CS English 2021 - Written Assignment 1

Due Tuesday, Sep. 20, 2021 at 11:59 PM

1. Ex2.1 说明，做这题同学们要意识到正则式其实就是集合，所以要确保答案给出的正则式所产生的集合，和题意必须完全匹配。

   (a) a[a-z]*a | a

   (b) a[a-z]* | [a-z]*a | a[a-z]*a
   题目要求是写“一个”正则式，有同学写了多个正则式，然后用语言说，“或者 XXX，或者 XXX”是不符合要求的，这属于概念性错误。

   (c) [1-9][0-9]* | 0

   (d) [0-9]*(0|2|4|6|8)

   (e) [^9]*[^2]*
   说明：在 9 出现之前，可以任意写其它数字，9 出现后，不能写 2

   (f) ((ϵ|b|bb)a)*|(ϵ|b|bb)
   说明：所有在 b 中间隔 a* 的写法是不对的，因为 a* 隔不住任何东西

   (g) ((b*ab*ab*)*b*ab*)|((a*ba*ba*)*a*ba*)

   (h) ((aa|bb)|(ab|ba)(aa|bb)*(ab|ba))*
   说明：aa 和 bb 的组合可以，ab 和 ba 的组合也可以，但是要求它们可以出现在任意的位置。另外，注意这题用的是 and 也就是说同时要满足 a 和 b 都是偶数，所以出现任何一个 a* 或 b* 的都是错的。

   (i) there is no regular expression, because regular expression can't count.
   说明：如果写 a 和 b 不能数量相等是错的。如果回答 no regular expression 注意一定要写 why，因为题目就是这样问的。这里的不能数，是指“requires counting of arbitrary many a's”是 impossible 的，而要“count 3 个 a”或者“偶数个 a”是可以的。

2. Ex2.2

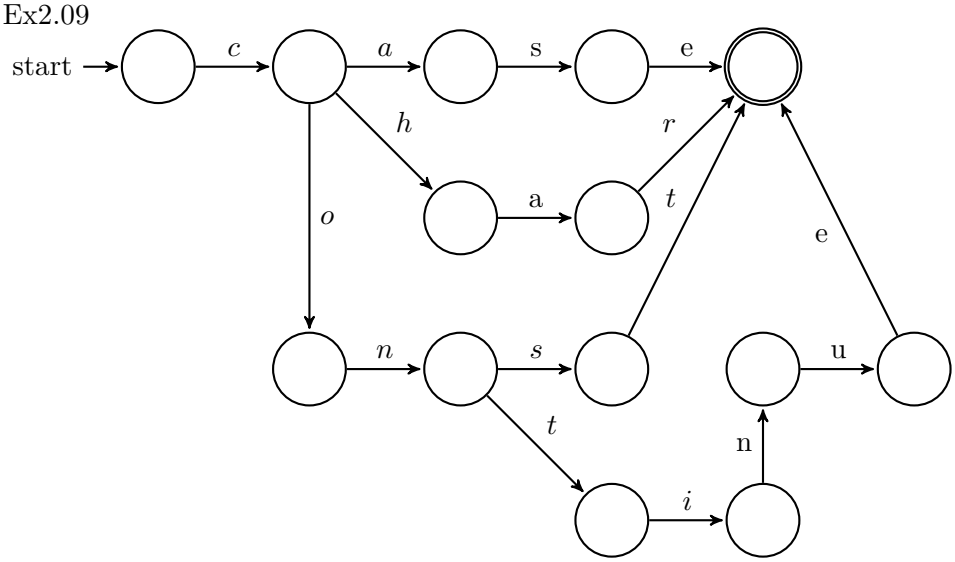
   (a) all the string of a's and b's { this is alphabet} , that end with a,ab or aa.
   说明，这题的要点也是“你用语言描述的集合，必须完全等同于正则式表达的集合”。

   (b) all word in the English alphabet{ this is alphabet} of one or more letters{ this is length description} , which start with one capital letter and dont contain any other capital letters.

