CS 383
Exam 2 Solutions

The exam has 6 questions. #1 is worth 20 points; the other five are worth 16 points each

1. Which of the following languages are context-free? Read the descriptions carefully. Write "CF" next to the languages that are context-free, "N" next to the ones that are not. No proofs are necessary.
   a. $\{0^n1^n0^n \mid n >= 0\}$ Not CF; a pumping lemma argument shows this.

   b. $\{0^n1^n1^n \mid n >= 0\}$ CF; this is the same as $\{0^n1^{2n} \mid n>=0\}$

   c. $\{0^n1^m1^m0^n \mid m, n >= 0\}$ CF; this is $\{0^n1^k0^n \mid n>=0$ and k is even$\}$

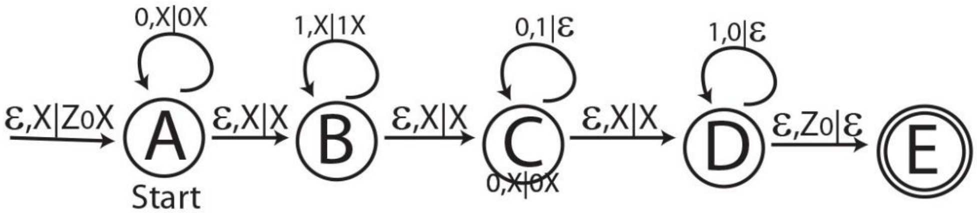
   d. Strings of 0s and 1s with odd length that have 0 as the middle element, such as 1110101 or 000 or 101. CF. Here's a grammar: S=> 0S0|0S1|1S0|1S1|0

   e. {vw | v is a string of 0s and 1s with length 3 or more and w is the first 3 letters of v} For example, 01101011 is a string in this language. CF; it is even Regular.

   f. $\{w0^n \mid$ w is a string of 0s and 1s and n is the length of w$\}$ For example, 01110000 is a string in this language. CF. Here's a grammar: S=>0S1|1S1|ε

   g. {uvw | u,v,w are all strings of 0s and 1s with the same length } For example, 011100 is a string in this language. CF. This is just the set of strings whose lengths are divisible by 3; it is Regular.

2. Construct a PDA that accepts by final state the language $\{0^n 1^m 0^m 1^n \mid m, n >= 0\}$



3. Here is a grammar. Use this grammar to construct either a parse tree or a derivation (your choice; one is about as easy or hard as the other) for the string 001122:

   A ➔ 0A2| BC

   B ➔ 0B2 | C

   C ➔ 1A1 | 1

Here's a derivation:

A ➔ 0 A 2

➔ 0 0 A 2 2

➔ 0 0 B C 2 2

➔ 0 0 C C 2 2

➔ 0 0 1 1 2 2

4. Convert the following grammar to Chomsky Normal Form:

   $A \rightarrow 0A2 \mid BC$
   $B \rightarrow 0B2 \mid C$
   $C \rightarrow 1A1 \mid 1$
   ==================
   $B \rightarrow C$ is a unit rule; remove it

   $A \rightarrow 0A2 \mid BC$
   $B \rightarrow 0B2 \mid 1A1 \mid 1$
   $C \rightarrow 1A1 \mid 1$
   ====================
   get rid of the terminal symbols in rules that have a mix of terminals and non-terminals

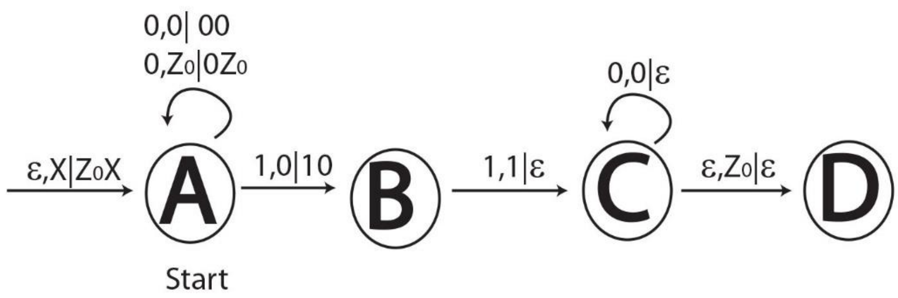
   $A \rightarrow ZAT \mid BC$
   $B \rightarrow ZBT \mid NAN \mid 1$
   $C \rightarrow NAN \mid 1$
   $Z \rightarrow 0$
   $N \rightarrow 1$
   $T \rightarrow 2$
   =====================
   break down to rules of length 2

   $A \rightarrow ZA_1 \mid BC$
   $A_1 \rightarrow AT$
   $B \rightarrow ZB_1 \mid NA_2 \mid 1$
   $B_1 \rightarrow BT$
   $A_2 \rightarrow AN$
   $C \rightarrow NA_2 \mid 1$
   $Z \rightarrow 0$
   $N \rightarrow 1$
   $T \rightarrow 2$

5. Give a careful pumping lemma proof that $\{0^i1^j2^k \mid 0<j,k<i\}$ is not a context free language. If you aren't clear about the language, it is the subset of 0*1*2* where there are more 0s than either 1s or 2s.

   Suppose this language is context free; let p be its pumping constant. Consider the string $z = 0^{p+1}1^p2^p$, which is in the language and longer than p. It should be pumpable. Consider any decomposition z=uvwxy where |vwx| < p and vx is not empty. Since |vwx| < p we know that v and x can't include all three digits. Suppose v and x together include at least one 0; then they include no 2. So uv^0wx^0y has no more 0s than 2s, so it isn't in the language. On the other hand, if v and x include no 0s then uv^1wx^1y has at least as many 1s or 2s as 0s, so it isn't in the language. In other words, for any decomposition of z there is some i for which uv^iwx^iy is not in the language, so z isn't pumable. This violates the pumping lemma, so our assumption that the language is context fee must be false.

6. In class we developed an algorithm by Noam Chomsky that constructs a grammar equivalent to a given PDA. Apply this algorithm to the following PDA and give the derivation in this grammar of the string 001100. Note that the PDA accepts by empty stack.



S ➔ [A Z_0 D]

➔0 [A 0 C] [C Z_0 D]

➔0 0[A 0 C] [C 0 C] [C Z_0 D]

➔0 0 1[B 1 C] [C 0 C} [C 0 C] [C Z_0 D]

➔0 0 1 1[C 0 C} [C 0 C] [C Z_0 D]

➔0 0 1 1 0 [C 0 C] [C Z_0 D]

➔0 0 1 1 0 0 [C Z_0 D]

➔0 0 1 1 0 0

You can use this page as extra space for any problem.

Please write and sign the Honor Pledge when you have finished the exam. If you didn't take the exam with the rest of the class also write your starting and stopping times.

Name________________________

CS 383
Exam 1 Solutions

There are 6 numbered questions. The 6 parts of Question 1 are worth 4 points each. Questions 2 through 6 are worth 15 points each. You get one point for free.

1. Which languages are regular? You don't need to prove your answers. Write an "R" in the blank next to the description of each language you think is regular. Write "N" for any language you think is not regular. In each case the alphabet is $\Sigma=\{0,1\}$

   a. __**R**__Strings that end in exactly five 1s. So 01011111 is in this language but 010111111 is not.

   b. __**R**__Strings with any number of 0s followed by an even number of 1s.

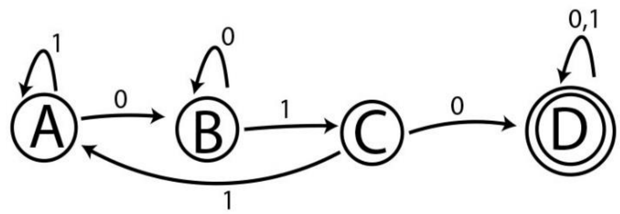
   c. __**R**__ $\{0^m1^n$ | if m is even then n is also even; if m is odd then n is also odd$\}$

   d. __**R**__Strings where the digits sum to a number divisible by 5 (i.e., the digits sum to 0, 5, 10, 15, etc.)

   e. __**N**__Strings where there are at least as many 0s as 1s.

   f. __**R**__$0^*\mathcal{L}$ where $\mathcal{L} = \{0^n |$ n is prime $\}$. Note that strings in this language have any number of 0s followed by a prime number of 0s.

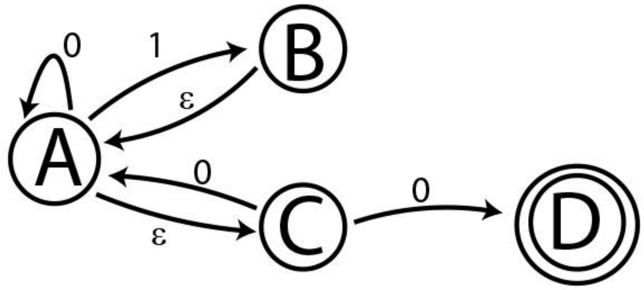
2. Give a DFA for the strings of 0s and 1s that contain the substring 010. For example, 110101 should be accepted by this DFA but 1001100 should not be accepted.



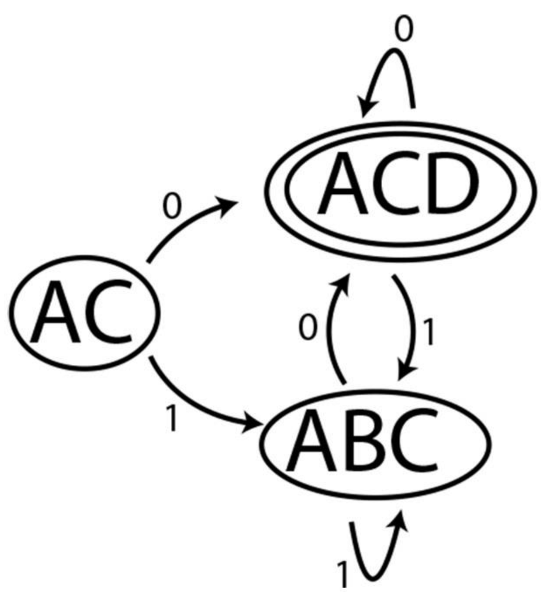
3. Here is an ε-NFA, with start state A.

a) Convert this NFA to a DFA

b) Describe in English the strings it accepts.



Solution:

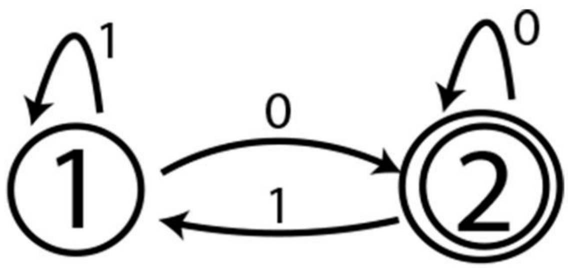


This accepts all strings ending in 0.

4. Suppose we know that for some language $\mathcal{L}$ the language $00\mathcal{L} = \{00\alpha \mid \alpha \epsilon \mathcal{L}\}$ is regular. Must $\mathcal{L}$ be regular? Either give an example where $\mathcal{L}$ is not regular and $00\mathcal{L}$ is regular, or else show that $\mathcal{L}$ must be regular if $00\mathcal{L}$ is.

The language $\mathcal{L}$ must be regular. Suppose $P = (\Sigma,Q,\delta,s,F)$ is a DFA accepting $00\mathcal{L}$. Let $q = \delta(s, 0)$ and let $q1=\delta(q,0)$. State q1 is where you get to in P on input 00. Let $P' = (\Sigma,Q,\delta,q1,F)$. P' is the same as P only with start state q1. Now suppose string α is in $\mathcal{L}$. Then 00α is in 00L and takes P from state s to q to q1 and then eventually to a final state. So α takes P' from q1 to a final state, and P' accepts α. Similarly, if α takes P' from q1 to a final state then 00α takes P from s to a final state, so 00α is in $00\mathcal{L}$ and α must be in $\mathcal{L}$. Altogether, the DFA P' accepts α if and only if α is in $\mathcal{L}$, so $\mathcal{L}$ is regular.

5. Consider the following DFA. We had an algorithm for converting a DFA to a regular expression. This involved making a table of regular expressions r_{ij}^k .



Here is the first column of a table of the r_{ij}^k expressions; find the 4 entries of the second column.

	k=0	k=1
r_{11}^k	ε+1	**1***
r_{12}^k	0	**1*0**
r_{21}^k	1	**11***
r_{22}^k	ε+0	**ε+1*0**

$$\boldsymbol{r_{ij}^1 = r_{ij}^0 + r_{i1}^0 (r_{11}^0)^* r_{1j}^0}$$

6. Use the pumping lemma to show carefully that the language $\{0^m1^n0^n \mid m>=2, n>=0\}$ is not regular.

   Suppose this language is regular; let p be its pumping constant. Let w = $0^21^p0^p$. This is longer than p, so let w=xyz be any decomposition of w where y is not empty and $|xy|<p$. All of y must come from the initial 0^21^p elements of w. If y contains any initial 0s then xy^0z has fewer than 2 initial 0s. If y contains any 1s then xy^0z has fewer 1s than trailing 0s. Either way, xy^0z is not an element of our language so our string w is not pumpable. This contradicts the Pumping Lemma, so our language can't be regular.

Name________________________

CS 383
Exam 1
October 5, 2018

There are 6 numbered questions. The 6 parts of Question 1 are worth 4 points each. Questions 2 through 6 are worth 15 points each. You get one point for free.

1. Which languages are regular? You don't need to prove your answers. Write an "R" in the blank next to the description of each language you think is regular. Write "N" for any language you think is not regular. In each case the alphabet is $\Sigma=\{0,1\}$

   a. _____Strings that end in exactly five 1s. So 01011111 is in this language but 010111111 is not.

   b. _____Strings with any number of 0s followed by an even number of 1s.

   c. _____ $\{0^m1^n \mid$ if m is even then n is also even; if m is odd then n is also odd$\}$

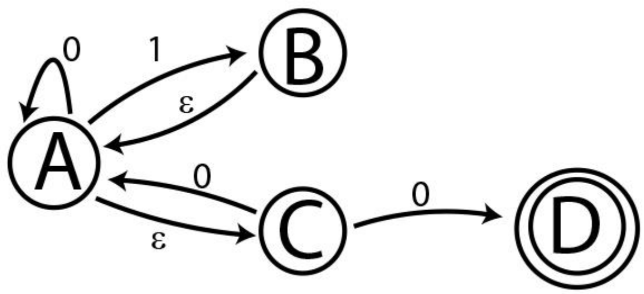
   d. _____Strings where the digits sum to a number divisible by 5 (i.e., the digits sum to 0, 5, 10, 15, etc.)

   e. _____Strings where there are at least as many 0s as 1s.

   f. _____$0^*\mathcal{L}$ where $\mathcal{L} = \{0^n \mid n \text{ is prime}\}$. Note that strings in this language have any number of 0s followed by a prime number of 0s.

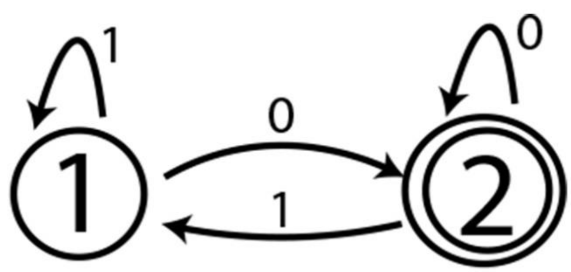
2. Give a DFA for the strings of 0s and 1s that contain the substring 010. For example, 110101 should be accepted by this DFA but 1001100 should not be accepted.

3. Here is an ε-NFA, with start state A.
   a) Convert this NFA to a DFA
   b) Describe in English the strings it accepts.



4. Suppose we know that for some language $\mathcal{L}$ the language $00\mathcal{L} = \{00\alpha \mid \alpha \in \mathcal{L}\}$ is regular. Must $\mathcal{L}$ be regular? Either give an example where $\mathcal{L}$ is not regular and $00\mathcal{L}$ is regular, or else show that $\mathcal{L}$ must be regular if $00\mathcal{L}$ is.

5. Consider the following DFA. We had an algorithm for converting a DFA to a regular expression. This involved making a table of regular expressions r_{ij}^{k} .



Here is the first column of a table of the r_{ij}^{k} expressions; find the 4 entries of the second column.

	k=0	k=1
r_{11}^{k}	ε+1	
r_{12}^{k}	0	
r_{21}^{k}	1	
r_{22}^{k}	ε+0	

6. Use the pumping lemma to show carefully that the language $\{0^m1^n0^n \mid m>=2,\ n>=0\}$ is not regular.

This page is extra space. If you want me to grade anything here indicate that clearly.

Please write and sign the Honor Pledge when you have finished the exam.

CS 383
Exam 1 Solutions
Fall 2019

There are 6 numbered questions. The 8 parts of Question 1 are worth 3 points each. Questions 2 through 6 are worth 15 points each. You get one point for free.

1. Which languages are regular? You don't need to prove your answers. Write an "R" in the blank next to the description of each language you think is regular. Write "N" for any language you think is not regular. In each case the alphabet is $\Sigma=\{0,1\}$

   a. __R__Strings of 0s and 1s that start and end on the same digit, such as 10010101 and 010. = 0(0+1)*0 + 1(0+1)*1

   b. __N__Strings with odd length that have 0 as the center element, such as 101 or 01000

   c. ___R_ Strings of 0s, 1s and 2s whose digits sum to more than 9. For example, 21202201, whose digits sum to 10, is such a string.

   d. __R_Strings whose length is divisible by 5, such as $0^3 1^7$.

   e. __N__Strings whose length is a perfect square, such as $0^9 1^7$.

