2.1

a. $c^*a\{abc\}^*$

b. $\{\{bc\}^* | a\{bc\}^*a\}^*$

c. $1\{01\}^*00 | 0$

d. $1\{01\}\{01\}\{01\}\{01\}\{01\}\{01\}^+ |$

$11\{01\}\{01\}\{01\}\{01\} |$

$1011\{01\}\{01\} |$

$10101\{01\}$

e. $\{\{ac\} | b\{ab|b\}^*\{c|ac\}\}^* \{b\{b|ab\}^*\{a|\varepsilon\} | \varepsilon\}$

f. $0 | \{1-9\}\{0-9\}^* | 0\{1-7\}\{0-7\}^*$

g. 由费马大定理得 使 $a^n+b^n=c^n$ n为非负整数

仅 $n=1,2$ 时成立

⇒ $1 | 10$

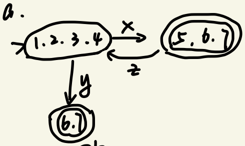
2.2 a. 比较字符a与b的个数多少，至少需要"记忆"a或b的个数，∵a或b的个数多少是随机的，假设状态数为n，当a、b的个数大于n时"记忆"工作将无法完成，此时，正则语言失效

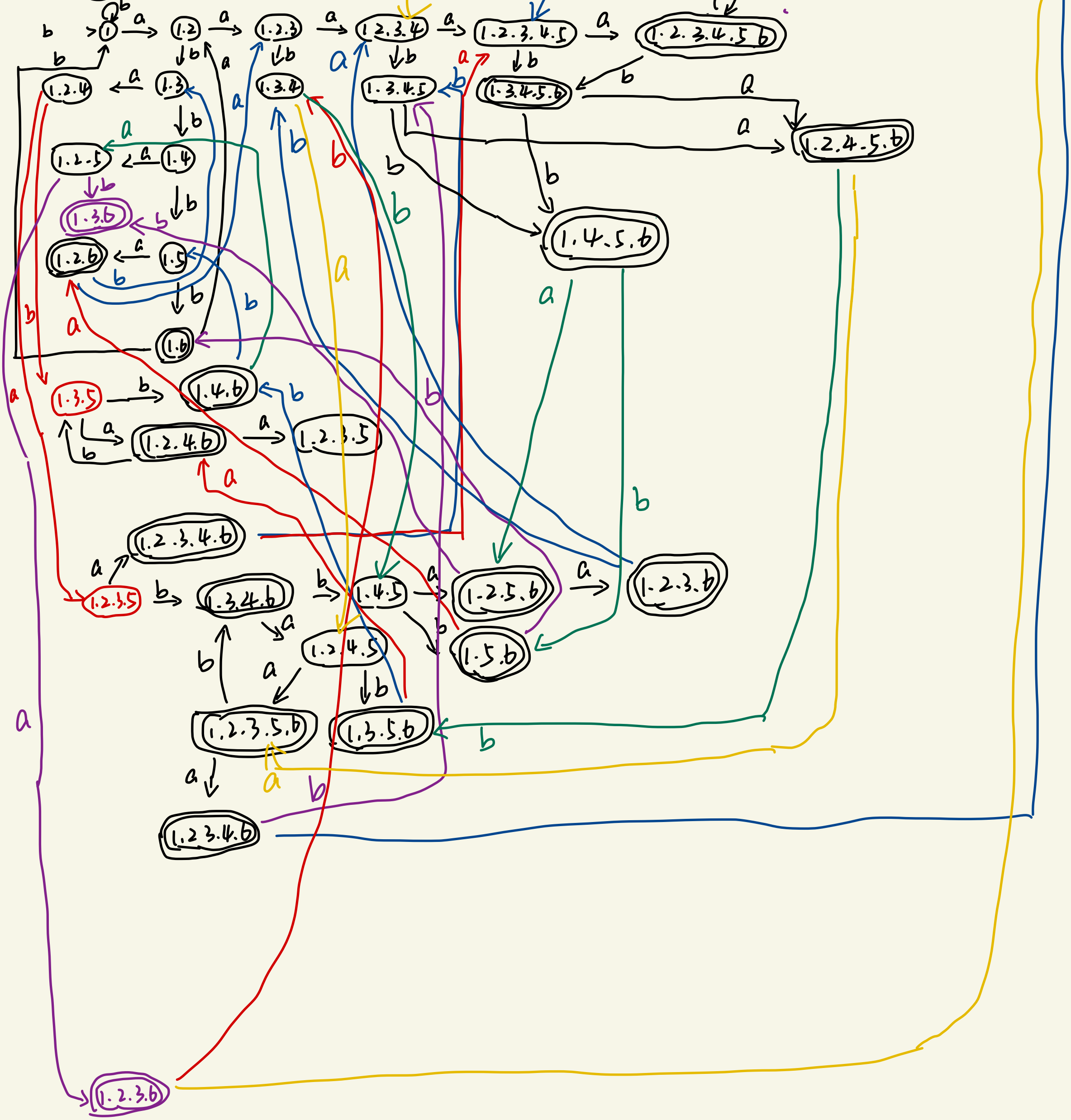
b. 回环序列的验证，假设回文序列为XYX的结构 $X\in\Sigma^+$，$Y\in\Sigma^*$ 此时我们仍假设DFA对应的状态数为n，当$|X|>n$时回文序列的前半"记忆"将无法完成，从而正则语言失效

c. 在正确语法的C语言程序中，一定存在匹配、等数量的"("与")"，假设状态数为n，当"("的数量超过n时，类似于比较"("与")"数是否一致，从而正则语言失效；另一方面，正则语言表达C语言，需要穷举所有的terminal，而Σ往往是可数的，但C语言的变量未必是可数的，从而正则语言失效。

2.5

a.

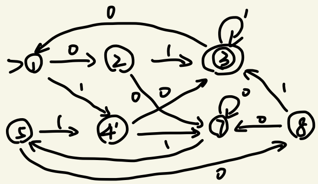




2.6

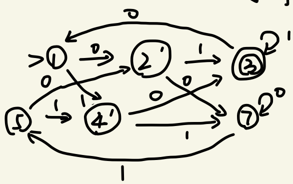
state 6 与 state 4 等价

⇒



state 2 与 state 8 等价

⇒



state1 与 state 5 等价

⇒