

Each of the problems is worth 10 points. Please write your solutions clearly and concisely. If you do not explain your answer you will be given no credit. You are encouraged to collaborate on the homework, but you *must* write your own solutions and list your collaborators on your solution sheet. Copying someone else's solution will be considered plagiarism and may result in failing the whole course.

Please turn in the solutions by 11.59pm on Thursday 24 September. The homework should be dropped off in the box labeled CSC 3130 on the 9th floor of SHB. Late homeworks will not be accepted.

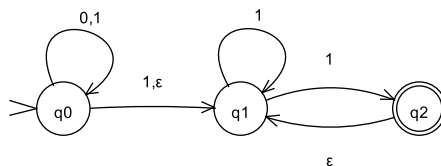
Problem 1

Give a DFA for the following languages, specified by a transition diagram. For each one of them, give a short and clear description of how the machine works. The alphabet is $\Sigma = \{0, 1, 2\}$:

- (a) $L_1 = \{w : w \text{ has fewer 0s than 1s and at most two 1s}\}$.
- (b) $L_2 = \{w : \text{the sum of the digits of } w \text{ is divisible by } 3\}$.
- (c) L_3 is the language described by $0^*1^*2^*$.
- (d) L_4 is the language described by $(12)(0 + 1 + 2)^*(21)$.

Problem 2

This problem concerns the following NFA:

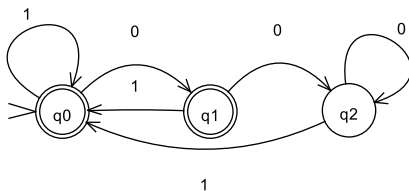


- (a) Convert this NFA into one without ε -transitions.
- (b) Convert the NFA from part (a) into a DFA using the method from class.

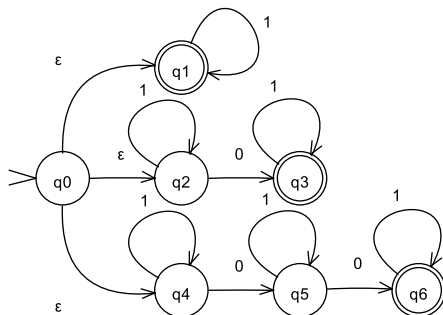
Problem 3

Consider the following languages over $\Sigma = \{0, 1\}$.

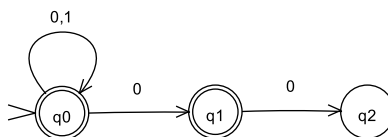
- L_1 is the language of all strings that do not end in 00.
- L_2 is the language described by $\varepsilon + 0 + 1 + (0 + 1)(0 + 1)^*$.
- L_3 is the language of all strings that do not contain the pattern 00.
- L_4 is the language of the following DFA:



- L_5 is the language described by $(0 + 1)^*(01 + 10 + 11)$.
- L_6 is the language of the following NFA:



- L_7 is the language of the following NFA:



- L_8 is the language described by $1^*(01^*)^*(01^*)^*$.

Which of these languages are the same and which are different? To show two languages are the same give a short explanation, and to show two languages are different give a string that is in one but not in the other. (You must provide an explanation to get credit.)

Problem 4

In this problem you will design an NFA that checks whether a web address is formatted correctly. Examples of correctly formatted web addresses are:

```
http://www.gov.hk/en/about/govdirectory/
http://www.cse.cuhk.edu.hk/
http://www.google.com
http://en.wikipedia.org/wiki/automaton
http://www.travelchinaguide.com/attraction/hongkong/island/victoria.htm
```

Examples of incorrect addresses are:

```
http://.invalid.domain.name/
http://no/domain/name/here.html
http:///too.many.slashes/
no.protocol.identifier
the://protocol.must.be.http
http://www.acme.com/web:page:name:cannot:contain:colons.html
```

In general, web addresses follow the Unified Resource Locator (URL) format, but for this problem you can make some simplifying assumptions about the kind of web addresses you are dealing with. You can assume that the address consists of the protocol identifier `http://`, followed by the domain name (e.g., `www.cse.cuhk.edu.hk`) ending in some suffix like `.com`, `.hk`, possibly followed by a slash `/`, then some optional directory structure (e.g., `en/about/govdirectory/`), and finally an optional web page name (e.g., `victoria.htm`).

For simplicity, assume web addresses consist only of the lowercase letters 'a', 'b', up to 'z' and the special symbols '/', ':', and '.'.

When drawing the transition diagram of your NFA, you can use the shorthand notation `[a-z]` to describe transitions labeled by all the letters 'a', 'b', ... 'z'. You can also label a single transition by multiple symbols: For instance, a transition labeled by 'cat' stands for three consecutive transitions labeled by 'c', 'a', and 't' respectively.

This is a design problem, and part of your job is to figure out a way to distinguish among correct and incorrect web addresses. (There may not be a single right answer.) You must describe your reasoning clearly in your solution. Solutions that only provide a picture of an NFA with no explanation will get no credit.