Homework 1 for CS310M

Autumn 2021

1 Finite State Automata and Regular Expressions

1. Using notation $\#a(x)$ to denote count of letter a in word x, define the langauge L having all words x over the alphabet $\{a, b\}$ where the absolute value of difference between $\#a(x)$ and $\#b(x)$ is at most 2. ~~Construct a~~ no DFA recoginizing this language.

   Proof : Let $\exists$ finite state then consider $a^n b^{n+2}$.
   $a, a^2, \ldots a^n$. atleast two in same state.
   Then a^i, a^j $(i<j)$ b^{i+2} (one accept one reject)

2. Define in simple terms the language accepted by the following DFA. You may use $\#a(x)$ to denote the count of letter a in word x.

		a	b
$\rightarrow$	1	2	3
	2	3	1
	$3F$	1	2

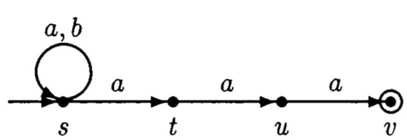
3. Construct DFA for the following languages
   - Kozen HW1 Ex. 1(b) 1(c) 1(d)
   - set of words over $\{0, 1\}$ which end with 00
   - set of words over $\{0, 1\}$ which contain 000 occuring consecutively (not necessarily at the end)
   - set of words over $\{0, 1\}$ where letters 011 occur in sequence but not necessarily consecutively.

4. (level *) Construct DFA for the following languages over the alphabet $\{0, 1\}$.
   - Each block of five consecutive letters contains at least 2 occurrences of 0.
   - Biary strings x such that the number $\hat{x}$ corresponding to string x is divisible by three. (Leading 0 are allowed).

5. (product construction) Kozen HW1 Ex 2. (trivial)

6. Give ϵ-NFA (which can even be an NFA or a DFA, as convenient) over the alphabet $\{0,1\}$ for accepting each of the following languages.
   - The beginning and the ending letter are the same.
   - Letter 0 can only occur at positions which are multiple of 3, but letter 1 can occur at any position.
   - Letter 0 occurs at least twice but letter 1 occurs at most once.
   - Words obtained by deleting zero or more letters (at arbitrary positions) from 10110.
   - words containg subword 000 but not containing subword 001.
7. (Word not in language) Kozen Misc Ex 3. (trivial)
8. Construct a regular expression over alphabet $\{0,1\}$ for the languages below
   - words where count of 0 is multiple of 3. (Include the word ϵ too).
   - Letter 0 can only occur at positions which are multiple of 3, but letter 1 can occur at any position. (positions are counted as $1,2,3,\ldots$).
   - Words obtained by deleting 0 or more letters (at arbitrary positions) from 10110.
   - Letter 0 occurs at least twice but letter 1 occurs at most once.
   - words that do not contain 01 as a subword.
   - Kozen HW3 Ex 1(a,b,c).
9. Construct Pattern Expression for valid email addresses as defined below.
   - Must only contain characters a b 0 1 . @
   - Symbols . and @ are not adjecent
   - must start with a letter (a or b)
   - Has exactly one @
   - No digit $(0,1)$ after @
   - At least one . after @

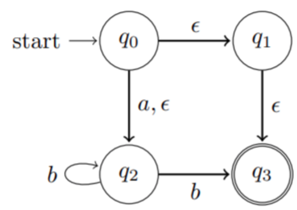
   Recall that pattern expression has more operators than regular expression. You may use the expression $\#^*$ to denote all words in you pattern. Do you think it is easy to give a regular expession for the valid email addresses?
10. For the following NFA construct DFA using the subset construction, retaining only the reachable states. Clearly label each state of DFA with the subset of NFA states it represents.



11. For the following ϵ-NFA construct DFA using the subset construction, retaing only the reachable states. Clearly label each state of DFA with the subset of NFA states it represents.

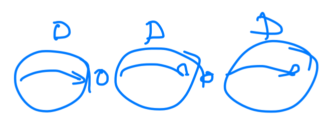
	ϵ	a	b	c
$\rightarrow p$	$\{q,r\}$	$\emptyset$	$\{q\}$	$\{r\}$
q	$\emptyset$	$\{p\}$	$\{r\}$	$\{p,q\}$
$*r$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

12. For the following ϵ-NFA construct DFA using the subset construction, retaing only the reachable states. Clearly label each state of DFA with the subset of NFA states it represents.

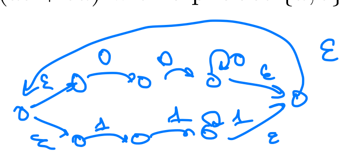


By guessing, give a regular expression for the language recognized by the above automaton.

13. Give ϵ-NFA for the following regular expressions. Determinize the resulting automata (retaining only the reachable states).



