Collaborating on homework is encouraged, but you must write your own solutions in your own words and list your collaborators. Copying someone else's solution will be considered plagiarism and may result in failing the whole course.

Please answer clearly and concisely. Explain your answers. Unexplained answers will get lower scores or even no credits.

(1) (18 points) Briefly justify (in about 3-6 sentences) the following statements.

(a) If L is regular language over $\Sigma = \{0, 1\}$, then

$$L' = \{w \in L \mid w \text{ contains no } \texttt{1s}\}$$

is also regular.

(b) There are irregular languages L_1 and L_2 such that $L_1 \cup L_2$ is regular.

(c) There is a regular language L such that

$$L^A = \{u \mid u \text{ is an anagram of } w \in L\}$$

is irregular.

Note: A string u is an anagram of a string w if u is obtained from w by rearranging the symbols. For example, $\texttt{state}$ is an anagram of $\texttt{taste}$.

Formally, if $w = w_1 w_2 \cdots w_n$, then $u = w_{\sigma(1)} w_{\sigma(2)} \cdots w_{\sigma(n)}$ is an anagram of w, where σ is a permutation of $\{1, \cdots, n\}$.

(2) (30 points) Show that the following languages are irregular using the pumping lemma or pairwise distinguishable strings.

(a) $L_1 = \{w \in \{0, 1\}^* \mid w \text{ has no more 1's than 0's}\}$
   e.g. $\texttt{01} \in L_1$ but $\texttt{101} \notin L_1$

(b) $L_2 = \{ww^R \mid w \in \{0, 1\}^*\}$
    w^R denotes the string w written backwards, e.g. if $w = \texttt{war}$ then $w^R = \texttt{raw}$

(c) $L_3 = \{u \# v \mid u, v \in \{0, 1\}^* \setminus \{\varepsilon\} \text{ and } \text{val}(u) \geqslant \text{val}(v)\}$
    Here $\text{val}(u)$ denotes the natural number represented by the nonempty bit-string u in binary representation. For example, $\text{val}(\texttt{101}) = 5$ and $\texttt{101\#100} \in L_3$.

(3) (34 points) Consider the following DFA:
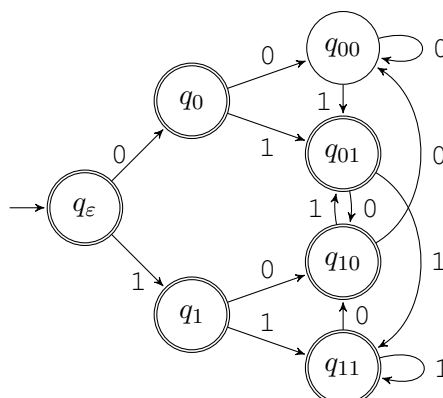
(a) Run the minimization algorithm on this DFA. Show the table of pairs of distinguishable states at the end of the algorithm (see page 22 of Lecture 7). Also draw the minimized DFA.

(b) Show that every pair of distinct states in the minimized DFA from part (a) is distinguishable (by giving a string to distinguish them, similar to page 10 of Lecture 7).

(c) Convert the following NFA into a regular expression using the conversion algorithm from class. Show the preprocessing step and how the NFA changes after each state is eliminated.
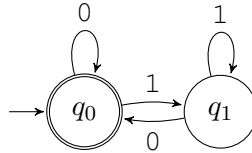
(4) (18 points) This is the only problem in this course where we deal with POSIX-style regular expressions, as opposed to regular expressions in formal language theory.

For this question, we consider a, e, i, o, u, y as a vowel. Any other letter is a consonant.

The file `propernames` contains a partial list of first names. Every line contains a name. Each name consists of one or more English letters in upper or lower case. To search for each of the following piece of information in the file, write a `grep` command of the form

```
grep -iE 'regex' propernames
```

Also give a short explanation (1-3 sentences) how your regex works in each case.

Recall that the option `-i` ignores distinction between upper and lower case. Test your commands with the file `http://www.cse.cuhk.edu.hk/~siuon/csci3130/other/propernames`. It's the best to run `grep` on Linux; `grep` in MacOS may behave a bit differently.

(a) Any name that begins and ends with vowel(s), such as **E**dd**y**.

(b) Any name without consecutive consonants, so Au**dr**ey is excluded.

(c) Any name that begins and ends with the same letter, such as **D**onal**d**.
For part (c) you may want to use the backreference feature of `grep`.

Your solution must not use `-v` flag of `grep` which swaps non-matches with matches.