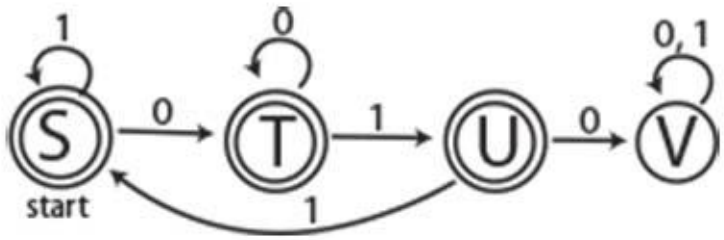
   f. __N__ $\{ 0^j 0^k 1^k \mid j>=0, k>=0 \}$

   g. __R__ $\{ 0^i 0^k 1^k 1^n \mid i>=0, k>=0, n>=0 \}$ = 0*1*

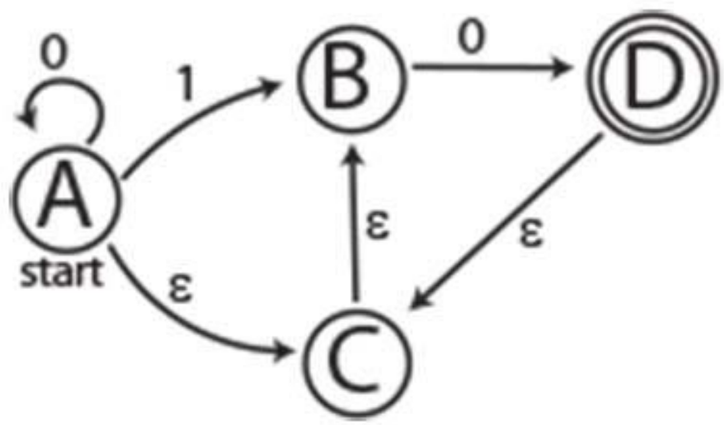
   h. ___R_ $\{ 0^k 0^k \mid k >= 0 \}$ =strings of 0s with even length

2. Give a DFA for the strings of 0s and 1s that **do not** contain 010 as a substring.

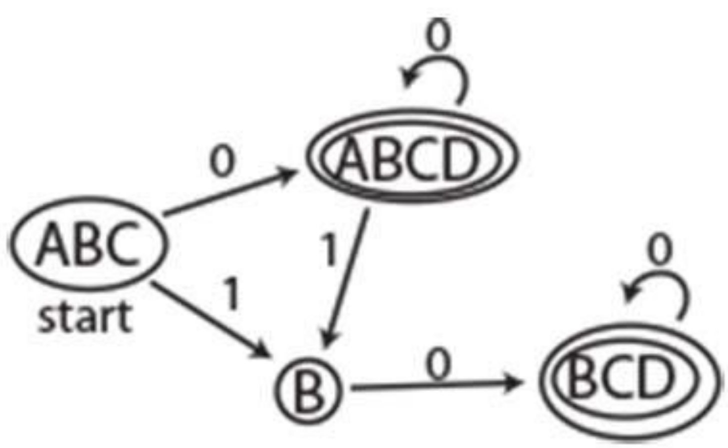
   If you can make a DFA for the strings that DO contain 010 and this DFA has all possible transitions from each node, just invert the final states:



3. Here is an ε-NFA, with start state A.
   a) Convert this NFA to a DFA
   b) Give a regular expression for the strings it accepts.



Here's my DFA:

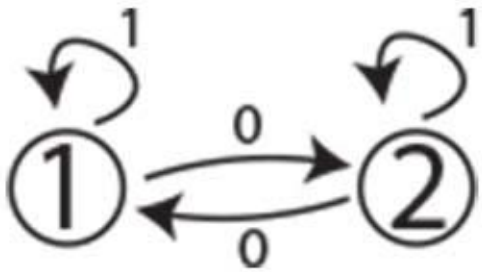


Regular expression: $0^{+}+0^{*}10^{+}$

4. Suppose $\mathcal{L}^*$ is regular. Must $\mathcal{L}$ be regular? Either prove that it must be or give an example to show it ain't necessarily so.

No. If $\mathcal{L}$ is any language that contains both 0 and 1 then $\mathcal{L}^*$ contains all strings of 0s and 1s, so $\mathcal{L}^*$ is regular. For the example we just need any language that contains 0 and 1 and isn't regular, such as $\mathcal{L} = \{0^n1^n \mid n >= 0\} \cup \{0,1\}$. The usual pumping lemma argument shows $\mathcal{L}$ is not regular, but $\mathcal{L}^*$ is denoted by the regular expression (0+1)*.

5. Consider the following DFA. We had an algorithm for converting a DFA to a regular expression. This involved making a table of regular expressions r_{ij}^{k} .



Here is the first column of a table of the r_{ij}^{k} expressions; find the 4 entries of the second column.

	k=0	k=1
r_{11}^{k}	ε+1	1*
r_{12}^{k}	0	1*0
r_{21}^{k}	0	01*
r_{22}^{k}	ε+1	=ε+1+01*0

$r_{ij}^{1}=r_{ij}^{0}+r_{i1}^{0}(r_{11}^{0})^{*}\, r_{1j}^{0}$

r_{11}^{1}=ε+1+(ε+1)1*(ε+1) = 1*
r_{12}^{1}=0+(ε+1)1*0 = 1*0
r_{21}^{1}=0+01*(ε+1) = 01*
r_{22}^{1}=ε+1+01*0

6. Consider the language of even-length strings of 0s and 1s whose first half is all 0s. For example, 000101, 000011 and even 000000 are all strings in this language. Use the Pumping Lemma to show carefully that this language is not regular.

If this language is regular, let p be its pumping constant. Consider the string $w=0^p1^p$, which is in this language. I will show that w is not pumpable. Let w=xyz be any decomposition of w into 3 parts with y not empty and |xy| <= p. Since the first p letters of w are all 0, y must consist of a positive number of 0s. The string xz has fewer 0s than 1s, so either it has odd length or its first half contains some 1s. Either way, xz is not in the language. Since there is no decomposition of w that can be pumped, this language can't be regular.

Topics for Exam 1

Basic stuff:

- DFAs
- NFAs,
- ε-NFAs
- Regular Expressions
- Regular Languages

Algorithms:

- Converting an NFA to a DFA
- Converting an ε-NFA to a DFA
- Converting a regular expression to an ε-NFA
- Converting a DFA to a regular expression
- Finding reachable states
- Finding a DFA with minimal number of states equivalent to a given DFA
- Finding if 2 DFAs are equivalent

Theorems:

- Most of our theorems proved that the constructions work
- The Pumping Lemma is used to show that some languages aren't regular
- Regular languages are closed under union, intersection, complements and reversals

Tests:

- To show that a language is regular give a regular expression or DFA for it.
- To show that a language is not regular use the pumping lemma.

Name_________________________

CS 383
Exam 2
November 19, 2018

The exam has 6 questions. #1 is worth 20 points; the other five are worth 16 points each

1. Which of the following languages are context-free? Read the descriptions carefully. Write "CF" next to the languages that are context-free, "N" next to the ones that are not. No proofs are necessary.
   a. $\{0^n1^n0^n \mid n >= 0\}$

   b. $\{0^n1^n1^n \mid n >= 0\}$

   c. $\{0^n1^m1^m0^n \mid m, n >= 0\}$

   d. Strings of 0s and 1s with odd length that have 0 as the middle element, such as 1110101 or 000 or 101.

   e. {vw | v is a string of 0s and 1s with length 3 or more and w is the first 3 letters of v} For example, 01101011 is a string in this language.

   f. $\{w0^n \mid$ w is a string of 0s and 1s and n is the length of w$\}$ For example, 01110000 is a string in this language.

   g. {uvw | u,v,w are all strings of 0s and 1s with the same length } For example, 011100 is a string in this language.

2. Construct a PDA that accepts by final state the language $\{0^n 1^m 0^m 1^n \mid m, n >= 0\}$

3. Here is a grammar. Use this grammar to construct either a parse tree or a derivation (your choice; one is about as easy or hard as the other) for the string 001122:

   A ➔ 0A2| BC

   B ➔ 0B2 | C

   C ➔ 1A1 | 1

4. Convert the following grammar to Chomsky Normal Form:

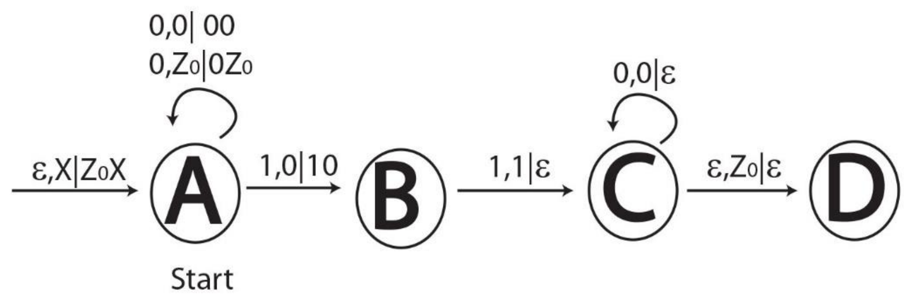
   A ➔ 0A2| BC

   B ➔ 0B2 | C

   C ➔ 1A1 | 1

5. Give a careful pumping lemma proof that $\{0^i 1^j 2^k \mid 0<j,k<i\}$ is not a context free language. If you aren't clear about the language, it is the subset of $0^*1^*2^*$ where there are more 0s than either 1s or 2s.

6. In class we developed an algorithm by Noam Chomsky that constructs a grammar equivalent to a given PDA. Apply this algorithm to the following PDA and give the derivation in this grammar of the string 001100. Note that the PDA accepts by empty stack.



You can use this page as extra space for any problem.

Please write and sign the Honor Pledge when you have finished the exam. If you didn’t take the exam with the rest of the class also write your starting and stopping times.

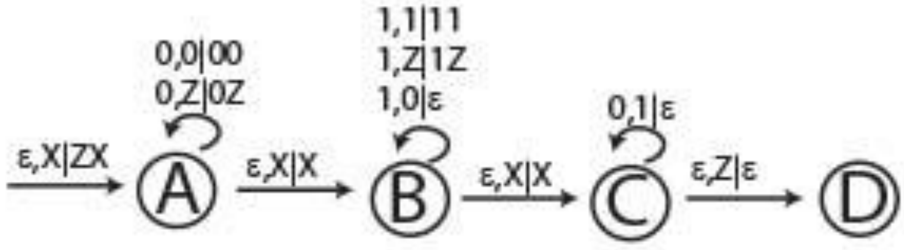
Name________________________

CS 383
Exam 2 Solutions
November 2019

1. Which of the following languages are context-free? Read the descriptions carefully. Write "C" next to the languages that are context-free, "N" next to the ones that are not. No proofs are necessary.
   a. __**C**___ $\{0^n1^n1^n \mid n >= 0\}$ This is 0^n1^{2n}. S=>0S11 | e will generate it.
   b. __N____ $\{0^n1^n1^n0^n \mid n>=0\}$ This is $0^n1^{2n}0^n$ A pumping lemma argument shows it isn'r CF
   c. __C___ $\{0^n1^m \mid$ n and m are either both even or both odd$\}$ This is Regular
   d. ___C___Strings of the digits 0-9 whose digits sum to an even number, such as 24473 or 112233. This is Regular. Have states that track whether the sum so far is even or odd.
   e. __C___ $\{0^n1^n0^m1^m \mid n>0 \text{ and } m>0\}$ This is the concatenation of 2 CF languages.
   f. ___C__ $\{0^n1^n \mid n > 0$ and n is odd$\}$ This is the intersection of 0^n1^n (Context-Free) and $\{0^n1^m \mid$ n and m are odd$\}$ which is Regular.
   g. ___N__ Strings of the form vcv where v is a string of 0s, 1s, and 2s (and c is just the letter c), such as 0210c0210. This is just like {vv} which we showed in class is not C-F.
   h. __C__ Strings of the form vcw where v and w are both strings of 0s, 1s, and 2s (and c is just the letter c), where v and w have the same length. 1210c2020 is such a string. Make a PDA that pushes A on any digit before c, then pops A on any digit after c.
   i. ___C__ Strings of the form vcw, where v and w are both strings of 0s, 1s ,and 2s (and c is just the letter c), such that the digits of v sum to the same value as the digits of w. For example, 012011c221 is such a string because the digits before and after c both sum to 5. Before c push d As on digit d. After c pop d As on digit d.

2. Construct a PDA that accepts by final state the language $\{0^n 1^{n+m} 0^m \mid m >= 0, n >= 0\}$

The following uses "Z" as the stack bottom symbol.



3. Here is a grammar:
   S => 0A2 | BC
   A => 0A2 | 02
   B =>0B | 0
   C => C2 | 2

   a. Use this grammar to construct either a parse tree or a derivation (your choice; one is about as easy or hard as the other) for the string 00022.

      S => BC
        => 0BC
        => 00BC
        => 000C
        => 000C2
        => 00022

   b. Find a string that has two completely different parse trees (or derivations) with this grammar,

      This is ambiguous for every string 0^n2^n, such as 0022:

S => 0A2	S => BC
=> 0022	=> 0BC
	=> 00C
	=> 00C2
	=> 0022

4. Convert the following grammar to Chomsky Normal Form:

A => 0A2| BC

B => 0B2 | C | Ɛ

C => 1A1 | 1

Step 1: B is nullable, so we modify all rules containing B:

A => 0A2 | BC | C

B => 0B2 | 02 | C

C => 1A1 | 1

Step 2: All symbols are reachable and generating.

Step 3: Remove unit rules

A => 0A2 | BC | 1A1 | 1

B => 0B2 | 02 | 1A1 | 1

C => 1A1 | 1

Step 4: Break down to binary rules

A => ZA_1 | BC | NA_2 | 1

A_1 => AT

A_2 => AN

Z => 0

N => 1

T => 2

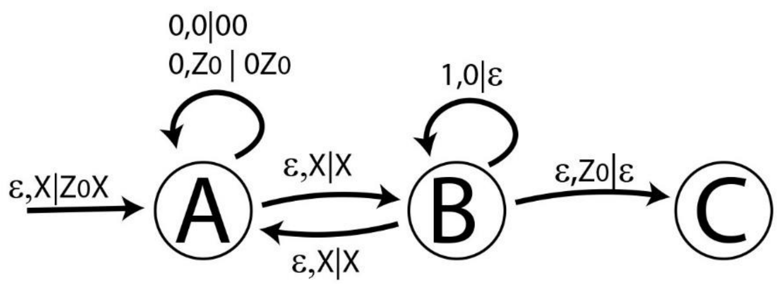
B => ZB_1 | ZT | NA_2 | 1

C => NA_2 | 1

5. Give a careful pumping lemma proof that $\{1^n2^n3^m \mid n>0, m>n\}$ is not a context free language. If you aren't clear about the language, it is the subset of 1*2*3* with the same number of 1s and 2s, and more 3s.

Suppose this language is context-free. Let p be its pumping constant. Let z be the string $1^p2^p3^{p+1}$, which is certainly longer than p. Consider any decomposition z=uvwxy, where |vwx| <= p and v and x aren't both empty. Since |vwx| <= p, v and x can together contain at most 2 of the 3 digits. If v and x contain no 3s, then they must contain either 1s or 2s (or both), so uv^2wx^2y does not have more 3s than 1 and 2s. If v and x do contain at least one 3, then v and x contain no 1s, so uv^0wx^0y contains the same number of 1s as z and fewer 3s than z, which means that uv^0wx^0y does not have more 3s than 1s. Either way we have found a value of n for which uv^nwx^ny is not in the language. This contradicts the pumping lemma, so the language must not be context-free.

6. In class we developed an algorithm by Noam Chomsky that constructs a grammar equivalent to a given PDA. Apply this algorithm to the following PDA and give the derivation in this grammar of the string 001011. Note that the PDA accepts by empty stack.



S => [AZ_0C]
=> 0[A0B] [BZ_0C]
=> 00[A0B][B0B] [BZ_0C]
=> 00[B0B][B0B] [BZ_0C]
=> 001[B0B] [BZ_0C]
=> 001[A0B] [BZ_0C]
=> 0010[A0B][BOB] [BZ_0C]
=> 0010[B0B][BOB] [BZ_0C]
=> 00101[B0B] [BZ_0C]
=> 001011[BZ_0C]
=> 001011

Topics For Exam 2

1. Grammars, derivations and parsing. We talked a little about context-sensitive grammars but the exam is only over context-free grammars. We talked about how to design grammars to determine the precedence and associativity of operators and those things will not be on the exam. I do expect you to be able to write a grammar from the description of a language, to derive a string from a grammar or to construct the parse tree for a string derived from a grammar.

2. PDAs. You should be able to construct them.

3. PDAs are equivalent to grammars. Given a grammar you should be able to construct and use an equivalent PDA. Given a PDA you should be able to produce a derivation of any string accepted by the PDA from the grammar we constructed that is equivalent to the PDA. In other words, I am unlikely to ask for the full grammar, but you should be able to derive strings in it.

4. You should be able to convert a grammar to Chomsky Normal Form and know why we do that.

5. You should be able to use the Pumping Lemma to show that a language is not context free.

6. To show that a language is context free find either a grammar or a PDA for it.

7. You should know the closure properties of context-free languages (unions? intersections? complements?)

Show that the set of Turing Machines that accept only a finite number of strings is not Recursively Enumerable.

Name________________________

CS 383
Final Exam
December 13, 2017

The exam has 8 questions. #7 is worth 16 points; the other seven are worth 12 points each

1. Identify the seven languages below as
   - R = regular
   - C = context-free but not regular
   - D = recursive (decidable) but not context-free
   - E = recursively enumerable but not recursive
   - N = not recursively enumerable

   You don't need to justify your answers.

   a. $\{ (01)^n \mid n >= 0\}$ For example, 010101 is in this language

   b. $\{(0^n1)^n \mid n>0\}$ For example, 000100010001 is in this language.

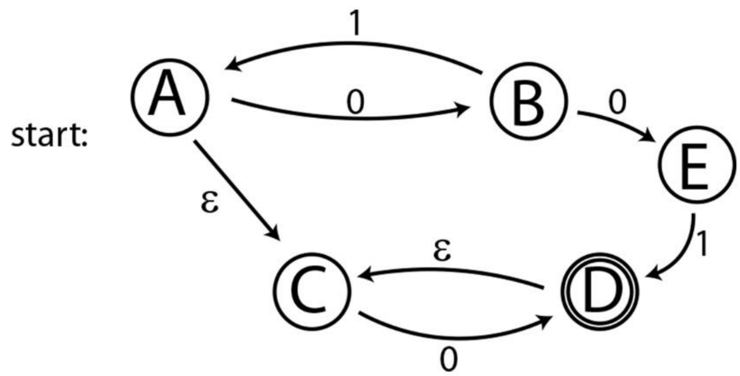
   c. Strings of the digits 0 thru 9 where no digit appears more than two times.

   d. Strings of the form *ww*, where *w* is a string of 0s and 1s.

   e. {m | m is a valid encoding of a Turing Machine} (Remember that we encoded a transition $\delta(q_i,t_j)=(q_k,t_L,d_m)$ as $0^i10^j10^k10^L10^m1$ and encoded the TM as a sequence of transitions followed by the final state).

   f. The set of encodings of Turing Machines that *do* accept their own encodings. Don't confuse this with the diagonal language, which is the set of TMs that *don't* accept their own encodings.

2. Here is an ε-NFA with A as its start state (the label didn't position quite right).



a) Convert this to a DFA.

b) Describe in English the strings that are accepted by these automata,

3. Is the language of strings of 0s and 1s that have different numbers of 0s than 1s regular? For example 001, 0101010 and 111 are all in this language. Either prove the language is regular or prove that it isn't.

4. Consider the language $\{0^n(0^m1^m)^n \mid n > 0, m>0\}$ Just to be clear, strings in this language start with n 0s. They then have n groups, where each group consists of some number of 0s followed by the same number of 1s. For example, 000001101000111 is in this language.

   a) Give a grammar for this language.

   b) Give a parse tree for the derivation of 000001101000111 with your grammar.