- $(ab + ba)(ab + ba)(ab + ba)$ with alphabet $\{a, b\}$
- $(000^* + 111^*)^*$.



14. Kozen Misc. Ex 13

15. (Closure Properties)

- Let prefix $x \leq_p y$ denote that x is a prefix of y, i.e. $\exists z$ s.t. $y = xz$. Define $Pref(L) = \{x \mid x \leq_p y, \text{ for some } y \in L\}$. Show that if L is regular then $Pref(L)$ is also regular.
- Kozen Misc 28 (delete one 1).
- (level *) Kozen HW2 Ex3 (Hamming Distance)

To do : 8, 13, 14

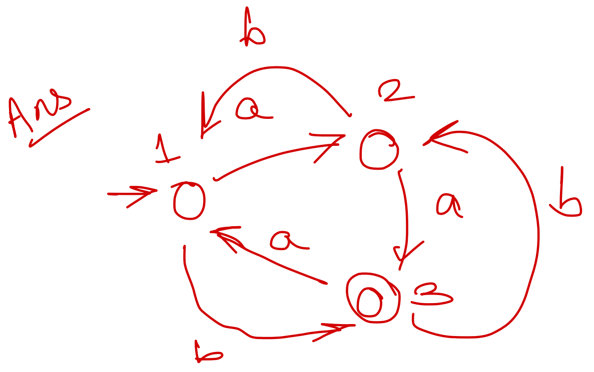
1. Using notation $\#a(x)$ to denote count of letter a in word x, define the langauge L having all words x over the alphabet $\{a, b\}$ where the absolute value of difference between $\#a(x)$ and $\#b(x)$ is at most 2. Construct a DFA recogininzing this language.

a^n . then add b''

2. Define in simple terms the language accepted by the following DFA. You may use $\#a(x)$ to denote the count of letter a in word x.