(c) all the string of a's and b's { this is alphabet} , that can be divided into 2 substrings, where in the left substring, the even number of consecutive a's are seperated by b's, while in the right part, the even number of consecutive b's are seperated by a's.{ refere the Qestion 2.13}

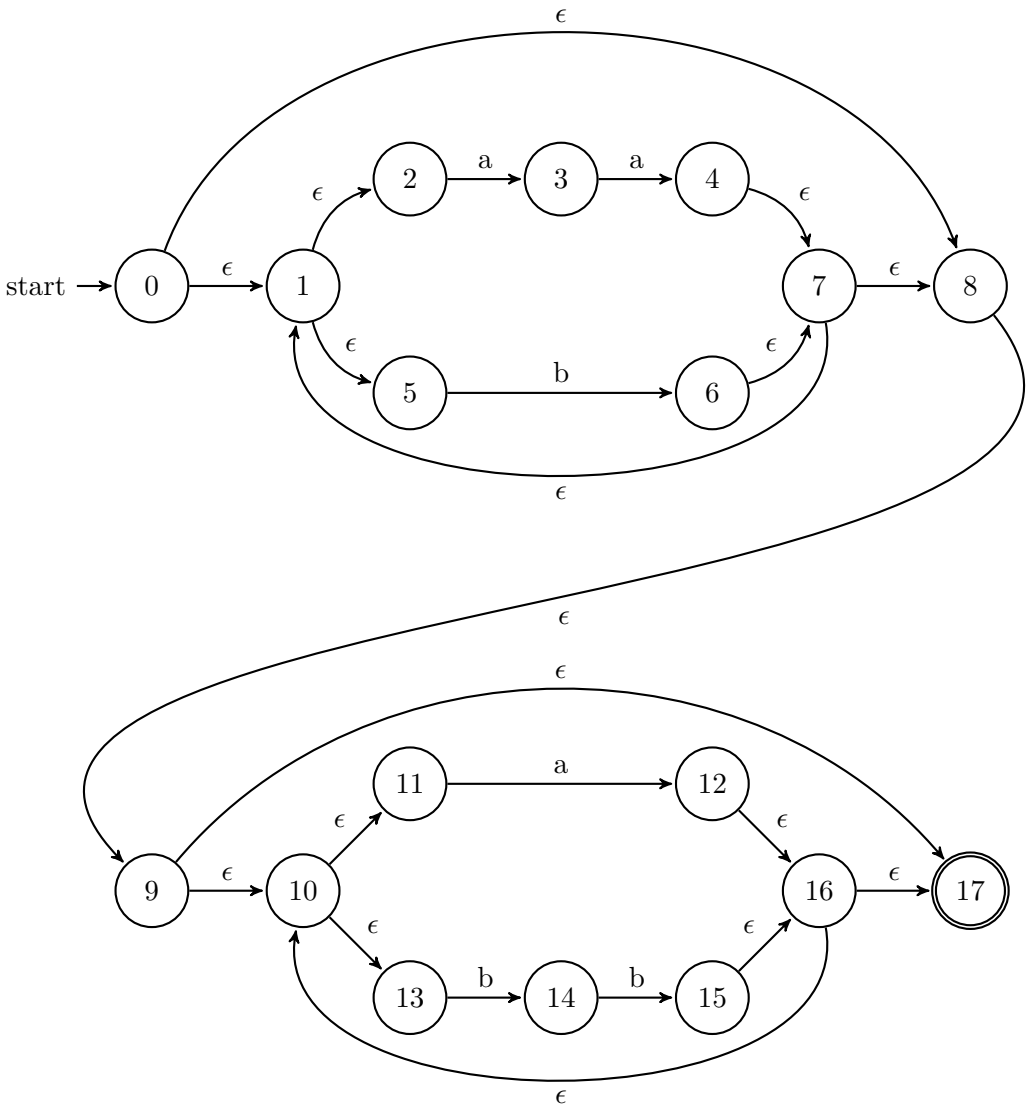
(d) all hexdecimal numbers { this is alphabet} of length one or more using the numbers zero through nine and capital letters A through F, and they are denoted with a lowercase or upcase "X" at the end of the string.