5. Show that the language $\{0^n 1^m 2^n \mid n > 0, m < n\}$ is not context-free.

6. Describe a TM that takes as input a string of n 0s and halts with 2^n 0s on its tape. You can use as many tapes as you want, though the number of tapes needs to be a constant and can't depend on n. It is not necessary to give all of the machine's transitions; just break this down to simple steps that can clearly be performed by a Turing Machine.

7. Let $\mathcal{L}_{\text{hippy-dippy}}$ be the set of encodings of Turing Machines that accept all strings. Our friend Happy (actually, his encoding) is a member of $\mathcal{L}_{\text{hippy-dippy}}$. The complement of $\mathcal{L}_{\text{hippy-dippy}}$ is $\mathcal{L}_{\text{skeptical}}$, the set of Turing Machines that fail to accept at least one string. Rice's Theorem tells us that neither of these sets is Recursive. Are either of them Recursively Enumerable? You can use facts we proved in class about the Diagonal language, the Universal language, the Halting language, and the Empty and Non-Empty languages (and the complements of any of these). Anything else you use you need to prove.
    a. Either prove that $\mathcal{L}_{\text{hippy-dippy}}$ is Recursively Enumerable or prove it isn't.

    b. Either prove that $\mathcal{L}_{\text{skeptical}}$ is Recursively Enumerable or prove it isn't.

8. Explain in English what Cook's Theorem (aka the Cook-Levin Theorem) means, without using the terms $\mathcal{P}$, $\mathcal{NP}$, NP-Complete and NP-Hard

You can use this page as extra space for any problem.

Please write and sign the Honor Pledge when you have finished the exam. If you didn't take the exam with the rest of the class also write your starting and stopping times.

CS 383

HW 3

Due in class Wednesday, October 2.

This one should be typed. Several of the questions ask for either a DFA or a regular expression. It is fine to attach a page to your solutions on which you have drawn the DFAs.

1. Either prove or give an example that disproves: For any regular expressions E and F, (E+F)* =E*+F*.
2. Show that the language of strings of balanced parentheses (e.g. "((())())") is not a regular language.
3. Give a careful proof that $\{a^n b^m c^n \mid n>=0, m>=0\}$ is not regular.
4. For each of the following languages, either prove that it is regular (by giving a regular expression or DFA for it) or use the Pumping Lemma to prove that it isn't regular.
    a. The set of strings of 0's and 1's where the digits sum to 5, such as 110111 and 11111.
    b. The set of strings of 0's and 1's where the digits sum to an even number.
    c. The set of strings of 0's and 1's where the digits sum to a perfect square.
    d. The set of strings of 0's and 1's such that in every prefix the number of 0's and the number of 1's never differ by more than 1. 011001 is such a string.
5. If L is a language and a is a symbol then L/a (called the *quotient* of L and a) is the set of strings w such that wa is in L. For example, if L = {a, aab, baa} then L/a = {ε, ba}. Show that if L is regular then L/a is also regular.
6. Suppose I give you a very complicated DFA for a language over the alphabet {0,1}. Give an algorithm for determining if the language accepted by that DFA is infinite. The algorithm doesn't need to be efficient, it just needs to be able to eventually give a definite "yes" or "no" answer. *Hint*: The Pumping lemma helps. *Another hint*: It isn't sufficient to treat the DFA like a directed graph and ask if it has a cycle.

CS 383
HW 3A

Due in class on Friday, September 27

Here is a question from last fall's first exam:

> Use the pumping lemma to show carefully that the language $\{0^m1^n0^n \mid m>=2, n >=0\}$ is not regular. There are two issues here. One is what to do about those leading 0s. The other is the usual issue: how do you write a Pumping Lemma proof?

Write this up carefully, and hand it in on Friday September 27.

CS 383
HW 4 Solutions

1. Remember quotient languages from HW 3: If L is a regular language over Σ and a $\in \Sigma$ then L/a is the set of strings w such that wa is in L. Either prove or disprove the following identities:
   a. (L/a)a = L No. Use L={011, 000} (L/1)1 = {011}
   b. (La)/a = L Yes. La={wa|w∈L}. (La)/a={α|αa∈La}={α|α=w for some w in L} = L

2. Suppose L is a regular language. Show that min(L) is also regular, where min(L)= {w | w is in L but no proper prefix of w is in L}

   Consider a DFA P for L. Let P' be P with all of the transitions out of final states removed. If string w is accepted by P' then it takes P to a final state, so it is accepted by P and must be in min(L) because it doesn't pass through (enter, then leave) any final states. On the other hand, if w is in min(L) it must take P to a final state without passing through any final states, so none of the transitions it needs are removed in P', which means that w is accepted by P' as well. In other words w is in min(L) if and only if w is accepted by P', so min(L) is regular.

3. Suppose L is regular. Show that prefix(L) is also regular, where prefix(L) = {w | wx is in L for some x (including x=ε)}. prefix(L) is the set of all prefixes of all strings in L. These don't need to be proper prefixes, so L is a subset of prefix(L)

   Start with a DFA for L with a minimum number of states. If there was a state in this DFA from which it was impossible to get to a final state we could remove it and get an even smaller DFA that accepted the same language, so every state in the minimal DFA can reach a final state. Now make a new DFA identical to the minimal one only have all of the states be final. This accepts string w if and only if w is a prefix of some string in L.

4. For any language L let powers(L) = {x^n | n >= 0 and x ∈ L}. Find an example where L is regular but powers(L) is not regular.

   Let L = 0*1. powers(L)={ $(0^j1)^n$| j>=0 and n>=0}. Suppose this is regular and let p be its pumping constant. Let w = 0^p10^p1. If we had a decomposition w=xyz satisfying the Pumping Lemma, the y portion would consist of just 0's so xy^2z would be $0^{p+j}10^p1$. This does not have the form w^n for any w in L. So our string w is not pumpable and powers(L) is not regular.

5. Design a context-free grammar for {0^n1^n | n>=1}
   S => 0S1 | 01

6. Design a context-free grammar for $\{a^i b^j c^k \mid i \mathrel{!=} j\}$

   $S \Rightarrow AC$
   $A \Rightarrow A_1 \mid A_2$
   $A_1 \Rightarrow a\, A_1\, b \mid a\, A_3 \mid a$
   $A_3 \Rightarrow a\, A_3 \mid a$
   $A_2 \Rightarrow a\, A_2\, b \mid B$
   $B \Rightarrow Bb \mid b$
   $C \Rightarrow cC \mid c \mid \varepsilon$

   There may be shorter solutions but this one is straightforward. A generates strings with either more a's than b's or fewer a's than b's. A_1 generates the former, A_2 the latter.

7. Here is a context-free grammar:
   $S \Rightarrow aS \mid Sb \mid a \mid b$
   Prove by induction on the string length that no string in the language represented by this grammar has ba as a substring.

I claim that every string of length n in this language does not have ba as a substring. It is certainly true of strings of length 1. Suppose this is true for all strings of length up to n, and w is a string in the language of length n+1. Then w must be $a\alpha$ where $S \overset{*}{\Rightarrow} \alpha$ and α has length n, or w must be βb where $S \overset{*}{\Rightarrow} \beta$ and β has length n. By the induction hypothesis neither α nor β contains ba as a substring. Preceding α by a can't introduce ba as a substring; following β with b can't introduce ba. So w also can't contain ba as a substring. So if strings of length up to n have no substring ba neither do strings of length n+1. By induction no strings in the language have ba as a substring.

CS 383
HW 4

Due in class Friday, October 11 though it would help you to do the first four problems before taking Exam 1

This one should be typed.

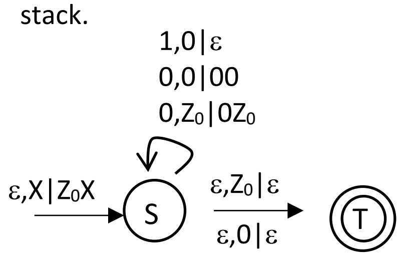
1. Remember quotient languages from HW 3: If L is a regular language over Σ and $a \in \Sigma$ then L/a is the set of strings w such that wa is in L. Either prove or disprove the following identities:
   a. (L/a)a = L
   b. (La)/a = L
2. Suppose L is a regular language. Show that min(L) is also regular, where min(L)= {w | w is in L but no proper prefix of w is in L}
3. Suppose L is regular. Show that prefix(L) is also regular, where prefix(L) = {w | wx is in L for some x (including x=ε)}. prefix(L) is the set of all prefixes of all strings in L. These don't need to be proper prefixes, so L is a subset of prefix(L)
4. For any language L let powers(L) = {x^n | n >= 0 and $x \in$ L}. Find an example where L is regular but powers(L) is not regular.
5. Design a context-free grammar for {0^n1^n | n>=1}
6. Design a context-free grammar for {$a^ib^jc^k$ | i != j }
7. Here is a context-free grammar:

   S => aS | Sb | a | b

   Prove by induction on the string length that no string in the language represented by this grammar has ba as a substring.

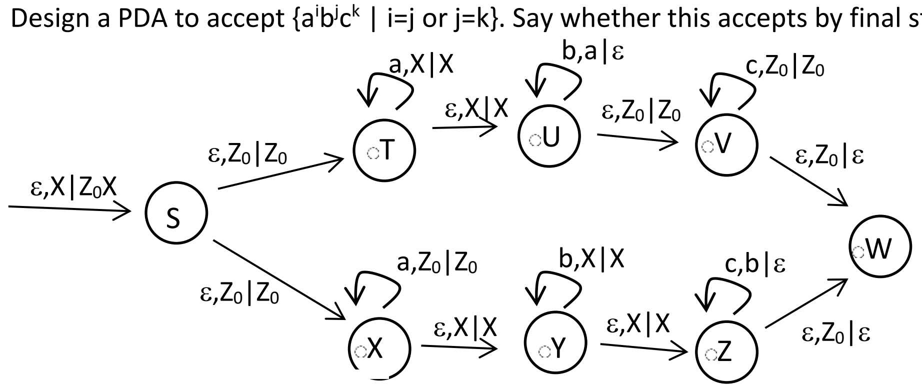
CS 383
HW 5 Solutions

1. Design a PDA to accept the strings in (0+1)* such that no prefix has more 1's than 0's. 01001011001 is a string in this language. Say whether your PDA accepts by final state or empty stack.



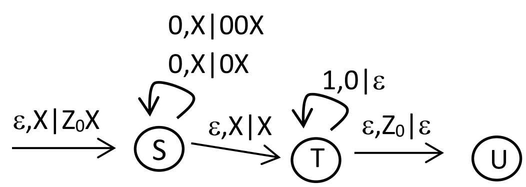
Accepts by final state.

2. Design a PDA to accept $\{a^ib^jc^k \mid i=j \text{ or } j=k\}$. Say whether this accepts by final state or empty stack.



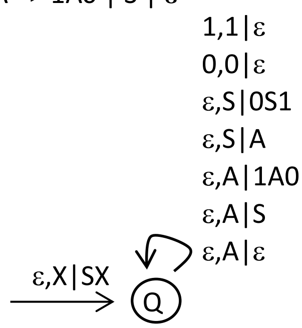
Accepts by empty stack

3. Design a PDA to accept $\{0^n1^m \mid n <= m <= 2n\}$

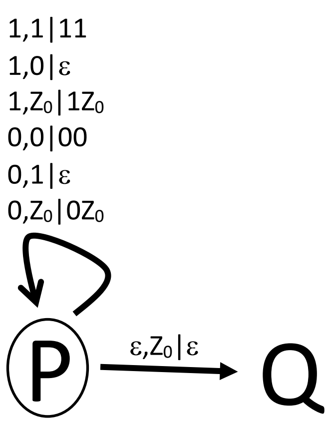


Accepts by empty stack

4. Convert the following grammar into a PDA that accepts by empty stack.
   S => 0S1 | A
   A => 1A0 | S | ε



5. Here is a PDA that accepts strings in (0+1)* with the same number of 0's and 1's. This PDA accepts by empty stack. Chomsky's algorithm gives a grammar equivalent to this PDA, with grammar symbols of the form [pXq]. Give a derivation in this grammar for the string 0101.



Derivation:
Start => [PZ_0Q]
=> 0[P0P][PZ_0Q]
=> 01[PZ_0Q]
=> 010[P0P][PZ_0Q]
=> 0101[PZ_0Q]
=> 0101

CS 383
HW 5

Due in class on Friday, November 1

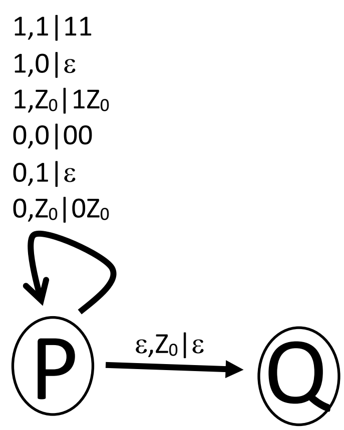
1. Design a PDA to accept the strings in (0+1)* such that no prefix has more 1's than 0's. 01001011001 is a string in this language. Say whether your PDA accepts by final state or empty stack.

2. Design a PDA to accept $\{a^i b^j c^k \mid i=j \text{ or } j=k\}$. Say whether this accepts by final state or empty stack.

3. Design a PDA to accept $\{0^n 1^m \mid n <= m <= 2n\}$

4. Convert the following grammar into a PDA that accepts by empty stack.
   S => 0S1 | A
   A => 1A0 | S | ε

5. Here is a PDA that accepts strings in (0+1)* with the same number of 0's and 1's. This PDA accepts by empty stack. Chomsky's algorithm gives a grammar equivalent to this PDA, with grammar symbols of the form [pXq]. Give a derivation in this grammar for the string 0101.



CS 383

HW 6 Solutions

1. Convert the following grammar into Chomsky Normal Form:
   S => ASB | ε
   A => aAS | a
   B => SbS | A | bb

   If you remove the ε-rule and the one unit rule (B => A), the rest is just splitting the rules into 2 nonterminals on the right hand side:

   $S \Rightarrow S_1B \mid AB$
   $S_1 \Rightarrow AS$
   $A \Rightarrow A_1S \mid A_2A \mid a$
   $A_1 \Rightarrow A_2A$
   $A_2 \Rightarrow a$
   $B \Rightarrow B_1S \mid B_2S \mid SB_2 \mid A_1S \mid A_2S \mid a \mid B_2B_2$
   $B_1 \Rightarrow SB_2$
   $B_2 \Rightarrow b$

2. Chomsky Normal Form forces parse trees to be binary trees. Some people like trinary trees. Say that a grammar is in "Bobsky Normal Form" if all rules have the form A => BCD or A => a. Can all context free grammars be converted to Bobsky Normal Form? Either find a grammar that can't be converted or prove that all can.

   No, Bobsky Normal Form is stupid. A => aa can't be converted to Bobsky Normal Form. For that matter, any string derived from a Bobsky Normal Form grammar must have odd length.

3. Show that $\{0^i 1^j 2^k \mid i < j < k\}$ is not context-free

   Suppose it is context-free; let p be its pumping constant. Let z = $0^p 1^{p+1} 2^{p+2}$. Let z=uvwxy be any pumping decomposition of z with |vwx| <= p. vwx can't contain both 0s and 2s because they are separated by p+1 symbols. If vwx contains any 0s then uv^3wx^3y has more 0s than 2s. If vwx has 1s but no 0s then uv^0wx^0y does not have more 1s than 0s. So vwx can't have any 0s or any 1s. If it is all 2s then uv^0wx^0y does not have more 2's than 1's. Any way you try, the string z is not pumpable.

4. For each of the following languages either prove that the language is context-free or prove that it isn't:
   a. $\{0^n 1^m \mid n, m > 0\}$
   b. $\{0^n 1^m \mid n > 0, m=n\}$
   c. $\{0^n 1^m \mid n > 0, 0 < m < 2n\}$
   d. $\{0^n 1^m 2^n \mid n, m > 0\}$
   e. $\{0^n 1^m 2^n\ n, m > 0, 0 < m < n\}$

   (a) is regular (and so context-free): 0^+1^+
   (b) is context free: S => 0S1 | 01
   (c) is context-free: make a PDA that on seeing a 0 nondeterministically pushes 2 0s onto the stack. When it sees a 1 transitions to a state that pops a 0 every time it sees a 1. If there are still 0s on the stack at the end of the input it accepts.
   (d) is context-free: make a PDA that reads 0s and pushes them onto the stack. When it sees a 1 it transitions to a state that reads and ignores 1, not changing the stack. When it sees a 2 it pops the stack and transitions to a state that reads 2s and pops the stack. The stack should be empty at the end of the input.
   (e) is not context-free. If it was let p be its pumping constant, let $z=0^{p+1}1^p2^{p+1}$. A pumping decomposition z=uvwxzy can't have vwx containing both 0s and 2s. If it contains 0s and not 2s or 2s and not 0s then after pumping we get different numbers of 0s and 2s. If it contains only 1s then after pumping we get more 1s than 0s or 2s. Any way we slice it, z can't be pumped.

5. Give an algorithm for determining if the language derived from a given context-free grammar is infinite.

   Here is how I would go about this. You can give a different algorithm based on grammars (by checking if it is possible to have a repeated symbol in a grammar derivatiuon). Let p be the language's pumping constant ($2^{\text{# nonterminals in a CNF grammar}}$) If the language contains a string longer than p that string is pumpable and the language is infinite. If the language has a long string and we pump it 0 times we remove at most p symbols from it, so the language is infinite if and only if it contains a string w of length between p and 2p. So generate all of the strings with length between p and 2p and check each for membership in the language. This might take a while, but it will eventually terminate.

6. Give an algorithm for determining if the language derived from a context-free grammar G is empty (i.e., the grammar derives no strings).

   In class we gave an algorithm for marking the nonterminal symbols that generate terminal strings (If A => w is a rule where w consists only of terminal symbols, mark A. If A=>α is a rule where every symbol in a is either terminal or marked, mark A. Repeat until nothing else can be marked) The language is empty if and only if the start symbol for the grammar is not marked.

CS 383
HW 6

Due in class on Friday, November 15.

This one should be typed.

1. Convert the following grammar into Chomsky Normal Form:
   S => ASB | ε
   A => aAS | a
   B => SbS | A | bb

2. Chomsky Normal Form forces parse trees to be binary trees. Some people like trinary trees. Say that a grammar is in "Bobsky Normal Form" (BNF) if all rules have the form A => BCD or A => a, where A,B,C, and D are grammar (nonp-terminal) symbols and a is a terminal symvol (i.e., a is in Σ). Can all context free grammars be converted to Bobsky Normal Form? Either find a grammar that can't be converted or prove that all can.

3. Show that $\{0^i1^j2^k \mid i < j < k\}$ is not context-free

4. For each of the following languages either prove that the language is context-free or prove that it isn't:
   a. $\{0^n1^m \mid n, m > 0\}$
   b. $\{0^n1^m \mid n > 0, m=n\}$
   c. $\{0^n1^m \mid n > 0, 0 < m < 2n\}$
   d. $\{0^n1^m2^n \mid n, m > 0\}$
   e. $\{0^n1^m2^n\ n, m > 0, 0 < m < n\}$

5. Give an algorithm for determining if the language derived from a given context-free grammar is infinite. Your algorithm must terminate for every context free grammar.

6. Give an algorithm for determining if the language derived from a context-free grammar G is empty (i.e., the grammar derives no strings).

CS 383
HW 7

Due in class Wednesday, December 4.

This doesn't need to be typed but it must be legible.

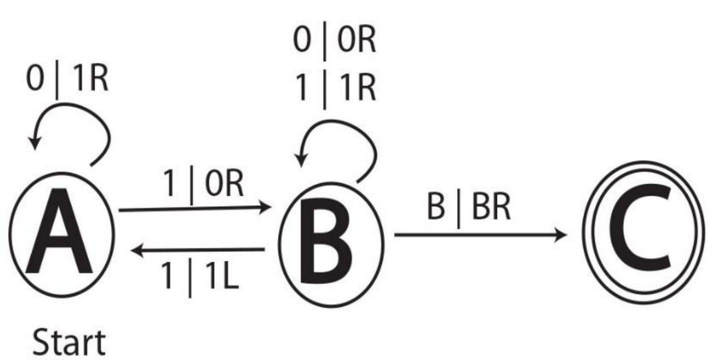
1. Design a TM to accept $\{ww^{rev} \mid w \in (0+1)^*\}$ (i.e., even-length palindromes) For this one draw out the TM as a complete state-transition diagram.
   ==
   For the remaining questions give a step-by-step description in English of the TM, but it is not necessary to draw out the state-transition diagram. For example, one step with question (1) might be "If the first letter of the input is 0, overwrite it with a blank and move right until you find a blank, then move one step left…."

2. Design a Turing Machine to accept the strings that have the same number of 0's and 1's, such as 000111 and 010101.

3. Design a TM to accept $\{ww \mid w \in (0+1)^*\}$ You might find non-determinism helpful.

4. Design a TM that starts with the binary code for a number N on its tape and ends with the code for N+1. So if it starts with 10011 it ends with 10100 and if it starts with 1111 it ends with 10000.

5. Here is a non-deterministic TM. Find all configurations that can be derived from A011



CS 383
HW 8 Solutions

1. Alan Turing was interested In modeling computations rather than accepting/rejecting inputs. His TMs had not Accept state. Given an input they either halted (which is good) or ran forever. So let $\mathcal{L}_{halt}$ = {(M,w) | M is a TM that halts (whether or not in a final state) on input w} If you prefer you can write this as {m111w | m is the encoding of a TM that halts on input w}. Show that $\mathcal{L}_{halt}$ is recursively enumerable but not recursive.

   To see that $\mathcal{L}_{halt}$ is recursively enumerable, let M* be a TM that takes as input (M,w) and simulates the computation of M on w. If M ever enters a final state M* accepts (M,w). If M ever enters a situation where it is not in a final state and it has no transition fitting its current state and tape symbol (i.e. M halts not in a final state) the M* accepts (M,w). Otherwise M* keeps running. M* accepts the language $\mathcal{L}_{halt}$.

   To see that $\mathcal{L}_{halt}$ isn't recursive we reduce the universal language to it. Remember that the universal language is {(M,w) | M accepts w} Suppose we had a decider for $\mathcal{L}_{halt}$. Given an (M,w) pair ask if M halts on w; if not reject (M,w). If M halts on w run a simulator of M on w to see what state it halts in. If it halts in a final state accept (M,w) otherwise reject (M,w). This makes a decider for the universal language, which we know can't be.

2. We showed that if a language and its complement are both RE then both are recursive. Suppose we have 3 recursively enumerable languages that are disjoint (no string is in two of them) and whose union is the set of all strings. Show that all three must be recursive.

   One way to do this is to say that the union of two recursively enumerable languages is recursively enumerable, so the complement of each language, which is the union of the other two, must be RE. We know that if both a language and its complements are RE then the language must be recursive.

   Here's a more constructive solution. We start with languages L1, L2, and L3 and TMs M1, M2, and M3 that accept them. Build a 4-tape TM M* with input w on Tape 1, and tapes 2,3,and 4 simulating the tapes of M2, M3, and M4. The states of M* will be triples consisting of a state of M1, a state of M2 and a state of M3. Given an input M, M* runs simultaneous simulations of the computations of M1, M2, and M3 on w. If any of the TMs enter a final state M* halts. Every string is in one of the three languages so M* always halts. We could build 3 versions of M*. Version 1 accepts w if M* halts in and accept state for M1, and rejects w if M* halts in an accept state for M2 or M3. This is a decider for L1. The other two versions give deciders for L2 and L3.

3. Suppose $\mathcal{L}_1$ and $\mathcal{L}_2$ are both recursively enumerable. Is the concatenation $\mathcal{L}_1\mathcal{L}_2$ RE? Why or why not?

   Yes. Suppose $\mathcal{L}_1$ and $\mathcal{L}_2$ are accepted by M1 and M2. Given a string w with |w|=n, we need to run n+1 simultaneous pairs of simulations: (ε on M1 and w on M2), (w[0] on M1 and w[1:] on M2), (w[0:2] on M1 and w[2:] on M2), etc. If any pair of simulations ever halts and both computations accept, our simulator halts and accepts w. Alternatively, use a nondetermi

4. We know $\mathcal{L}_{ne}$ is recursively enumerable but not recursive. Let $\mathcal{L}_{2ne}$ be {m | m encodes TM that accepts at least 2 strings} Rice's Theorem says $\mathcal{L}_{2ne}$ is not recursive. Is it recursively enumerable? Why or why not?

   Sure. Let N be a non-deterministic TM that, given an encoding m, writes two arbitrary strings after it, so we have (m,w1,w2). N then simulates the computation of m on w1 and the computation of m on w2. If both halt and accept N enters an accept state. This non-deterministic TM accepts $\mathcal{L}_{2ne}$ and we know that any language accepted by a non-deterministic TM is also accepted by a deterministic TM.

5. Let $\mathcal{L}_{inf}$ be {m | m encodes a TM that accepts infinitely many strings}. Is $\mathcal{L}_{inf}$ RE?

   I thought we should go out on a hard problem. Neither $\mathcal{L}_{inf}$ nor its complement are RE. To show $\mathcal{L}_{inf}$ is not RE we reduce the complement of $\mathcal{L}_{halt}$ (which we know from Q1 isn't RE) to it. Given an (M,w) pair build M': For input x, M' simulates the computation of M on w for |x| steps. If M is still running (i.e, after |x| steps M is at a state-tape combination for which there is a valid transition) then M' accepts x.

   If M halts on w after N steps, then M' does not accept any x with |x| > N. This means the language accepted by M' is finite.
   If M does not halt on w then M' accepts all strings, so the language accepted by M' is infinite.

   A recognizer for $\mathcal{L}_{inf}$ would allow us to recognize if M does not halt on w, so that would accept the complement of the halting language.

   You can construct almost the same argument around the complement of the "universal language" {(M,w) | M accepts w} We showed the universal language is RE but not recursive, so its complement isn't RE.

6. Let $\mathcal{L}_{\text{hippy-dippy}}$ be the set of encodings of Turing Machines that accept all strings. Our friend Happy (actually, his encoding) is a member of $\mathcal{L}_{\text{hippy-dippy}}$. The complement of $\mathcal{L}_{\text{hippy-dippy}}$ is $\mathcal{L}_{\text{skeptical}}$, the set of Turing Machines that fail to accept at least one string. Rice's Theorem tells us that neither of these sets is Recursive.
    a. Prove that $\mathcal{L}_{\text{hippy-dippy}}$ is not Recursively Enumerable. You might try reducing the complement of the halting langue to $\mathcal{L}_{\text{hippy-dippy}}$.
    b. Either prove that $\mathcal{L}_{\text{skeptical}}$ is Recursively Enumerable or prove it isn't.

Neither of these is recursively enumerable.

a) We can reduce the complement of the halting language to $\mathcal{L}_{\text{hippy-dippy}}$. to see this, start with an (M, w) pair. Create a new Turing Machine M' that runs on input x as follows: M' simulates M on w for |x| steps. If M halts on w within |x| steps, M' rejects x; otherwise M' accepts x. M does not halt on w exactly when M' is in $\mathcal{L}_{\text{hippy-dippy}}$. So a TM that recognizes the latter would recognize the complement of the halting language, and we know the latter is not recursively enumerable.

b) This one is easier. We reduce the complement of the universal language to this. Given an (M, w) pair create a new Turing Machine M'. For any input s, M' accepts s if s is not w, and M' simulates M on w if s is w. M' accepts all strings if an only if M accepts w. So if we could recognize if M' is in $\mathcal{L}_{\text{skeptical}}$ then we can recognize that (M, w) is in the complement of the universal language, which we know we can't do.

CS 383
HW 8

Due on the last day of classes: Thursday, December 12.

1. Alan Turing was interested in modeling computations rather than accepting/rejecting inputs. His TMs had no Accept state. Given an input they either halted (which is good) or ran forever. So let $\mathcal{L}_{halt}$ = {(M,w) | M is a TM that halts (whether or not in a final state) on input w} If you prefer you can write this as {m1111w | m is the encoding of a TM that halts on input w}. Show that $\mathcal{L}_{halt}$ is recursively enumerable but not recursive.

2. We showed that if a language and its complement are both RE then both are recursive. Suppose we have 3 recursively enumerable languages that are disjoint (no string is in two of them) and whose union is the set of all strings. Show that all three must be recursive.

3. Suppose $\mathcal{L}_1$ and $\mathcal{L}_2$ are both recursively enumerable. Is the concatenation $\mathcal{L}_1\mathcal{L}_2$ RE? Why or why not?

4. We know $\mathcal{L}_{ne}$ is recursively enumerable but not recursive. Let $\mathcal{L}_{2ne}$ be {m | m encodes a TM that accepts at least 2 strings} Rice's Theorem says $\mathcal{L}_{2ne}$ is not recursive. Is it recursively enumerable? Why or why not?

5. Let $\mathcal{L}_{inf}$ be {m | m encodes a TM that accepts infinitely many strings}. Is $\mathcal{L}_{inf}$ RE?

6. Let $\mathcal{L}_{hippy\text{-}dippy}$ be the set of encodings of Turing Machines that accept all strings. Our friend Happy (actually, his encoding) is a member of $\mathcal{L}_{hippy\text{-}dippy}$. The complement of $\mathcal{L}_{hippy\text{-}dippy}$ is $\mathcal{L}_{skeptical}$, the set of Turing Machines for which there is at least one string the machine fails to accept. Rice's Theorem tells us that neither of these sets is Recursive.
   a. Prove that $\mathcal{L}_{hippy\text{-}dippy}$ is not Recursively Enumerable. You might try reducing the complement of the halting language from Question 1 to $\mathcal{L}_{hippy\text{-}dippy}$.
   b. Either prove that $\mathcal{L}_{skeptical}$ is Recursively Enumerable or prove it isn't.

CS 383
HW 1

due in class Wednesday, September 18

You can do it by hand as long as you do it neatly.

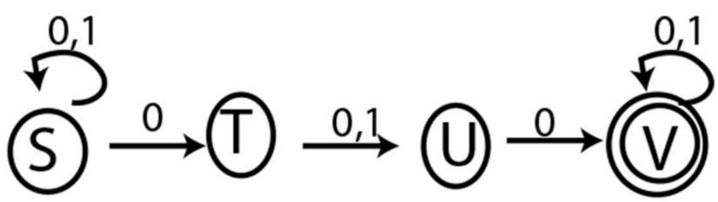
1. Give DFAs accepting the following languages over the alphabet {0,1}:
   a. The set of strings ending in 000.
   b. The set of strings containing 000 as a substring.
   c. The set of strings containing exactly three 0's.

2. Give an NFA that accepts the set of strings over the alphabet {0,1,2,3} such that the final digit in the string *has* appeared before.

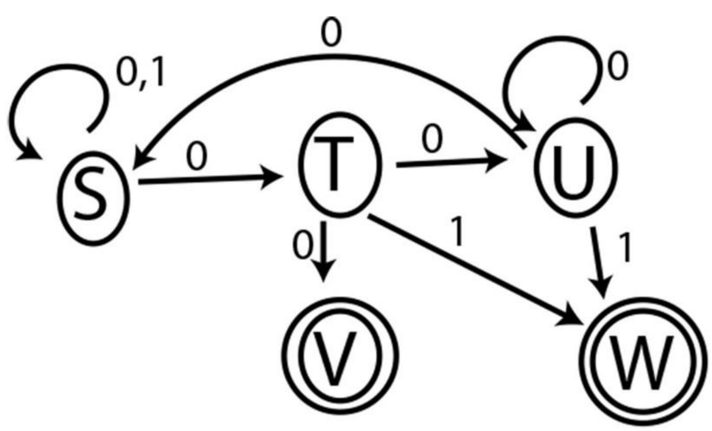
3. Give an NFA that accepts the set of strings over the alphabet {0,1,2,3} such that the final digit in the string *has not* appeared before.

4. In Java an identifier (name) must be composed of letters, digits, underscore and dollar sign and can't begin with a digit. Give a DFA that accepts the valid identifiers. You can use symbols <L> and <D> to represent "any letter" and "any digit" respectively.

5. Convert the following NFA to a DFA and describe in English what strings it accepts:



6. Convert the following NFA to a DFA and describe in English what strings it accepts:

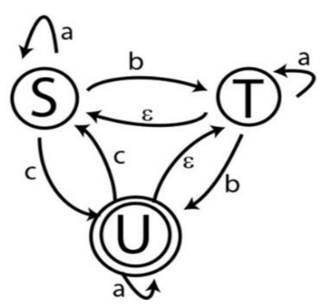


CS 383
HW 2

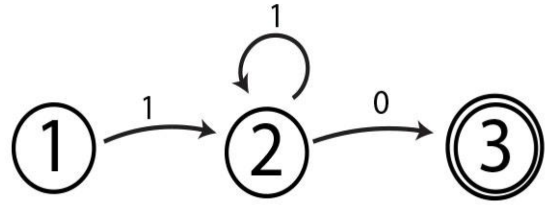
Due in class Wednesday, September 25

Again, you can do this one on paper if you do it neatly and legibly.

1. Here is an ε-NFA. Convert it to a DFA and find all of the strings of length 2 accepted by it. S is the start state.



2. Design an ε-NFA for the set of strings consisting of either 01 repeated 1 or more times or 010 repeated 1 or more times.
3. Give a regular expression for the set of strings over the alphabet {a,b,c} containing at least one a and at least one b.
4. Give a DFA for the set of strings with an even number of zeros.
5. Give a regular expression for the set of strings with an even number of zeros.
6. Describe in English the language denoted by the regular expression $(1+\varepsilon)(00^{*}1)^{*}0^{*}$
7. Suppose we have a finite automaton with no transitions into the start state and none out of the final state. This automaton accepts language $\mathcal{L}$. If we modify the automaton by adding an ε-transition from the final state to the start state, what language will it accept?
8. Convert the regular expression (0+1)(01)* into an ε-NFA using the construction we developed in class.
9. Convert $(1+\varepsilon)(00^{*}1)^{*}0^{*}$ into an ε-NFA any way you wish.
10. Convert the following DFA into a regular expression using the construction we developed in class. S is the start state.



Midterm Exam I
CIS 341: Foundations of Computer Science II — **Fall 2005, day section**
Prof. Marvin K. Nakayama

Print family (or last) name: ______________________________

Print given (or first) name: ______________________________

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

____
Signature and Date

- This exam has 6 pages in total, numbered 1 to 6. Make sure your exam has all the pages.
- This exam will be 1 hour and 25 minutes in length.
- This is a closed-book, closed-note exam.
- For all problems, follow these instructions:
  1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
  2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step.

Problem	1	2	3	4	5	Total
Points						

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — If A is a nonregular language and B is a language such that $B \subseteq A$, then B must be nonregular.

(b) TRUE FALSE — If A is a regular language and B is a language such that $B \subseteq A$, then B must be regular.

(c) TRUE FALSE — A regular expression for the language $\{ a^n b^n \mid n \geq 0 \}$ is $\varepsilon \cup ab \cup aabb \cup aaabbb \cup \cdots$.

(d) TRUE FALSE — An NFA may have no accept states.

(e) TRUE FALSE — The regular expressions $(0^*1^*)^*$ and $(0 \cup 1)^*$ generate the same language.

(f) TRUE FALSE — If A is a language recognized by an NFA and B is the complement of a language having a regular expression, then $\overline{(A \circ B)}$ is regular.

(g) TRUE FALSE — The class of regular languages is closed under union.

(h) TRUE FALSE — If A is a regular language, then A is finite.

(i) TRUE FALSE — A DFA accepts ε if and only if the start state is also an accept state.

(j) TRUE FALSE — Every DFA is also an NFA.

2. **[20 points]** Give definitions or meanings of the following terms and phrases. Each answer should be at most two sentences. Be sure to define any notation that you use.

   (a) The complement of a language A defined over alphabet Σ.

   (b) The transition function δ of an NFA.

   (c) Nonregular language.

   (d) The class of regular languages is closed under intersection.

3. **[20 points]** Let A be the language over the alphabet $\Sigma = \{a, b\}$ defined by regular expression $((ab)^* b \cup aa)^*$. Give an NFA that recognizes A.

Give NFA for A here.

Scratch-work area

4. **[20 points]** Let $\Sigma = \{0, 1\}$. A string over Σ is said to contain a double symbol if it contains either 00 or 11 as a substring. Give a regular expression for each of the languages below.

   (a) $A = \{ w \in \Sigma^* \mid w \text{ does not end in a double symbol} \}$.

   Answer:

   (b) $B = \{ w \in \Sigma^* \mid w \text{ contains exactly one double symbol} \}$. (The string 10010 has exactly one double symbol, but 100010 has two double symbols.)

   Answer:

Scratch-work area

5. **[20 points]** Recall the pumping lemma:

Theorem: If A is a regular language, then $\exists$ number p (pumping length) where, if $s \in A$ with $|s| \geq p$, then $\exists$ strings x, y, z such that $s = xyz$ and

(i) $xy^i z \in A$ for each $i \geq 0$,

(ii) $|y| > 0$, and

(iii) $|xy| \leq p$.

Prove that $C = \{ ww \mid w \in \Sigma^* \}$ is a nonregular language, where $\Sigma = \{0, 1\}$.

Midterm Exam II
CIS 341: Foundations of Computer Science II — **Fall 2005, day section**
Prof. Marvin K. Nakayama

Print family (or last) name: ____

Print given (or first) name: ____

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

Signature and Date: ____

- This exam has 6 pages in total, numbered 1 to 6. Make sure your exam has all the pages.
- This exam will be 1 hour and 25 minutes in length.
- This is a closed-book, closed-note exam.
- For all problems, follow these instructions:
    1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
    2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton; CFG stands for context-free grammar; PDA stands for pushdown automaton.
    3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step. Unless you are specifically asked to prove a theorem from the book, you may assume that the theorems in the textbook hold; i.e., you do not have to reprove the theorems in the textbook. When using a theorem from the textbook, make sure you provide enough detail so that it is clear which result you are using; e.g., say something like, "By the theorem that shows every NFA has an equivalent DFA, it follows that . . . "

Problem	1	2	3	4	5	Total
Points						

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

   (a) TRUE FALSE — If language A has a CFG, then A has a regular expression.

   (b) TRUE FALSE — If language A has a regular expression, then A has a CFG.

   (c) TRUE FALSE — A Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ will either accept or reject any string $w \in \Sigma^*$.

   (d) TRUE FALSE — There is a language A that is recognized by a 3-tape Turing machine but is not recognized by any 1-tape Turing machine.

   (e) TRUE FALSE — Every Turing-recognizable language is also Turing-decidable.

   (f) TRUE FALSE — If language A has a PDA, then A has a CFG in Chomsky normal form.

   (g) TRUE FALSE — The language $A = \{ ww \mid w \in \Sigma^* \}$ with $\Sigma = \{0, 1\}$ is context-free.

   (h) TRUE FALSE — If a language is Turing-decidable, then it is also context-free.

   (i) TRUE FALSE — There is an algorithm to determine if a regular expression R generates a string w.

   (j) TRUE FALSE — Every nondeterministic Turing machine has an equivalent deterministic Turing machine.

2. **[24 points]** Give a short answer (at most two sentences) for each part below. No proofs are required, but be sure to define any notation that you use.

   (a) What is a configuration of a Turing machine?

   (b) Give the language corresponding to the DFA acceptance problem.

   (c) Give a CFG for the language $L = \{ b^n a^n \mid n \geq 1 \}$.

   (d) What is the Church-Turing Thesis?

3. [**10 points**] Show that the collection of Turing-decidable languages is closed under union.

4. **[26 points]** Consider the following CFG $G = (V, \Sigma, R, S)$, where $V = \{S, T, X\}$, $\Sigma = \{a, b\}$, the start variable is S, and the rules R are

$$
\begin{aligned}
S &\rightarrow TX \\
T &\rightarrow TSS \mid a \\
X &\rightarrow b \mid \varepsilon
\end{aligned}
$$

Give a PDA that recognizes the language $L(G)$.

Give PDA here.

Scratch-work area

5. **[20 points]** Let $A\varepsilon_{\text{CFG}} = \{ \langle G \rangle \mid G \text{ is a CFG that generates } \varepsilon \}$. Show that $A\varepsilon_{\text{CFG}}$ is decidable.

Midterm Exam
CIS 341-451: Foundations of Computer Science II — **Fall 2005, eLearning section**
Prof. Marvin K. Nakayama

Print family (or last) name: ____

Print given (or first) name: ____

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

____

Signature and Date

- This exam has 8 pages in total, numbered 1 to 8. Make sure your exam has all the pages.
- The exam is to be given on Saturday, October 22, 2005, 12:30-3:00pm, EST.
- This is a closed-book, closed-note exam. No calculators are allowed.
- For all problems, follow these instructions:
    1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
    2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton; CFG stands for context-free grammar; PDA stands for pushdown automaton; TM stands for Turing machine.
    3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step.

Problem	1	2	3	4	5	6	7	Total
Points								

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — If A is recognized by an NFA, then A is a context-free language.

(b) TRUE FALSE — Every context-free language is regular.

(c) TRUE FALSE — Every context-free language is not regular.

(d) TRUE FALSE — A language L has a CFG if and only if L is recognized by a PDA.

(e) TRUE FALSE — If a language A is not context-free, then it must be infinite.

(f) TRUE FALSE — The class of Turing-decidable languages is closed under union.

(g) TRUE FALSE — If A is a nonregular language, then A is recognized by an NFA.

(h) TRUE FALSE — A regular expression for the language $\{\, 0^n 1^n \mid n \geq 0 \,\}$ is $\varepsilon \cup 01 \cup 0011 \cup 000111 \cup \cdots$.

(i) TRUE FALSE — $\emptyset$ is a context-free language.

(j) TRUE FALSE — Every DFA is also an NFA.

2. **[20 points]** Give definitions or meanings of the following terms and phrases. **Each answer should be at most two sentences. Be sure to define any notation that you use.**

   (a) The complement of a language A defined over alphabet Σ.

   (b) Chomsky normal form.

   (c) The difference between a Turing-decidable language and a Turing-recognizable language.

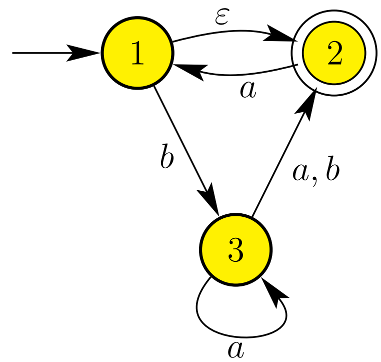
   (d) The class of context-free languages is closed under concatenation.

3. **[10 points]** Let A be the language over the alphabet $\Sigma = \{a, b\}$ defined by regular expression $(a \cup ba)^*(ab)^*$. Give an NFA that recognizes A.

Draw an NFA for A here.

Scratch-work area

4. **[10 points]** Convert the following NFA N into an equivalent DFA.



Answer:

Scratch-work area

5. **[20 points]** Consider the language $A = \{\, w \in \Sigma^* \mid w = w^{\mathcal{R}} \,\}$, where $\Sigma = \{a, b\}$ and $w^{\mathcal{R}}$ denotes the reverse of string w.

   (a) Give a context-free grammar that describes A.

   (b) Give a pushdown automaton that recognizes A.

Scratch-work area

6. [**10 points**] Show that the collection of Turing-recognizable languages is closed under union.

7. [**10 points**] Recall the pumping lemma for context-free languages:

Theorem: For every context-free language L, there exists a pumping length p such that, if $s \in L$ with $|s| \geq p$, then we can write $s = uvxyz$ with

(i) $uv^ixy^iz \in L$ for each $i \geq 0$,

(ii) $|vy| > 0$, and

(iii) $|vxy| \leq p$.

Prove that $A = \{\, a^{3n}b^nc^{2n} \mid n \geq 0 \,\}$ is not a context-free language.

Midterm Exam I
CIS 341: Foundations of Computer Science II — **Spring 2005, day section**
Prof. Marvin K. Nakayama

Print family (or last) name: ____

Print given (or first) name: ____

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

____
Signature and Date

- This exam has 6 pages in total, numbered 1 to 6. Make sure your exam has all the pages.
- This exam will be 1 hour and 25 minutes in length.
- This is a closed-book, closed-note exam.
- For all problems, follow these instructions:
  1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
  2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step.

Problem	1	2	3	4	Total
Points					

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — If A is a regular language, then $\overline{A}$ is a nonregular language.

(b) TRUE FALSE — If language A_1 is recognized by an NFA and language A_2 is defined by a regular expression, then $A_1 \cup A_2$ is recognized by some DFA.

(c) TRUE FALSE — A regular expression for the language $\{\, a^n b^n \mid n \geq 0 \,\}$ is $\varepsilon \cup ab \cup aabb \cup aaabbb \cup \cdots$.

(d) TRUE FALSE — Every DFA is also an NFA.

(e) TRUE FALSE — Every NFA is also a DFA.

(f) TRUE FALSE — A DFA may have no accept states.

(g) TRUE FALSE — The language $\{a^{2n} : n \geq 0\}$ is a nonregular language.

(h) TRUE FALSE — If A is a nonregular language, then there is an NFA that recognizes A.

(i) TRUE FALSE — If A is a nonregular language and B is a language with $B \subseteq A$, then B must be nonregular.

(j) TRUE FALSE — An NFA accepts ε if and only if the start state is also an accept state.

2. **[20 points]** Give definitions or meanings of the following terms and phrases. Each answer should be at most two sentences. Be sure to define any notation that you use.

   (a) Regular language.

   (b) The transition function δ of an NFA.

   (c) The complement of a language A over an alphabet Σ.

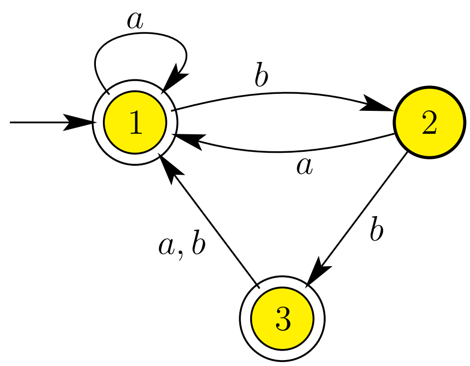
   (d) The class of regular languages is closed under Kleene-star.

3. **[20 points]** Let A be the language over the alphabet $\Sigma = \{a, b\}$ defined by regular expression $((aa)^* b \cup a)^*$. Give an NFA that recognizes A.

Give NFA for A here.

Scratch-work area

4. **[20 points]** Give a regular expression for the language recognized by the DFA below.



Regular expression: ______________________________

Scratch-work area

5. **[20 points]** Recall the pumping lemma:

Theorem: If A is a regular language, then $\exists$ number p (pumping length) where, if $s \in A$ with $|s| \geq p$, then $\exists$ strings x, y, z such that $s = xyz$ and

(i) $xy^iz \in A$ for each $i \geq 0$,

(ii) $|y| > 0$, and

(iii) $|xy| \leq p$.

Let $\Sigma = \{a, b\}$, and define language $B = \{\, w \in \Sigma^* \mid w = w^R \text{ and } |w| \text{ is even} \,\}$, where $w^{\mathcal{R}}$ denotes the reverse of the string w. Prove that B is a nonregular language.

Midterm Exam II
CIS 341: Foundations of Computer Science II — **Spring 2005, day section**
Prof. Marvin K. Nakayama

Print family (or last) name: ____________________________________

Print given (or first) name: ____________________________________

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

Signature and Date: ____________________________________

- This exam has 6 pages in total, numbered 1 to 6. Make sure your exam has all the pages.
- This exam will be 1 hour and 25 minutes in length.
- This is a closed-book, closed-note exam.
- For all problems, follow these instructions:
  1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
  2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton; CFG stands for context-free grammar; PDA stands for pushdown automaton.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step. Unless you are specifically asked to prove a theorem from the book, you may assume that the theorems in the textbook hold; i.e., you do not have to reprove the theorems in the textbook. When using a theorem from the textbook, make sure you provide enough detail so that it is clear which result you are using; e.g., say something like, "By the theorem that shows every NFA has an equivalent DFA, it follows that ..."

Problem	1	2	3	4	5	Total
Points						

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — If A has a CFG, then $\overline{A}$ has a PDA.

(b) TRUE FALSE — If language A_1 is recognized by an NFA and language A_2 has a PDA, then $A_1 \cup A_2$ has a CFG.

(c) TRUE FALSE — If A is a context-free language, then A has a CFG G in Chomsky normal form.

(d) TRUE FALSE — A Turing machine will halt on every input.

(e) TRUE FALSE — If A is a Turing-decidable language, then A is also Turing-recognizable.

(f) TRUE FALSE — Every nondeterministic Turing machine has an equivalent Turing machine.

(g) TRUE FALSE — The language $A = \{ a^n b^n c^n \mid n \geq 0 \}$ is context-free.

(h) TRUE FALSE — It is impossible to determine if a CFG generates ε.

(i) TRUE FALSE — The language

$$A_{\text{DFA}} = \{ \langle M, w \rangle \mid M \text{ is a DFA that accepts } w \}$$

is Turing-decidable.

(j) TRUE FALSE — There are some languages recognized by a 4-tape Turing machine that cannot be recognized by a 1-tape Turing machine.

2. **[20 points]** Give a short answer (at most two sentences) for each part below. No proofs are required, but be sure to define any notation that you use.

   (a) Give an example of a context-free language that is not a regular language.

   (b) Explain the difference between a Turing-decidable language and a Turing-recognizable language.

   (c) What is the Church-Turing Thesis?

   (d) What does it mean for a CFG G to be in Chomsky normal form?

3. **[20 points]** Give an implementation-level description of a Turing machine that decides the language $L = \{\, a^n b^n c^n \mid n \geq 0 \,\}$.

Scratch-work area

4. **[20 points]** Consider the following statement:

> The class of context-free languages is closed under intersection.

Is the statement TRUE or FALSE. If it is TRUE, give a proof. If it is FALSE, give a counterexample. Be sure to explain your answer in detail.

TRUE **FALSE** (circle one)

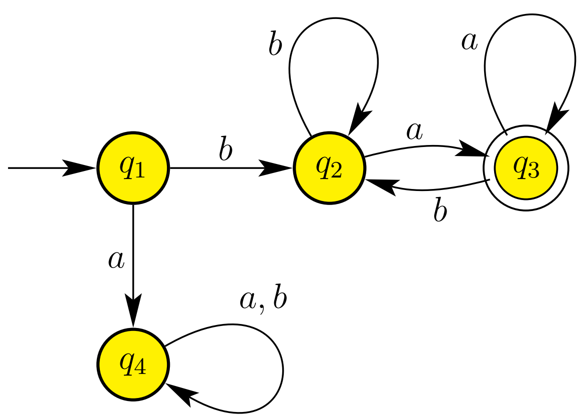
5. **[20 points]** Consider the problem of testing whether a DFA and regular expression are equivalent. Express this problem as a language and show that it is decidable.

CS 341, Fall 2006
Solutions for Midterm I

1. (a) False. The empty language $\emptyset$ has no strings at all, not even the empty string ε.
   (b) False. $\emptyset$ is a language and ε is a string, so they can't be equal.
   (c) False. $L(R \circ \emptyset) = \emptyset \neq L(R)$ if $R \neq \emptyset$.
   (d) True, by Corollary 1.40.
   (e) True, by Corollary 1.40.
   (f) False. The language $\{a, b\}^*$ is infinite and has a DFA (see lecture notes 1-18), so it is regular.
   (g) False. The string 0 begins and ends with 0 but cannot be generated by R.
   (h) True, by Homework 2, problem 5.
   (i) False. $\emptyset^* = \{\varepsilon\} \neq \emptyset$.
   (j) False. $R \circ \varepsilon = R$ by lecture notes 1-62, so $L(R \circ \varepsilon) \neq \emptyset$ when $L(R) \neq \emptyset$.

2. (a) $A \times B = \{(11, \varepsilon), (11, 1), (111, \varepsilon), (111, 1)\}$ and $A \circ B = \{11, 111, 1111\}$.
   (b) There are many sets S for which $S^* = S^+$, e.g., $S = \{\varepsilon\}$. In fact, $S^* = S^+$ if and only if $\varepsilon \in S$.
   (c) There are many sets S for which $S^* = S$, e.g., $S = \{\varepsilon\}$.
   (d) The difference between a DFA and an NFA is in the transition function δ. For a DFA, $\delta : Q \times \Sigma \rightarrow Q$. For an NFA, $\delta : Q \times \Sigma_\varepsilon \rightarrow P(Q)$, where $P(Q)$ is the power set of Q.

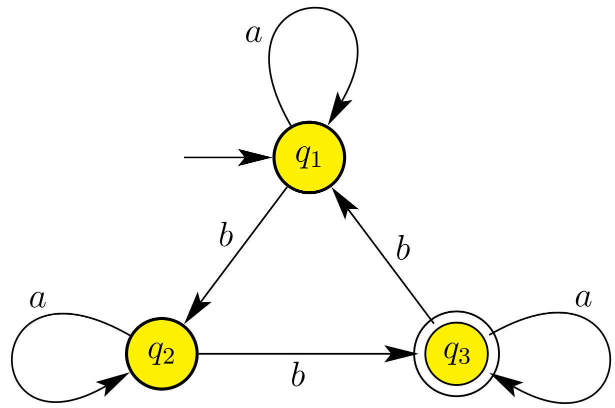
3. (a) Strings that begin with b and end with a.
   DFA



Regular expression: $b(a \cup b)^* a$

(b) Strings w such that $n_b(w) \bmod 3 = 2$

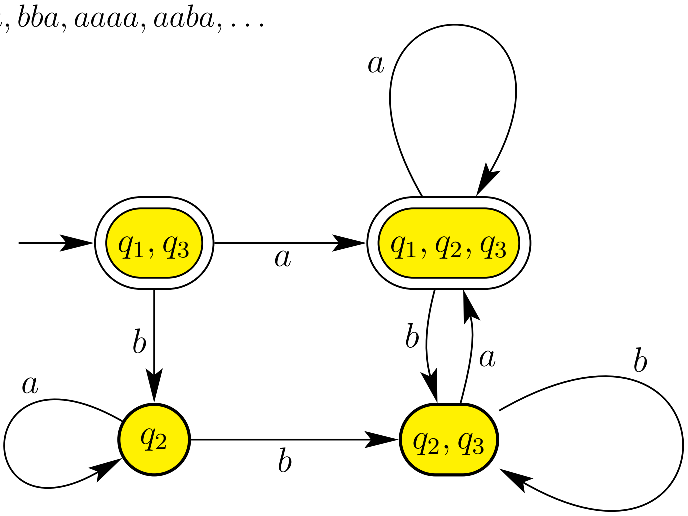
DFA



Regular expression: $a^*ba^*b(ba^*ba^*b \cup a)^*$

4. (a) $\varepsilon, a, aa, aaa, aba, bba, aaaa, aaba, \ldots$

(b)



5. We prove this by contradiction. Suppose that $\overline{M}$ is not a minimal DFA for $\overline{A}$. Then there exists another DFA D for $\overline{A}$ such that D has strictly fewer states than $\overline{M}$. Now create another DFA D' by swapping the accepting and non-accepting states of D. Then D' recognizes the complement of $\overline{A}$. But the complement of $\overline{A}$ is just A, so D' recognizes A. Note that D' has the same number of states as D, and $\overline{M}$ has the same number of states as M. Thus, since we assumed that D has strictly fewer states than $\overline{M}$, then D' has strictly fewer states than M. But since D' recognizes A, this contradicts our assumption that M is a minimal DFA for A. Therefore, $\overline{M}$ is a minimal DFA for $\overline{A}$.

Midterm Exam I
CS 341: Foundations of Computer Science II — **Fall 2006, day section**
Prof. Marvin K. Nakayama

Print family (or last) name: ______________________________

Print given (or first) name: ______________________________

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

____
Signature and Date

- This exam has 6 pages in total, numbered 1 to 6. Make sure your exam has all the pages.
- Note the number written on the upper right-hand corner of the first page. On the sign-up sheet being passed around, sign your name next to this number.
- This exam will be 1 hour and 25 minutes in length.
- This is a closed-book, closed-note exam.
- For all problems, follow these instructions:
  1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
  2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step.

Problem	1	2	3	4	5	Total
Points						

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — $\emptyset = \{\varepsilon\}$.

(b) TRUE FALSE — $\emptyset = \varepsilon$.

(c) TRUE FALSE — If R is any regular expression, then $L(R \circ \emptyset) = L(R)$.

(d) TRUE FALSE — If A is recognized by an NFA, then A is regular.

(e) TRUE FALSE — If A is a regular language, then there is an NFA that recognizes A.

(f) TRUE FALSE — If A is a regular language, then $|A| < \infty$.

(g) TRUE FALSE — If regular expression $R = 0(0 \cup 1)^*0$, then $L(R)$ is the language of all strings over $\Sigma = \{0, 1\}$ that begin and end with 0.

(h) TRUE FALSE — The class of regular languages is closed under intersection.

(i) TRUE FALSE — $\emptyset^* = \emptyset$.

(j) TRUE FALSE — If R is any regular expression, then $L(R \circ \varepsilon) = \emptyset$.

2. **[20 points]** Give short answers to each of the following parts. Each answer should be at most three sentences. Be sure to define any notation that you use.

   (a) For the sets $A = \{11, 111\}$ and $B = \{\epsilon, 1\}$, what are $A \times B$ and $A \circ B$?

   (b) Give an example of a set S such that $S^* = S^+$.

   (c) Give an example of a set S such that $S^* = S$.

   (d) Explain the difference between a DFA and an NFA.

3. **[20 points]** For each of the following languages over the alphabet $\Sigma = \{a, b\}$, give a DFA and a regular expression for it. For the DFA, you only need to draw the graph; you do not need to formally define it as a 5-tuple. Also, for any string $w \in \Sigma^*$, define $n_b(w)$ to be the number of b's in w.

   (a) All strings that begin with b and end with a.
   Draw DFA here:

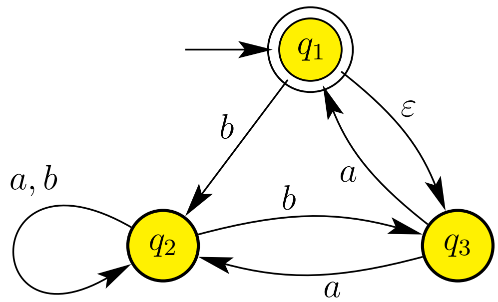
   Give regular expression here:

   (b) All strings w such that $n_b(w) \bmod 3 = 2$.
   Draw DFA here:

   Give regular expression here:

Scratch-work area

4. **[25 points]** Let N be the following NFA with $\Sigma = \{a, b\}$, and let $C = L(N)$.



(a) List the strings in C in lexicographic order. If C has more than 8 strings, list only the first 8 strings in C, followed by 3 dots.

(b) Give a DFA for C.

Scratch-work area

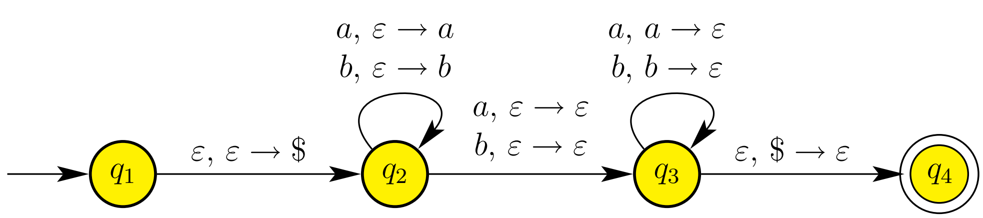
5. [**15 points**] We say that a DFA M for a language A is *minimal* if there does not exist another DFA M' for A such that M' has strictly fewer states than M. Suppose that $M = (Q, \Sigma, \delta, q_0, F)$ is a minimal DFA for A. Using M, we construct a DFA $\overline{M}$ for the complement $\overline{A}$ as $\overline{M} = (Q, \Sigma, \delta, q_0, Q - F)$. Prove that $\overline{M}$ is a minimal DFA for $\overline{A}$.

CS 341, Fall 2006
Solutions for Midterm II, Day Section

1. (a) False. Corollary 1.40.
   (b) True. Corollary 2.32.
   (c) True. Homework 5, problem 3(b).
   (d) False. All regular languages are also context-free by Corollary 2.32.
   (e) False. It is nonregular (see notes 1-90), so it cannot have a regular expression by Kleene's Theorem.
   (f) False. The language $\{a, b\}^*$ is infinite and has a DFA (see lecture notes 1-18), so it is regular.
   (g) False. Homework 6, problem 2(a).
   (h) False. A is finite, so it is regular by page 1-81 of the notes.
   (i) True. B^* is context-free by Homework 5, problem 3(c). A is context-free by Corollary 2.32, so $A \cup B^*$ is context-free by Homework 5, problem 3(a).
   (j) False. $\{a^n b^n c^n \mid n \geq 0\}$ is neither regular nor context-free.

2. (a) A CFG is in Chomsky normal form if each of its rules has one of the following 3 forms: $A \to BC$ or $A \to x$ or $S \to \varepsilon$, where A, B, C, S are variables; B and C are not the start variable S; and x is a terminal.
   (b) $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to P(Q \times \Gamma_\varepsilon)$, where Q is the set of states, Σ is the input alphabet, Γ is the stack alphabet, $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$, $\Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$, and $P(Q \times \Gamma_\varepsilon)$ denotes the power set of $Q \times \Gamma_\varepsilon$.
   (c) $G_3 = (V_3, \Sigma, R_3, S_3)$, where $V_3 = V_1 \cup V_2 \cup \{S_3\}$ and $R_3 = R_1 \cup R_2 \cup \{S_3 \to S_1 \mid S_2\}$.
   (d) See page 1-66 of notes.
   (e) See page 1-67 of notes.

3. (a) $G = (V, \Sigma, R, S)$, where $V = \{S\}$, $\Sigma = \{a, b\}$, and the rules are $S \to aSa \mid bSb \mid a \mid b$.
   (b) PDA



4. $(a \cup bb^*a)(ba \cup (a \cup bb)b^*a)^*$

5. Suppose A is context-free. Let $p \geq 1$ be the pumping length, and consider the string $s = c^p a^p b^p \in A$. Note that $|s| = 3p \geq p$, so the conclusions of the pumping lemma must hold; i.e., we can write $s = uvxyz$ with $uv^ixy^iz \in A$ for all $i \geq 0$, $|vy| \geq 1$, and $|vxy| \leq p$. Since $|vxy| \leq p$, vxy can span at most 2 types of symbols. Let's consider all of the possibilities for vxy:

   - v and y are both uniform (i.e., v contains at most one type of symbol, and y contains at most one type of symbol). Since $|vy| \geq 1$, v and y cannot both be empty. Thus, for the string uv^2xy^2z, we have increased the number of symbols of at least one type and at most two types, so since there are 3 types of symbols, there must be more of one type of symbol than another type. Hence, uv^2xy^2z does not have the same number of c's, a's and b's, so $uv^2xy^2z \notin A$.
   - v and y are not both uniform (i.e., v contains two types of symbols, or y contains two types of symbols). Then uv^2xy^2z does not have all of the c's before all of the a's and all of the a's before all of the b's. Hence, $uv^2xy^2z \notin A$.

   Thus, all possibilities lead to a contradiction, so A must not be context-free.

Midterm Exam II
CIS 341: Foundations of Computer Science II — **Fall 2006, day section**
Prof. Marvin K. Nakayama

Print family (or last) name: ____________________________________

Print given (or first) name: ____________________________________

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

Signature and Date: ____________________________________

- This exam has 7 pages in total, numbered 1 to 7. Make sure your exam has all the pages.
- This exam will be 1 hour and 25 minutes in length.
- This is a closed-book, closed-note exam.
- For all problems, follow these instructions:
  1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
  2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton; CFG stands for context-free grammar; PDA stands for pushdown automaton.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step. Unless you are specifically asked to prove a theorem from the book, you may assume that the theorems in the textbook hold; i.e., you do not have to reprove the theorems in the textbook. When using a theorem from the textbook, make sure you provide enough detail so that it is clear which result you are using; e.g., say something like, "By the theorem that states $S^{**} = S^*$, it follows that …"

Problem	1	2	3	4	5	Total
Points						

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — If $A = L(N)$ for some NFA N, then A is nonregular.

(b) TRUE FALSE — If a language A has a DFA, then A is context-free.

(c) TRUE FALSE — The class of context-free languages is closed under concatenation.

(d) TRUE FALSE — If a language B is regular, then B cannot be context-free.

(e) TRUE FALSE — The language $\{ 0^n 1^n \mid n \geq 0 \}$ has a regular expression.

(f) TRUE FALSE — If a language L is regular, then L is finite.

(g) TRUE FALSE — If A and B are context-free languages, then $A \cap B$ is a context-free language.

(h) TRUE FALSE — For the alphabet $\Sigma = \{0, 1\}$, the language

$$A = \{ w \in \Sigma^* \mid w = w^{\mathcal{R}}, |w| \leq 100 \}$$

is not regular.

(i) TRUE FALSE — If a language A is regular and a language B is context-free, then $A \cup B^*$ is context-free.

(j) TRUE FALSE — Every language is regular or context-free.

2. **[30 points]** Give a short answer (at most three sentences) for each part below. Be sure to define any notation that you use.

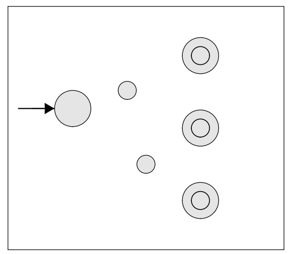
   (a) What does it mean for a context-free grammar to be in Chomsky normal form?

   (b) Give the formal definition of the transition function δ of a PDA.

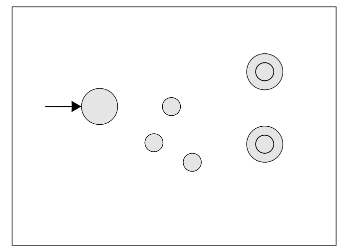
   (c) Suppose that language A_1 has CFG $G_1 = (V_1, \Sigma, R_1, S_1)$, and language A_2 has CFG $G_2 = (V_2, \Sigma, R_2, S_2)$. Give a CFG G_3 for $A_1 \cup A_2$ in terms of G_1 and G_2. You do not have to prove the correctness of your CFG G_3.

(d) Suppose that language A_1 is recognized by NFA N_1 below, and language A_2 is recognized by NFA N_2 below. Note that the transitions are not drawn in N_1 and N_2. Draw a picture of an NFA for $A_1 \circ A_2$.

N_1

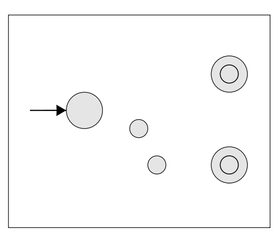


N_2



(e) Suppose that language A is recognized by the NFA N below. Note that the transitions are not drawn in N. Draw a picture of an NFA for A^*.

N



3. **[20 points]** Consider the alphabet $\Sigma = \{a, b\}$ and the language

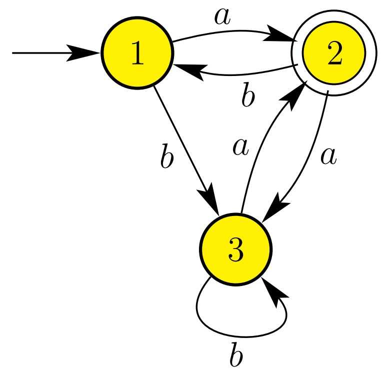
$$L = \{ \, w \in \Sigma^* \mid w = w^{\mathcal{R}} \text{ and } |w| \text{ is odd} \, \}.$$

(a) Give a context-free grammar G for L. Be sure to specify G as a 4-tuple $G = (V, \Sigma, R, S)$.

(b) Give a PDA for L. You only need to draw the graph.

Scratch-work area

4. **[15 points]** For the DFA M below, give a regular expression for $L(M)$.



Answer:

Scratch-work area

5. [**15 points**] Recall the pumping lemma for context-free languages:

Theorem: For every context-free language L, there exists a pumping length p such that, if $s \in L$ with $|s| \geq p$, then we can write $s = uvxyz$ with

(i) $uv^ixy^iz \in L$ for each $i \geq 0$,

(ii) $|vy| \geq 1$, and

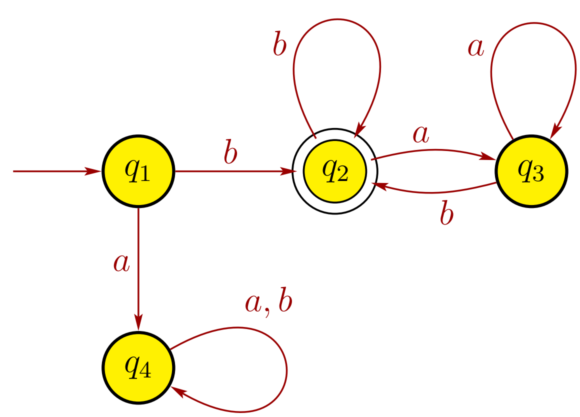
(iii) $|vxy| \leq p$.

Prove that $A = \{\, c^n a^n b^n \mid n \geq 0 \,\}$ is not a context-free language.

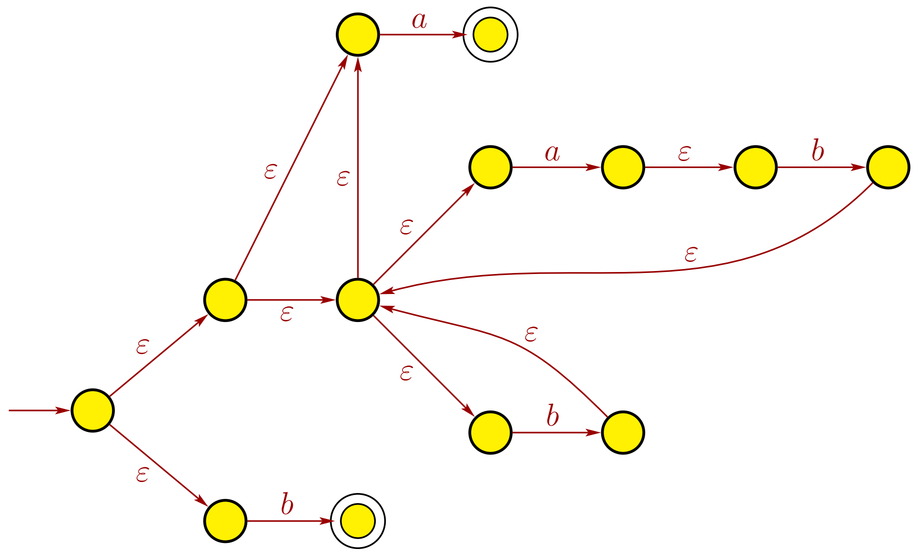
CS 341, Fall 2006
Solutions for Midterm, eLearning Section

1. (a) False. $A = \{0^n 1^n \mid n \geq 0\}$ is a context-free language, but it is nonregular. Hence, A cannot have an NFA.
   (b) True. Homework 5, problem 3(b).
   (c) True. Homework 5, problem 3(a).
   (d) True. Theorem 2.20.
   (e) True. If A is finite, then it must be regular (page 1-81 of notes).
   (f) False. $\{0, 1\}^*$ is a regular language that is infinite.
   (g) True, by Kleene's Theorem.
   (h) False. 0^*1^* generates $00111 \notin \{0^n 1^n \mid n \geq 0\}$.
   (i) False. $A \times B$ contains pairs of strings, whereas $A \circ B$ contains just strings.
   (j) True. page 1-50 of notes.

2. (a) Lexicographic order means that shorter strings appear before longer strings, and strings of the same length are in alphabetical order.
   (b) The difference is in the transition function δ. For a DFA, $\delta : Q \times \Sigma \to Q$, where Q is the set of states and Σ is the alphabet. For an NFA, $\delta : Q \times \Sigma_\varepsilon \to P(Q)$, where $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ and $P(Q)$ is the power set of Q.
   (c) $b \cup b(a \cup b)^* b$
   (d) DFA



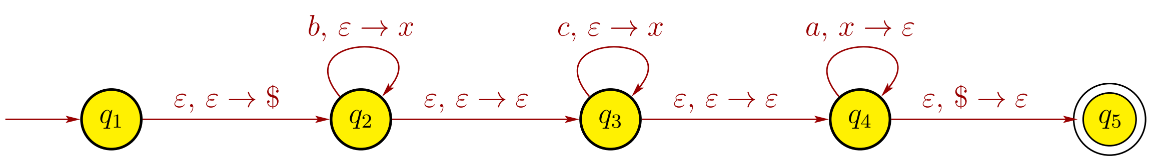
3. NFA



4. $(a \cup ba^*b)(ba \cup (bb \cup a)a^*b)$

5. (a) $G = (V, \Sigma, R, S)$, where $V = \{S, X\}$, $\Sigma = \{a, b, c\}$, and the rules are $S \to bSa \mid X$, $X \to cXa \mid \varepsilon$.

   (b) PDA



6. No, the class of context-free languages is not closed under intersection. Consider the languages $A = \{a^n b^n c^k \mid n, k \geq 0\}$ and $B = \{a^n b^k c^n \mid n, k \geq 0\}$. A CFG for A is $G_1 = (V_1, \Sigma, R_1, S_1)$, with $V_1 = \{S_1, X, Y\}$, $\Sigma = \{a, b, c\}$, and rules $S_1 \to XY$, $X \to aXb \mid \varepsilon$, $Y \to cY \mid \varepsilon$. A CFG for B is $G_2 = (V_2, \Sigma, R_2, S_2)$, with $V_2 = \{S_2, Z\}$, $\Sigma = \{a, b, c\}$, and rules $S_2 \to aS_2c \mid Z$, $Z \to bZ \mid \varepsilon$. Then $A \cap B = \{a^n b^n c^n \mid n \geq 0\}$, which is not context-free (see page 2-105 of the notes).

7. Suppose that $A = \{a^{3n} b^n c^{2n} \mid n \geq 0\}$ is a regular language. Let p be the pumping length, and consider the string $s = a^{3p} b^p c^{2p} \in A$. Note that $|s| = 6p \geq p$, so the pumping lemma implies we can write $s = xyz$ with $xy^i z \in A$ for all $i \geq 0$, $|y| > 0$, and $|xy| \leq p$. Now, $|xy| \leq p$ implies that x and y have only a's (together up to p

in total) and z has the rest of the a's followed by $b^p c^{2p}$. Hence, we can write $x = a^j$ for some $j \geq 0$, $y = a^k$ for some $k \geq 0$, and $z = a^\ell b^p c^{2p}$, where $j + k + \ell = 3p$ since $xyz = s = a^{3p} b^p c^{2p}$. Also, $|y| > 0$ implies $k > 0$. Now consider the string $xyyz = a^j a^k a^k a^\ell b^p c^{2p} = a^{3p+k} b^p c^{2p}$ since $j + k + \ell = 3p$. Note that $xyyz \notin A$, which is a contradiction, so A is not a regular language.

Midterm Exam
CS 341-451: Foundations of Computer Science II — **Fall 2006, eLearning section**
Prof. Marvin K. Nakayama

Print family (or last) name: ________________________________

Print given (or first) name: ________________________________

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

____

Signature and Date

- This exam has 8 pages in total, numbered 1 to 8. Make sure your exam has all the pages.
- The exam is to be given on Sunday, October 22, 2006, 12:30-3:00pm, EST.
- This is a closed-book, closed-note exam. No calculators are allowed.
- For all problems, follow these instructions:
  1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
  2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton; CFG stands for context-free grammar; PDA stands for pushdown automaton.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step.

Problem	1	2	3	4	5	6	7	Total
Points								

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — If A is a context-free language, then A is recognized by an NFA.

(b) TRUE FALSE — If A and B are context-free languages, then so is $A \circ B$.

(c) TRUE FALSE — If A and B are context-free languages, then so is $A \cup B$.

(d) TRUE FALSE — A language L has a CFG if and only if L is recognized by a PDA.

(e) TRUE FALSE — If a language A is not regular, then it must be infinite.

(f) TRUE FALSE — If a language is infinite, then it must not be regular.

(g) TRUE FALSE — If A is a regular language, then A has a regular expression.

(h) TRUE FALSE — A regular expression for the language $\{ 0^n 1^n \mid n \geq 0 \}$ is $0^* 1^*$.

(i) TRUE FALSE — If $A = \{01, 1\}$ and $B = \{\varepsilon\}$, then $A \times B = A \circ B$.

(j) TRUE FALSE — The class of regular languages is closed under concatenation.

2. **[20 points]** Give short answers to each of the following parts. **Each answer should be at most three sentences. Be sure to define any notation that you use.**

   (a) What does it mean for a set A of strings to be in lexicographic order?

   (b) Explain the difference between a DFA and an NFA.

   (c) Give a regular expression for the language consisting of strings over the alphabet $\Sigma = \{a, b\}$ that start and end with b.

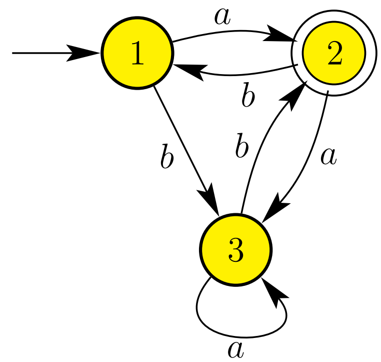
   (d) Give a DFA for the language consisting of strings over the alphabet $\Sigma = \{a, b\}$ that start and end with b. You only need to draw the graph; do not specify the DFA as a 5-tuple.

3. [**10 points**] Let A be the language over the alphabet $\Sigma = \{a, b\}$ defined by regular expression $(ab \cup b)^* a \cup b$. Give an NFA that recognizes A.

Draw an NFA for A here.

Scratch-work area

4. **[10 points]** For the DFA M below, give a regular expression for $L(M)$.



Answer:

Scratch-work area

5. **[20 points]** Consider the language $A = \{ b^i c^j a^k \mid i, j, k \geq 0 \text{ and } i + j = k \}$.

(a) Give a context-free grammar G that describes A. Be sure to specify G as a 4-tuple $G = (V, \Sigma, R, S)$.

(b) Give a pushdown automaton that recognizes A. You only need to draw the picture.

Scratch-work area

6. [**10 points**] Is the class of context-free languages closed under intersection?

Circle one: YES NO

- If YES, give a proof.
- If NO, give an example of two context-free languages A and B whose intersection is not context-free. Also, give the rules of the context-free grammars for A and B. You do not need to prove that the intersection of A and B is not context-free.

7. **[10 points]** Recall the pumping lemma for regular languages:

Theorem: For every regular language L, there exists a pumping length p such that, if $s \in L$ with $|s| \geq p$, then we can write $s = xyz$ with

(i) $xy^iz \in L$ for each $i \geq 0$,

(ii) $|y| > 0$, and

(iii) $|xy| \leq p$.

Prove that $A = \{\, a^{3n}b^nc^{2n} \mid n \geq 0 \,\}$ is not a regular language.

Midterm Exam I
CIS 341: Foundations of Computer Science II — **Spring 2006, day section**
Prof. Marvin K. Nakayama

Print family (or last) name: ____________________________________

Print given (or first) name: ____________________________________

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

____
Signature and Date

- This exam has 6 pages in total, numbered 1 to 6. Make sure your exam has all the pages.
- This exam will be 1 hour and 25 minutes in length.
- This is a closed-book, closed-note exam.
- For all problems, follow these instructions:
  1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
  2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step.

Problem	1	2	3	4	5	Total
Points						

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — If A is a regular language, then A is finite.

(b) TRUE FALSE — If R is any regular expression, then $L(R \cup \varepsilon) = L(R)$.

(c) TRUE FALSE — If R is any regular expression, then $L(R \circ \emptyset) = L(R)$.

(d) TRUE FALSE — A regular expression for the language $\{ b^n a^n \mid n \geq 0 \}$ is $\varepsilon \cup ba \cup bbaa \cup bbbaaa \cup \cdots$.

(e) TRUE FALSE — If language A has a regular expression, then $\overline{A}$ is recognized by some NFA.

(f) TRUE FALSE — If A is a nonregular language and B is a finite language, then $A \cup B$ must be nonregular.

(g) TRUE FALSE — If A is a nonregular language, then there exists a regular language B with $B \subseteq A$.

(h) TRUE FALSE — A DFA $M = (Q, \Sigma, \delta, q_0, F)$ may have $F = Q$.

(i) TRUE FALSE — A DFA $M = (Q, \Sigma, \delta, q_0, F)$ may have $F = \emptyset$.

(j) TRUE FALSE — If A is a regular language, then A^* must be nonregular.

2. **[20 points]** Give definitions or meanings of the following terms and phrases. Each answer should be at most two sentences. Be sure to define any notation that you use.

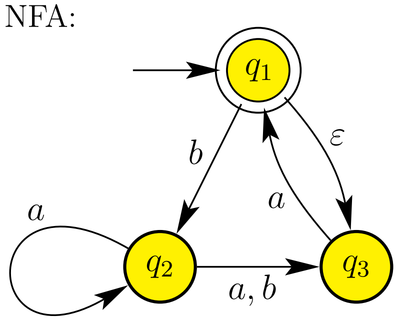
(a) For a machine M, what is $L(M)$?

(b) What does Kleene's Theorem say?

(c) Define the following statement: "The class of regular languages is closed under concatenation."

(d) Explain the difference between a DFA and an NFA.

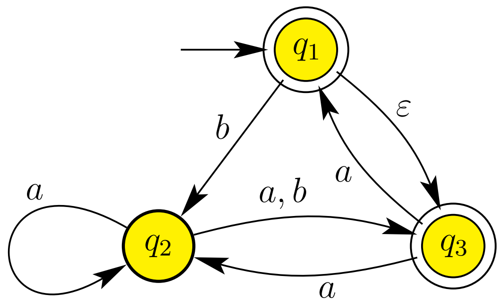
3. **[20 points]** Let A be the language over the alphabet $\Sigma = \{a, b\}$ defined by the following NFA:



Draw here a DFA that recognizes A.

Scratch-work area

4. **[20 points]** Let B be the language recognized by the following NFA:



Give a regular expression for B.

Answer:

Scratch-work area

5. **[20 points]** Recall the pumping lemma:

Theorem: If A is a regular language, then $\exists$ number p (pumping length) where, if $s \in A$ with $|s| \geq p$, then $\exists$ strings x, y, z such that $s = xyz$ and

(i) $xy^iz \in A$ for each $i \geq 0$,

(ii) $|y| > 0$, and

(iii) $|xy| \leq p$.

Prove that $C = \{\, b^{2n}a^{2n}b^{3n} \mid n \geq 0 \,\}$ is a nonregular language.

CIS 341, Spring 2006
Solutions for Midterm II

1. (a) True (Corollary 2.32)
   (b) False, e.g., $\{ 0^n 1^n \mid n \geq 0 \}$ is context-free (lecture notes, page 2-5) but not regular (lecture notes, page 1-90).
   (c) False, e.g., $\{ 0^n 1^n 2^n \mid n \geq 0 \}$ is decidable (Homework 7, problem 2) but not context-free (lecture notes, page 2-105).
   (d) False by Theorem 3.16.
   (e) False, TM M can loop on string $w \notin L(M)$.
   (f) False. Consider DFA A that always rejects ($L(A) = \emptyset$; see lecture notes, page 1-19) and DFA B that always accepts ($L(B) = \Sigma^*$; see lecture notes, page 1-18). Then $L(A) \neq L(B)$ but $L(A) \cap \overline{L(B)} = \emptyset$. (However, it is true that two DFAs A and B are equivalent if and only if $(L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B)) = \emptyset$; see Theorem 4.5.)
   (g) True by Theorem 3.21.
   (h) True by Theorem 4.2.
   (i) True by Homework 7, problem 3(b).
   (j) True by Homework 7, problem 3(a).

2. (a) A CFG $G = (V, \Sigma, R, S)$ is in Chomsky normal form means that each rule in R has one of the following forms:
      - $A \to BC$, where $A \in V$ and $B, C \in V - \{S\}$
      - $A \to x$, where $A \in V$ and $x \in \Sigma$
      - $S \to \varepsilon$ allowed.
   (b) A language L_1 that is Turing-recognizable has a Turing machine M_1 that may loop forever on a string $w \notin L_1$. A language L_2 that is Turing-decidable has a Turing machine M_2 that always halts.
   (c) The informal notion of an algorithm corresponds exactly to a Turing machine that always halts.
   (d) From Homework 5, problem 2(a).

$$S \to S \cup S \mid SS \mid S^* \mid (S) \mid 0 \mid 1 \mid e \mid \emptyset$$

3. This is from Homework 6, problem 2(a). Define the languages

$$\begin{aligned} A &= \{ a^m b^n c^n \mid m, n \geq 0 \} \text{ and} \\ B &= \{ a^n b^n c^m \mid m, n \geq 0 \} \end{aligned}$$

The language A is context free since it has CFG G_1 with rules

$$\begin{aligned} S &\to XY \\ X &\to aX \mid \varepsilon \\ Y &\to bYc \mid \varepsilon \end{aligned}$$

The language B is context free since it has CFG G_2 with rules

$$\begin{aligned} S &\to XY \\ X &\to aXb \mid \varepsilon \\ Y &\to aY \mid \varepsilon \end{aligned}$$

But $A \cap B = \{ a^n b^n c^n \mid n \geq 0 \}$, which we know is not context free from Example 2.36 of the textbook.

4. This is basically Homework 7, problem 1.

   (a) $q_1 000 \qquad \sqcup q_2 00 \qquad \sqcup x q_3 0 \qquad \sqcup x 0 q_4 \sqcup \qquad \sqcup x 0 \sqcup q_{\text{reject}}$

   (b) $q_1 00 \qquad \sqcup q_2 0 \qquad \sqcup x q_3 \sqcup \qquad \sqcup q_5 x \qquad q_5 \sqcup x \qquad \sqcup q_2 x \qquad \sqcup x q_2 \sqcup \qquad \sqcup x \sqcup q_{\text{accept}}$

5. This problem is a slight variation of Homework 8, problem 1. The equivalence problem for NFAs and regular expressions can be expressed as the language

   $EQ_{\text{NFA,REX}} = \{ \langle N, R \rangle \mid N$ is an NFA and R is a regular expression with $L(N) = L(R) \}$.

   The following TM M shows that $EQ_{\text{NFA,REX}}$ is decidable:

   M = "On input $\langle N, R \rangle$, where N is an NFA and R is a regular expression:
   1. Check if $\langle N, R \rangle$ is a proper encoding. If not, *reject*.
   2. Convert N into an equivalent DFA D_1 using the algorithm in the theorem (1.39) that shows every NFA has an equivalent DFA.
   3. Convert R into an equivalent DFA D_2 by first using the algorithm in Kleene's theorem for converting a regular expression into an equivalent NFA (Lemma 1.55), and then converting the NFA into an equivalent DFA (Theorem 1.39).
   4. Run TM S on input $\langle D_1, D_2 \rangle$, where S is the TM that decides EQ_{DFA}, the language corresponding to the equivalence problem for DFAs, which is decidable (Theorem 4.5).
   5. If S accepts, *accept*. If S rejects, *reject*.

Midterm Exam II
CIS 341: Foundations of Computer Science II — **Spring 2006, day section**
Prof. Marvin K. Nakayama

Print family (or last) name: ____________________________________

Print given (or first) name: ____________________________________

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

Signature and Date: ____________________________________

- This exam has 6 pages in total, numbered 1 to 6. Make sure your exam has all the pages.
- This exam will be 1 hour and 25 minutes in length.
- This is a closed-book, closed-note exam.
- For all problems, follow these instructions:
  1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
  2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton; CFG stands for context-free grammar; PDA stands for pushdown automaton.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step. Unless you are specifically asked to prove a theorem from the book, you may assume that the theorems in the textbook hold; i.e., you do not have to reprove the theorems in the textbook. When using a theorem from the textbook, make sure you provide enough detail so that it is clear which result you are using; e.g., say something like, "By the theorem that shows every NFA has an equivalent DFA, it follows that . . . "

Problem	1	2	3	4	5	Total
Points						

1. **[30 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — Every regular language is also context-free.

(b) TRUE FALSE — Every context-free language is also regular.

(c) TRUE FALSE — Every Turing-decidable language is also context-free.

(d) TRUE FALSE — There is a language A that is recognized by a nondeterministic Turing machine that is not recognized by any deterministic Turing machine.

(e) TRUE FALSE — For any Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ and any string $w \in \Sigma^*$, M will either accept or reject w.

(f) TRUE FALSE — Two DFAs A and B are equivalent if and only if $L(A) \cap \overline{L(B)} = \emptyset$.

(g) TRUE FALSE — A language A is Turing-recognizable if and only if some enumerator enumerates it.

(h) TRUE FALSE — The language

$$A_{\text{NFA}} = \{\, \langle N, w \rangle \mid N \text{ is an NFA that accepts } w \,\}$$

is Turing-decidable.

(i) TRUE FALSE — The class of Turing-recognizable languages is closed under union.

(j) TRUE FALSE — The class of Turing-decidable languages is closed under union.

2. **[20 points]** Give a short answer (at most two sentences) for each part below. No proofs are required, but be sure to define any notation that you use.

   (a) What does it mean for a context-free grammar to be in Chomsky normal form?

   (b) What is the difference between Turing-recognizable and Turing-decidable.

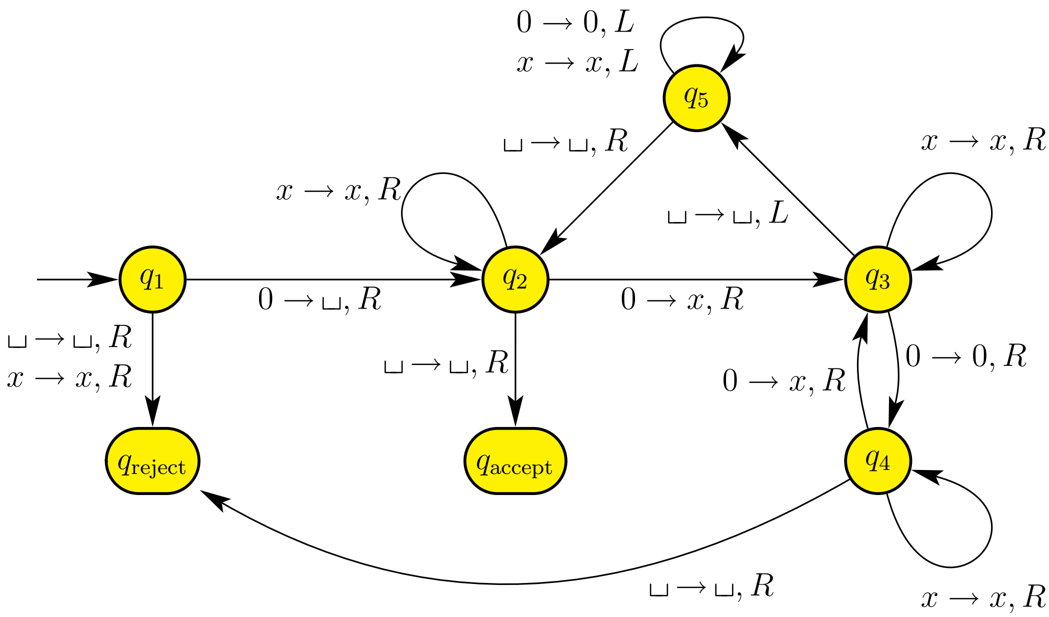
   (c) What is the Church-Turing Thesis?

   (d) Let $T = \{\, 0,\ 1,\ (,\ ),\ \cup,\ {}^*,\ \emptyset,\ e \,\}$. We may think of T as the set of symbols used by regular expressions over the alphabet $\{0, 1\}$; the only difference is that we use e for symbol ε, to avoid potential confusion. Give a CFG G with set of terminals T that generates exactly the regular expressions with alphabet $\{0, 1\}$.

3. [**15 points**] Give an example of two context-free languages A and B whose intersection $C = A \cap B$ is not context-free. Explain your answer, and follow these instructions:

   - Provide CFGs for A and B.
   - If we showed in class or in a homework that your language C is not context-free, then you do not need to provide a proof of this fact; however, be sure to clearly state that your language C was proven to be non-context-free in class or in a homework.
   - If we did not show your language C is context-free in class or in a homework, then you will need to provide an explicit proof of this fact.

4. **[20 points]** The Turing machine M below recognizes the language $A = \{ 0^{2^n} \mid n \geq 0 \}$.



In each of the parts below, give the sequence of configurations that M enters when started on the indicated input string.

(a) 000

(b) 00

Scratch-work area

5. **[15 points]** Consider the problem of testing whether an NFA and a regular expression are equivalent. Express this problem as a language and show that it is decidable. For this problem, you can apply any theorem that we went over in class without proving it, but make sure that you give enough details so that it is clear what theorem you are using (e.g., say something like, "By the theorem that says the class of regular languages is closed under union, we can show that ....")

CS 341, Fall 2007
Solutions for Midterm, eLearning Section

1. (a) False. The regular expression $a^*b^*a^*$ generates the string $aabaaa \notin A$. Moreover, A is not a regular language, so it cannot have a regular expression.