		a	b
$\rightarrow$	1	2	3
	2	3	1
	3F	1	2

Ans

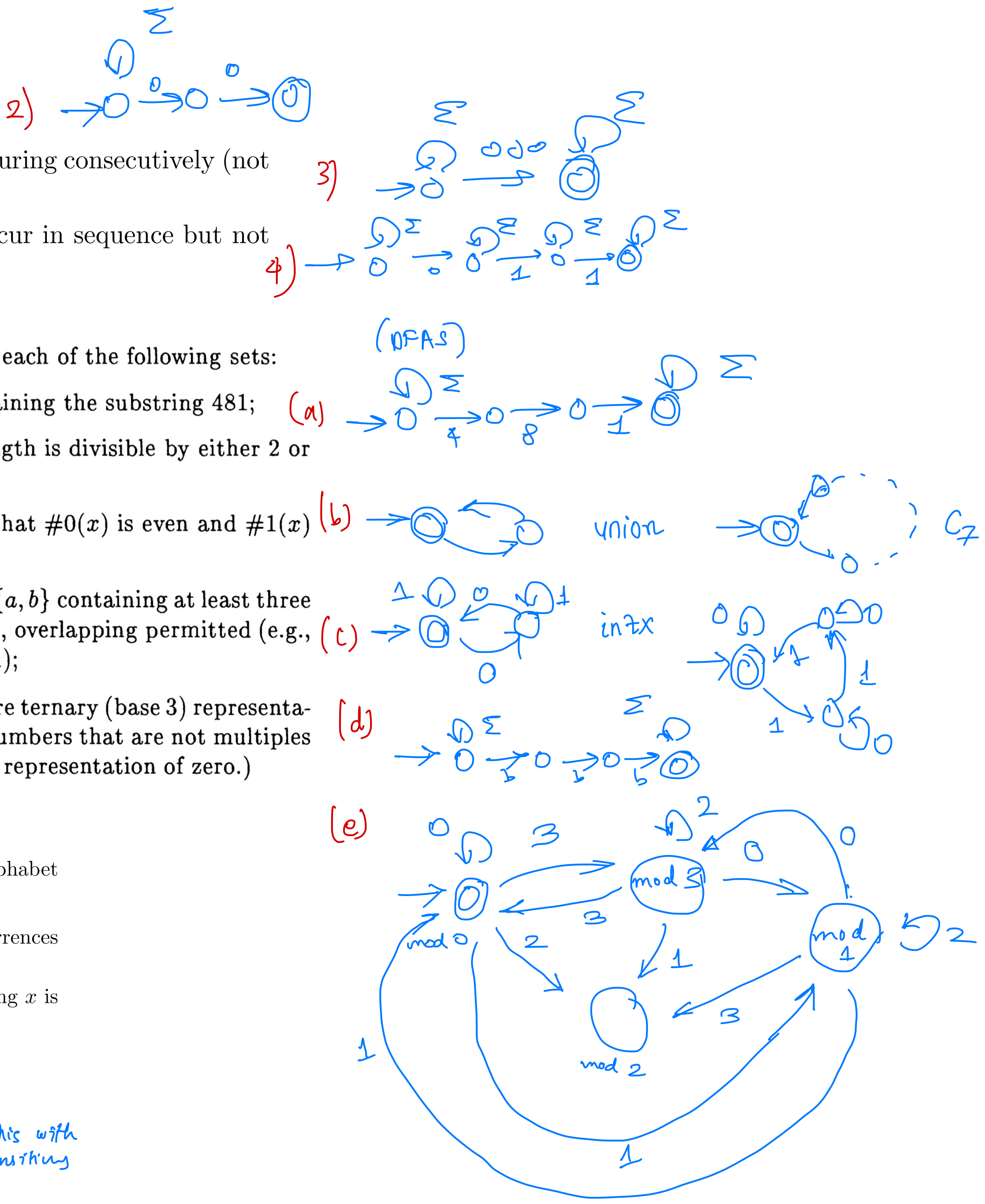


$\{x \mid \#a(x) - \#b(x) \equiv 2 \bmod 3\}$

3. Construct DFA for the following languages

- Kozen HW1 Ex. 1(b) 1(c) 1(d)
- set of words over $\{0, 1\}$ which end with 00
- set of words over $\{0, 1\}$ which contain 000 occuring consecutively (not necessarily at the end)
- set of words over $\{0, 1\}$ where letters 011 occur in sequence but not necessarily consecutively.

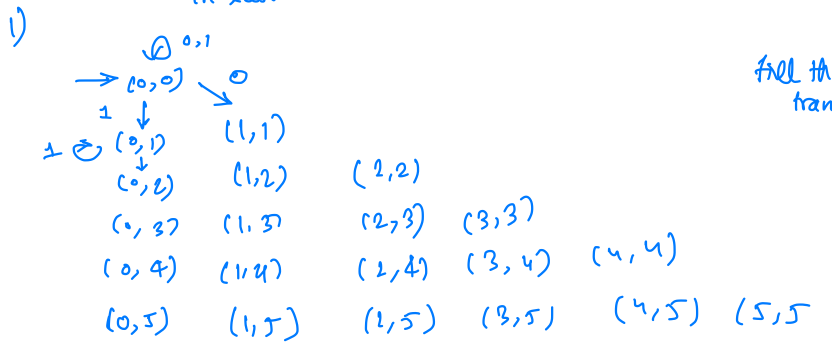
1. Design deterministic finite automata for each of the following sets:
   (a) the set of strings in $\{4, 8, 1\}^*$ containing the substring 481;
   (b) the set of strings in $\{a\}^*$ whose length is divisible by either 2 or 7;
   (c) the set of strings $x \in \{0, 1\}^*$ such that $\#0(x)$ is even and $\#1(x)$ is a multiple of three;
   (d) the set of strings over the alphabet $\{a, b\}$ containing at least three occurrences of three consecutive b's, overlapping permitted (e.g., the string $bbbbb$ should be accepted);
   (e) the set of strings in $\{0, 1, 2\}^*$ that are ternary (base 3) representations, leading zeros permitted, of numbers that are not multiples of four. (Consider the null string a representation of zero.)



4. (level *) Construct DFA for the following languages over the alphabet $\{0, 1\}$.

- Each block of five consecutive letters contains at least 2 occurrences of 0.
- Biary strings x such that the number $\hat{x}$ corresponding to string x is divisible by three. (Leading 0 are allowed).

1) a zeros in last b = (a, b)



fill this with transitions

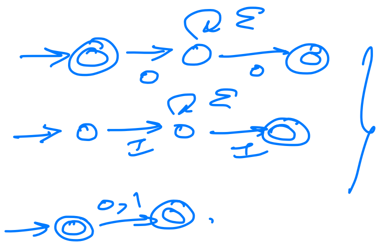
M2: make states with last 5 letters.

2) mod 3 constructions.

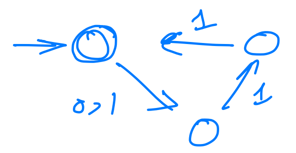
6. Give ϵ-NFA (which can even be an NFA or a DFA, as convenient) over the alphabet $\{0,1\}$ for accepting each of the following languages.

- The beginning and the ending letter are the same.
- Letter 0 can only occur at positions which are multiple of 3, but letter 1 can occur at any position.
- Letter 0 occurs at least twice but letter 1 occurs at most once.
- Words obtained by deleting zero or more letters (at arbitrary positions) from 10110.
- words containg subword 000 but not containing subword 001.