3. Ex2.09



说明：每个 label 只能放一个字符，有同学 "char" 写在一起是不对的，因为无法进行状态分析了。跟 input 的 token 有关的是 label 而不是状态，有同学把字符标在状态上，是概念性错误。

4. Ex2.13



b. The subsets constructed as follows:
{ $\overline{0}$ } = { 0,1,2,5,8,9,10,11,13,17 } 注：表示状态 0 的 ϵ 闭包
{ 0 } $_a$ = { 3,12 }
{ 0 } $_b$ = { 6,14 }

{ $\overline{3,12}$ } = { 3,10,11,12,13,16,17 }
{ 3,12 } $_a$ = { 4,12 }
{ 3,12 } $_b$ = { 14 }

{ $\overline{4,12}$ } = { 1,2,4,5,7,8,9,10,11,13,17 }
{ 4,12 } $_a$ = { 3,12 }
{ 4,12 } $_b$ = { 6,14 }

{ $\overline{6,14}$ } = { 1,2,5,6,7,8,9,10,11,13,14,17 }

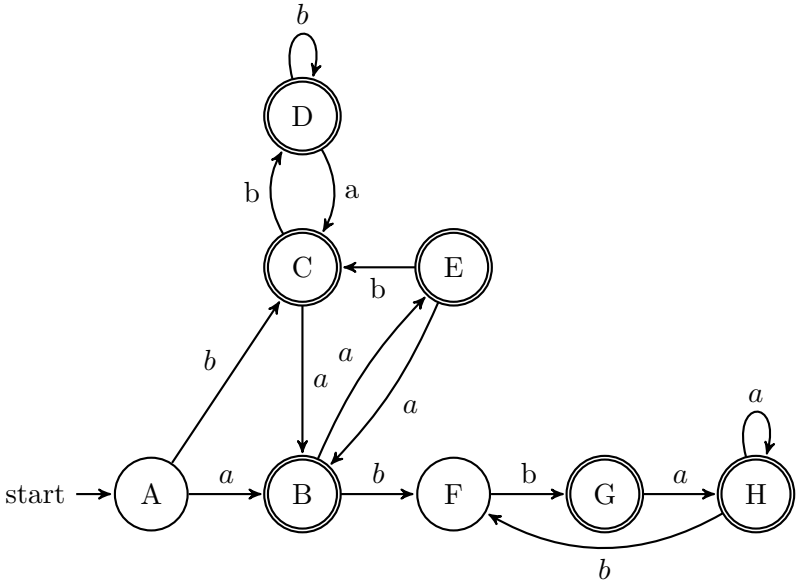
$\{ 6,14 \}_a = \{ 3,12 \}$
$\{ 6,14 \}_b = \{ 6,14,15 \}$

$\{ \overline{0} \} = \{ 1,2,5,6,7,8,9,10,11,13,14,15,16,17 \}$
$\{ 6,14,15 \}_a = \{ 3,12 \}$
$\{ 6,14,15 \}_b = \{ 6,14,15 \}$

$\{ \overline{14} \} = \{ 14\}$
$\{ 14 \}_a = \Phi$
$\{ 14 \}_b = \{ 15 \}$

$\{ \overline{15} \} = \{ 10,11,13,15,16,17\}$
$\{ 15 \}_a = \{ 12 \}$
$\{ 15 \}_b = \{ 14 \}$

$\{ \overline{12} \} = \{ 10,11,13,12,16,17\}$
$\{ 12 \}_a = \{ 12 \}$
$\{ 12 \}_b = \{ 14 \}$



说明：所有包含状态 17 的状态组合，均为接受状态 accepting state。
观察：正则式表明，接受的字符串分二部分，前半部分字符 a 成对出现，字符 b 任意，后半部分，字符 b 成对出现，字符 a 任意。在 DFA 中，{ C,D,E}3 个状态，基本建模了前部正则式的各种状态。{ F,G,H}3 个状态状态可看作后半部的活动范围。状态 A 和 B 可看作前部或后部的入口。B 承担了前后分隔的功能。当 DFA 到达状态 B，意味着它目前已经接受了奇数个 a，它检查下一个接收的字符是否仍是 a，如是，则仍停留在前部活动，如果此时输入为 b，就表示已出现单个 a，因此应进入正则式的后部，在后部 a 必须成双出现，而且后部不再能返回前部。

5. Ex2.16
   a.
   Step 1: Divide the state set into two subsets:

{ 1, 2, 3}
{ 4, 5}
Step 2: Further divide the subset 1,2,3 into two new subsets:
{1}
{2, 3}
Step 3: Can not divide the subsets any more, finally obtains three subsets:
{1}
{2, 3}
{4, 5}
Therefore, the minimized DFA is: