Homework 5

You may collaborate with *one or two* other students on the graded homework if you wish, or work alone. Collaboration must be true collaboration however, which means that the work put into each problem should be roughly equal and all parties should come away understanding the solution.

The ungraded homework may be done in whatever way works best for you. There are no rules regarding collaboration. The point of ungraded homework is to develop your abilities and prepare you for the quiz. Solutions will be provided, but they should be consulted only when you need a hint and/or afterward to compare and contrast your solution with mine.

Graded Homework

Completion of these tasks by 11:59pm Sunday April 3 is worth approximately 1-2% of your course grade. No late homework will be accepted.

1. I will soon place a "Homework quiz" on Canvas for you to complete. It will be untimed and you can take it as many times as you wish. You will be able to see your score after each submission, and the highest will be kept as your score.

Ungraded Homework

1. Create a deterministic FA equivalent to the second nondeterministic FA (NFA) given in the Nondeterministic finite automata notes. Follow the process given in the notes. The NFA given in the notes does not have state labels. When I give you my solution, I will label the NFA states "a", "b", "c", and "d" from left-to-right.
2. Following the algorithm seen in class, convert the following regular expressions into a NFA: (i) `a* + b*`, (ii) `a*ba*`, and (iii) `(a+b)*a(a+b)`.
3. Use the pumping lemma to argue that the language L = { $1^n2^n3^n$ | n ≥ 1} is not regular. This is the set of all strings that have 1's, then 2's, then 3's, and the number of each is the same: {123, 112233, 111222333, ...}. Do this by following the template in the reading. You need to come up with the value to replace `****` with and then try to explain why *xz* or *xyyz* is not in L.
4. Paste the following DFA into the "input automaton" box at http://ivanzuzak.info/noam/webapps/fsm_simulator/ and press "create automaton". The DFA you see accepts all strings over alphabet {a,b} that contain a multiple of 3 a's (ie, λ, b, bb, aaa, bbb, aaab, aaba, abaa, baaa, etc).

   ```
   #states
   s0
   s1
   s2
   #initial
   s0
   #accepting
   s0
   #alphabet
   a
   b
   #transitions
   s0:a>s1
   s0:b>s0
   s1:a>s2
   s1:b>s1
   s2:a>s0
   s2:b>s2
   ```

   Follow the algorithm seen in class for converting this DFA into an equivalent regular expression.
5. The reading on lexical analysis gives an example of converting a lexical specification into a DFA that can be used for scanning an input. An intermediate step in this process is creating an NFA based on the lexical specification. Given the following lexical specification, draw the NFA following the process. Next to each accept state write the token type associated with it.

   ```
   KEYWORD: print                      // "print" is the only string matched by this reg-ex
   ID: (p+r+i+i+n+t)(p+r+i+i+n+t)*     // An ID is one or more characters long
   NUM: (0+1)(0+1)*(.+lambda)(0+1)*    // At least one digit, optional decimal, any number of digits
   ```

6. The reading on lexical analysis explains how to convert an input into tokens. List the tokens generated by the above lexical specification for the following input: `print01 10printt0.1 printprint pri nt i`

Ungraded homework solutions

Study these after completing the homework or after struggling with it for a while.

Solutions