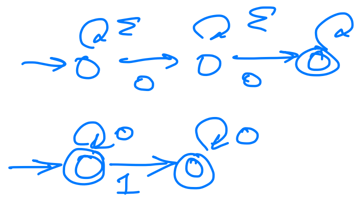
a)



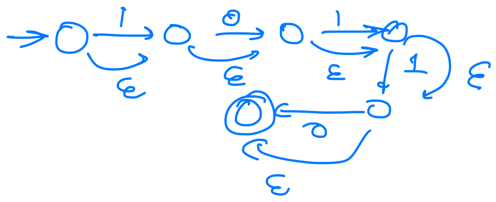
b)



c)



d).



e) complement 001 ∩ 000.

(TBD)

15. (Closure Properties)

- Let prefix $x \leq_p y$ denote that x is a prefix of y, i.e. $\exists z$ s.t. $y = xz$. Define $Pref(L) = \{x \mid x \leq_p y, \text{ for some } y \in L\}$. Show that if L is regular then $Pref(L)$ is also regular. → make states from which final states reachable accepting
- Kozen Misc 28 (delete one 1). → copy automata, ε transition
- (level *) Kozen HW2 Ex3 (Hamming Distance)

3. The *Hamming distance* between two bit strings x and y (notation: $H(x,y)$) is the number of places at which they differ. For example, $H(011,110) = 2$. (If $|x| \neq |y|$, then their Hamming distance is infinite.) If x is a string and A is a set of strings, the Hamming distance between x and A is the distance from x to the closest string in A:

$$H(x,A) \stackrel{\text{def}}{=} \min_{y \in A} H(x,y).$$

For any set $A \subseteq \{0,1\}^*$ and $k \geq 0$, define

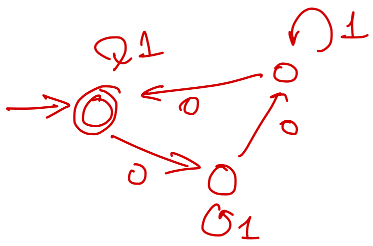
$$N_k(A) \stackrel{\text{def}}{=} \{x \mid H(x,A) \leq k\},$$

the set of strings of Hamming distance at most k from A. For example, $N_0(\{000\}) = \{000\}$, $N_1(\{000\}) = \{000, 001, 010, 100\}$, and $N_2(\{000\}) = \{0,1\}^3 - \{111\}$.

Prove that if $A \subseteq \{0,1\}^*$ is regular, then so is $N_2(A)$. (*Hint:* If A is accepted by a machine with states Q, build a machine for $N_2(A)$ with states $Q \times \{0,1,2\}$. The second component tells how many errors you have seen so far. Use nondeterminism to guess the string $y \in A$ that the input string x is similar to and where the errors are.)

- copy, put one 1 instead of 0
- repeat.

8. Construct a regular expression over alphabet $\{0,1\}$ for the languages below



- words where count of 0 is multiple of 3. (Include the word ϵ too).
- Letter 0 can only occur at positions which are multiple of 3, but letter 1 can occur at any position. (positions are counted as $1, 2, 3, \ldots$).
- Words obtained by deleting 0 or more letters (at arbitrary positions) from 10110.
- Letter 0 occurs at least twice but letter 1 occurs at most once.
- words that do not contain 01 as a subword.
- Kozen HW3 Ex 1(a,b,c).

a) $(1^*01^*01^*01^*)^*$

b) $(110+111)^*$

c) $(\varepsilon+1)(\varepsilon+0)(\varepsilon+1)\ldots$

d) $0^*(010+100+001)0^*$

(+, ·, *)

e) 1^*0^*

1. Give regular expressions for each of the following subsets of $\{a, b\}^*$.

   (a) $\{x \mid x$ contains an even number of a's$\}$

   (b) $\{x \mid x$ contains an odd number of b's$\}$

   (c) $\{x \mid x$ contains an even number of a's or an odd number of b's$\}$

   ~~(d)~~ $\{x \mid x$ contains an even number of a's and an odd number of b's$\}$

   Try to simplify the expressions as much as possible using the algebraic laws of Lecture 9. Recall that regular expressions over $\{a, b\}$ may use ϵ, $\emptyset$, a, b, and operators $+$, *, and $\cdot$ only; the other pattern operators are not allowed.

(a) $(b^*ab^*ab^*)^*$

(b) $a^*ba^*(a^*ba^*ba^*)^*$

(c) +

(d) $(b(a\ a)b)^*$

a a b a a b

aa + bb + ($(aa)^*(bb)^*b$ a a b)

4! 4! × 32

0 0 0 0

2 2 2 2 2

b a b b a b

r = both even. ← {

(odd a) b (bb*) a r

+ (aa)* b r

} for other cases.