even number of 1's and an even number of 0's. Budgets are tight so he can only authorize three states for this machine.

Give him the machine (Regular expression or FA diagram) or help him explain to Upper Management why his bonus shouldn't suffer even if he can't get this job done.

I saw a lot of “FSMs have no memory”, which is wrong – they have as much memory as they have states. I also saw “you need to count arbitrary number of 1’s and 0’s, and FSM’s can’t do that.”, Well, they can’t but you don’t – you only need to compute parity, which is a 2-state problem (remember even or odd). On bright side, I saw some very clear thinking about how many states you need, some nice constructive solutions with 4 states, and best of all some proofs that you can’t minimize states to fewer than 4 by our minimization method.

3 Misprints and Regular Expressions (15 min.)

Here’s a figure from Rosen, (one of our texts): *Discrete Mathematics and its Applications*, p. 816. He uses comma for alternation, so this is meant to be a Nondeterministic FA (since from s3 it can accept (emit) 0 or 1 when transitioning to s2).



- (2 min.) There seems to be an obvious misprint: what is it?
- (8 min.) You may ignore conversion procedures of the last problem if you like. Please create a deterministic finite automaton (circles and arrows) and a regular expression for the language accepted by the NFA Rosen had in mind.

s2 is the problem: many people looked at it as an error, non-accepting state, deleted it, and simplified the problem that way. That is kind of backwards to the intent: it introduces a big change in the diagram in order to preserve this interpretation. I like better either the interpretation that s2 is a terminating, accepting state and the misprint is not giving it two circles. Given what you see, namely the above diagram, maybe an even more elegant explanation is that the arrow to s2 from s3 points the wrong way. What you didn't see is text with figure that mentions the s3 to s2 transition explicitly, so in fact I think my misprint theory is a little more likely since it changes the accepted language least.

- New subject: (5 min.) Determine whether the string 01001 is in each of these regular sets (I’m using comma for alternation, so (0,1) means (0 | 1) , “zero or one”): give me “yes” or “no” (in or out) for each of the six RegExps below.
 - (0,1)*
 - 0*(10)1*
 - (010)*0*1
 - (010,011)(00,01)
 - (00)0*(01)
 - (01)*(01)*

I think everyone got this right.

4 Parsing (10 min.)

Here's a CFG with starting symbol S , nonterminals S, A, B, D, E, F ; terminals a, b, d, e, f ; λ the null symbol.

$S \rightarrow AB$

$A \rightarrow Aa$

$A \rightarrow \lambda$

$B \rightarrow Ab$

$B \rightarrow DE$

$B \rightarrow DF$

$D \rightarrow d$

$E \rightarrow e$

$F \rightarrow f$

1. (4 min.) Is this an LL (left to right, left derivation) grammar? If not please say why not and fix it so it is LL.
2. (3 min. – doesn't depend on 4.1) What are the FIRST and FOLLOW sets of nonterminal B ?
3. (3 min. – doesn't depend on 4.1) Is this grammar ambiguous? Show why or why not.

Generally everyone did fine. In fact I think I misgraded this, asserting that lambda isn't in FIRST(B). If you're affected see me.

5 Misprints and Grammars (10 min.)

Here's a quote from Rosen, (one of our texts): *Discrete Mathematics and its Applications*, p. 789.

One grammar that generates the set $\{0^n 1^n 2^n \mid n = 0, 1, 2, \dots\}$ is $G = (V, T, S, P)$ with [symbols] $V = \{0, 1, 2, S, A, B, C\}$, [terminals] $T = \{0, 1, 2\}$; starting state S ; and productions [with $\lambda =$ the null symbol]

$S \rightarrow C$

$C \rightarrow 0CAB$

$S \rightarrow \lambda$

$BA \rightarrow AB$

$0A \rightarrow 01$

$1A \rightarrow 11$

$1B \rightarrow 12$

$2B \rightarrow 22$

1. (6 min.) Give a derivation tree for 001122: If you have a problem, it should suggest that (and just how) the grammar is wrong. (Hint: one simple production is missing). Show the correction and the derivation tree from the corrected grammar.

2. (4 min.) What are the most specific technical names of the grammar's type, the associated language's type, and the associated automaton's type?

Generally people fine on this.