   (b) False. The conclusion doesn't make sense at all since $A \times B$ consists of pairs of strings but $A \circ B$ just consists of strings.

   (c) True. Corollary 2.32.

   (d) False. The language $\{ 0^n1^n \mid n \geq 0\}$ is context-free (see slide 2-5), but not regular (see slide 1-90).

   (e) False. The language $\{ 0^n1^n \mid n \geq 0\}$ is context-free (see slide 2-5), but is infinite.

   (f) False. The language $A = \{abb\}$ is finite and has CFG with rule $S \to abb$.

   (g) True, by Theorem 1.47.

   (h) True, by Kleene's Theorem (Theorem 1.54).

   (i) True, by Theorem 2.20.

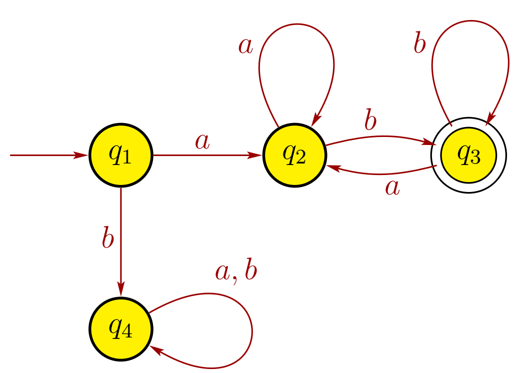
   (j) False. If A is recognized by an NFA, then A must be regular by Corollary 1.40.

2. (a) Lexicographic order means that shorter strings appear before longer strings, and strings of the same length are in alphabetical order.

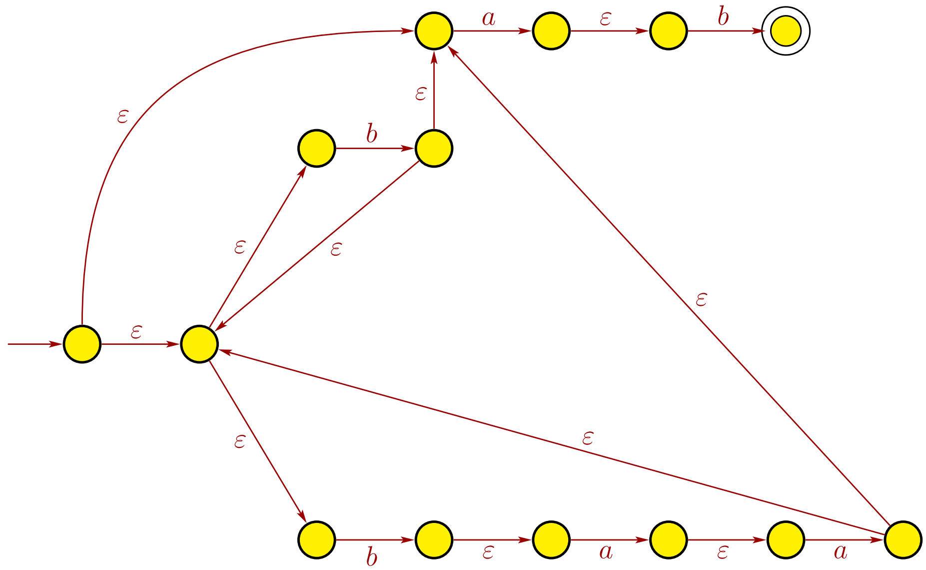
   (b) For a CFG $G = (V, \Sigma, R, S)$ to be in Chomsky normal form, each of its rules must have one of three forms: $A \to BC$, $A \to x$, or $S \to \varepsilon$, where A, B, C are variables, B and C are not the start variable, x is a terminal, and S is the start variable.

   (c) $a(a \cup b)^*b$

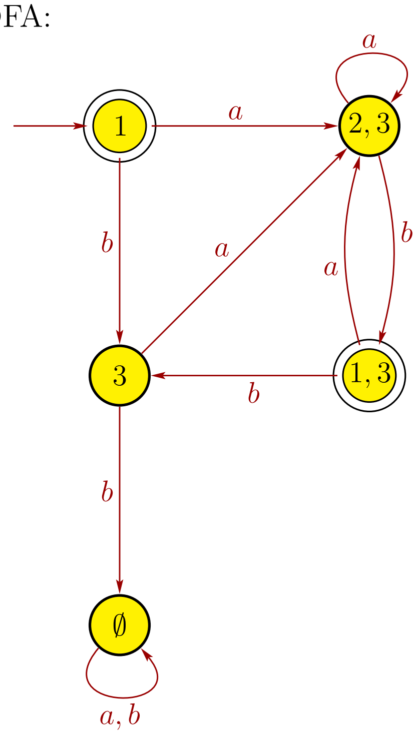
   (d) DFA



3. NFA

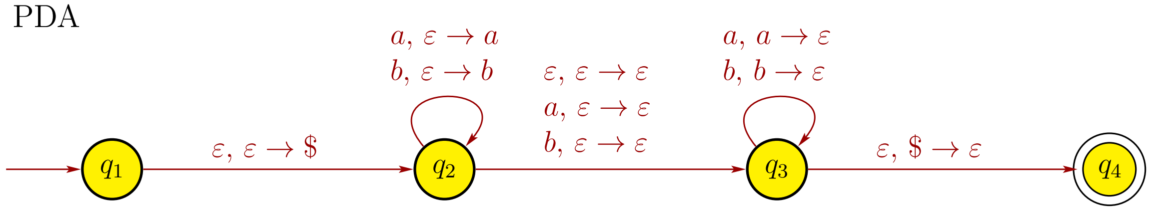


4. DFA:



5. (a) $G = (V, \Sigma, R, S)$, where $V = \{S\}$, $\Sigma = \{a, b\}$, and the rules are $S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$.

(b) PDA



6. No, the class of context-free languages is not closed under intersection. Consider the languages $A = \{a^n b^n c^k \mid n, k \geq 0\}$ and $B = \{a^n b^k c^n \mid n, k \geq 0\}$. A CFG for A is $G_1 = (V_1, \Sigma, R_1, S_1)$, with $V_1 = \{S_1, X, Y\}$, $\Sigma = \{a, b, c\}$, and rules $S_1 \to XY$, $X \to aXb \mid \varepsilon$, $Y \to cY \mid \varepsilon$. A CFG for B is $G_2 = (V_2, \Sigma, R_2, S_2)$, with $V_2 = \{S_2, Z\}$, $\Sigma = \{a, b, c\}$, and rules $S_2 \to aS_2c \mid Z$, $Z \to bZ \mid \varepsilon$. Then $A \cap B = \{a^n b^n c^n \mid n \geq 0\}$, which is not context-free (see page 2-105 of the notes).

7. Suppose that $A = \{w \in \Sigma^* \mid w = w^{\mathcal{R}}\}$ is a regular language. Let p be the pumping length, and consider the string $s = a^p b a^p \in A$. Note that $|s| = 2p + 1 \geq p$, so the pumping lemma implies we can write $s = xyz$ with $xy^iz \in A$ for all $i \geq 0$, $|y| > 0$, and $|xy| \leq p$. Now, $|xy| \leq p$ implies that x and y have only a's (together up to p in total) and z has the rest of the first set of a's, followed by ba^p. Hence, we can write $x = a^j$ for some $j \geq 0$, $y = a^k$ for some $k \geq 0$, and $z = a^\ell b a^p$, where $j + k + \ell = p$ since $xyz = s = a^p b a^p$. Also, $|y| > 0$ implies $k > 0$. Now consider the string $xyyz = a^j a^k a^k a^\ell b a^p = a^{p+k} b a^p$ since $j + k + \ell = p$. Note that $xyyz \notin A$ since $w \neq w^{\mathcal{R}}$, which is a contradiction, so A is not a regular language.

Midterm Exam
CS 341-451: Foundations of Computer Science II — **Fall 2007, eLearning section**
Prof. Marvin K. Nakayama

Print family (or last) name: ____

Print given (or first) name: ____

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

____
Signature and Date

- This exam has 8 pages in total, numbered 1 to 8. Make sure your exam has all the pages.
- Unless other arrangements have been made with the professor, the exam is to be given on Sunday, October 21, 2007. The exam is to last 2.5 hours.
- This is a closed-book, closed-note exam. No calculators are allowed.
- For all problems, follow these instructions:
  1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
  2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton; CFG stands for context-free grammar; PDA stands for pushdown automaton.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step.

Problem	1	2	3	4	5	6	7	Total
Points								

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — The language $\{ a^n b^n a^n \mid n \geq 0 \}$ has regular expression $a^* b^* a^*$.

(b) TRUE FALSE — If $A = \{\varepsilon\}$ and $B = \{01, 00\}$, then $A \times B = A \circ B$.

(c) TRUE FALSE — Every regular language is also context-free.

(d) TRUE FALSE — Every context-free language is also regular.

(e) TRUE FALSE — Every context-free language is finite.

(f) TRUE FALSE — Every context-free language is infinite.

(g) TRUE FALSE — The class of regular languages is closed under concatenation.

(h) TRUE FALSE — If a language is regular, then it must have a regular expression.

(i) TRUE FALSE — If a language is recognized by a PDA, then the language has a CFG.

(j) TRUE FALSE — NFAs recognize nonregular languages.

2. **[20 points]** Give short answers to each of the following parts. **Each answer should be at most three sentences. Be sure to define any notation that you use.**

   (a) What does it mean for a list of strings to be in lexicographic order?

   (b) What does it mean for a context-free grammar $G = (V, \Sigma, R, S)$ to be in Chomsky normal form?

   (c) Give a regular expression for the language L consisting of strings over the alphabet $\Sigma = \{a, b\}$ that begin with a and end with b.

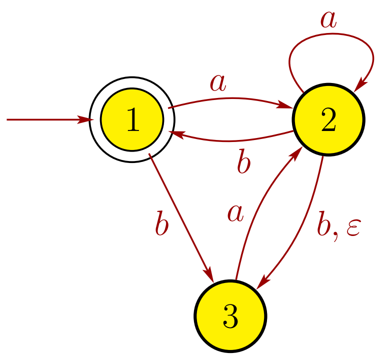
   (d) Give a DFA for the language consisting of strings over the alphabet $\Sigma = \{a, b\}$ that begin with a and end with b. You only need to draw the graph; do not specify the DFA as a 5-tuple.

3. [**10 points**] Let A be the language over the alphabet $\Sigma = \{a, b\}$ defined by regular expression $(b \cup baa)^* ab$. Give an NFA that recognizes A.

Draw an NFA for A here.

Scratch-work area

4. [**10 points**] Convert the following NFA into an equivalent DFA.



Answer:

Scratch-work area

5. **[20 points]** Let $\Sigma = \{a, b\}$, and consider the language $A = \{ w \in \Sigma^* \mid w = w^{\mathcal{R}} \}$, where $w^{\mathcal{R}}$ denotes the reverse of w.

(a) Give a context-free grammar G that describes A. Be sure to specify G as a 4-tuple $G = (V, \Sigma, R, S)$.

(b) Give a pushdown automaton that recognizes A. You only need to draw the picture.

Scratch-work area

6. [**10 points**] Is the class of context-free languages closed under intersection?

Circle one: YES NO

- If YES, give a proof.
- If NO, give an example of two context-free languages A and B whose intersection is not context-free. Also, give the rules of the context-free grammars for A and B. You do not need to prove that the intersection of A and B is not context-free.

7. **[10 points]** Recall the pumping lemma for regular languages:

Theorem: For every regular language L, there exists a pumping length p such that, if $s \in L$ with $|s| \geq p$, then we can write $s = xyz$ with

(i) $xy^iz \in L$ for each $i \geq 0$,

(ii) $|y| > 0$, and

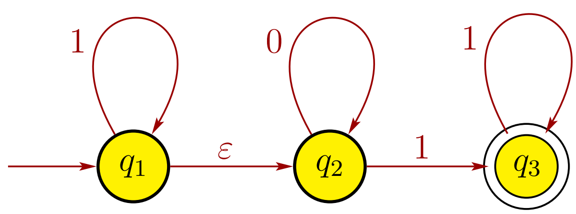
(iii) $|xy| \leq p$.

Let $\Sigma = \{a, b\}$, and consider the language $A = \{\, w \in \Sigma^* \mid w = w^{\mathcal{R}} \}$, where $w^{\mathcal{R}}$ denotes the reverse of w. Prove that A is not a regular language.

CS 341, Spring 2007
Solutions for Midterm 1

1. (a) False. Language a^* is infinite but regular.
   (b) False. a^*b^* generates $aaabb \notin \{a^nb^n | n \geq 0\}$.
   (c) True. If A has an NFA, then A is regular by Corollary 1.40. Corollary 2.32 implies that A is CFL, so A has a PDA by Lemma 2.21.
   (d) True. It has CFG $S \to 00S \mid 01S \mid 10S \mid 11S \mid \varepsilon$.
   (e) True. It has regular expression $((0 \cup 1)(0 \cup 1))^*$.
   (f) False. Homework 6, problem 2(a).
   (g) True. The class of languages recognized by DFAs is the class of regular languages, which is closed under complementation (Homework 2, problem 3).
   (h) False. Every CFL has a CFG in Chomsky normal form by Theorem 2.9. But not every CFL is regular, e.g., $\{a^nb^n | n \geq 0\}$.
   (i) False. $\{a^nb^nc^n | n \geq 0\}$ is neither regular nor context-free.
   (j) False. A nonregular language A cannot have a regular expression. If A had a regular expression, then Kleene's Theorem implies it must have a DFA, so A would be regular.

2. (a) Here is an NFA with exactly 3 states:



   (b) CFG $G = (V, \Sigma, R, S)$ is Chomsky normal form means that each rule in R has one of 3 forms:

$$\begin{array}{rcl} A & \to & BC \\ A & \to & x \\ S & \to & \varepsilon \end{array}$$

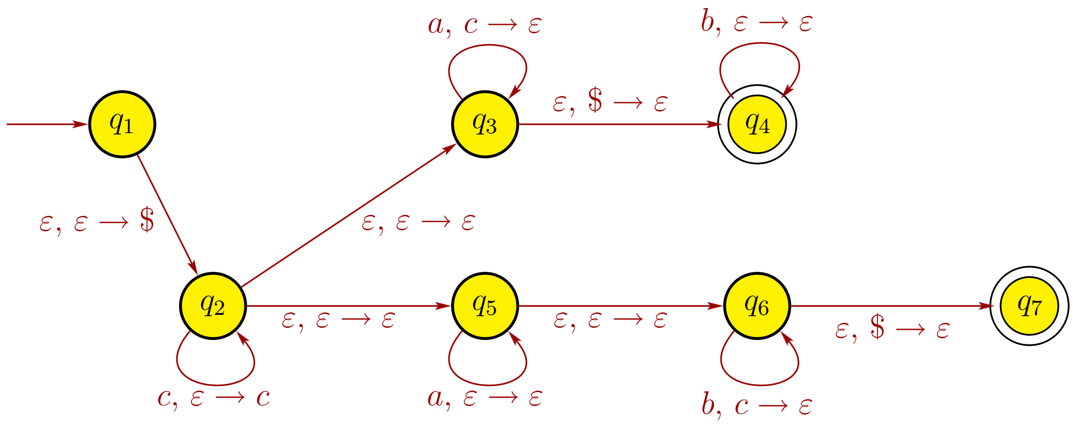
   where $A \in V$; $B, C \in V - \{S\}$; $x \in \Sigma$, and S is the start variable.
   (c) See page 1-50 of notes.
   (d) Homework 5, problem 3(c).

3. (a) $a, b, ab, ba, bb, aab, abb, bab, \ldots$
   (b) $(a \cup b^*(a \cup b))(b \cup a(b^*b \cup bb^*a))^*$

4. (a) $G = (V, \Sigma, R, S)$, where $V = \{S, W, X, Y, Z\}$, $\Sigma = \{a, b, c\}$, S is the start variable, and the rules R are

$$\begin{array}{rcl} S & \rightarrow & WX \mid Y \\ W & \rightarrow & cWa \mid \varepsilon \\ X & \rightarrow & bX \mid \varepsilon \\ Y & \rightarrow & cYb \mid Z \\ Z & \rightarrow & aZ \mid \varepsilon \end{array}$$

(b)



5. We prove this by contradiction. Suppose that A is a regular language. Let p be the "pumping length" of the Pumping Lemma. Consider the string $s = c^{3p}a^pb^{2p}$. Note that $s \in A$, and $|s| = 6p \geq p$, so the Pumping Lemma will hold. Thus, there exists strings x, y, and z such that $s = xyz$ and

(a) $xy^iz \in A_3$ for each $i \geq 0$,

(b) $|y| > 0$,

(c) $|xy| \leq p$.

Since the first p symbols of s are all c's, the third property implies that x and y consist only of c's. So z will be the rest of the c's, followed by a^pb^{2p}. The second property states that $|y| > 0$, so y has at least one c. More precisely, we can then say that

$$\begin{array}{rcl} x & = & c^j \text{ for some } j \geq 0, \\ y & = & c^k \text{ for some } k \geq 1, \\ z & = & c^m a^p b^{2p} \text{ for some } m \geq 0. \end{array}$$

Since $c^{3p}a^pb^{2p} = s = xyz = c^jc^kc^ma^pb^{2p} = c^{j+k+m}a^pb^{2p}$, we must have that $j + k + m = 3p$. The first property implies that $xy^2z \in A$, but

$$\begin{array}{rcl} xy^2z & = & c^jc^kc^kc^ma^pb^{2p} \\ & = & c^{3p+k}a^pb^{2p} \end{array}$$

since $j + k + m = 3p$. Hence, $xy^2z \notin A$ because $k \geq 1$, and we get a contradiction. Therefore, A is a nonregular language.

Midterm Exam 1
CS 341: Foundations of Computer Science II — **Spring 2007, day section**
Prof. Marvin K. Nakayama

Print family (or last) name: ______________________________

Print given (or first) name: ______________________________

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

____
Signature and Date

- This exam has 7 pages in total, numbered 1 to 7. Make sure your exam has all the pages.
- Note the number written on the upper right-hand corner of the first page. On the sign-up sheet being passed around, sign your name next to this number.
- This exam will be 1 hour and 25 minutes in length.
- This is a closed-book, closed-note exam.
- For all problems, follow these instructions:
    1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
    2. For any proofs, be sure to provide a step-by-step argument, with justifications for every step.

Problem	1	2	3	4	5	Total
Points						

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — A language is regular if and only if it is finite.

(b) TRUE FALSE — A regular expression for the language $\{\, a^n b^n \mid n \geq 0 \,\}$ is $a^* b^*$.

(c) TRUE FALSE — If a language A has an NFA, then A must be recognized by some PDA.

(d) TRUE FALSE — The language $\{\, w \in \{0,1\}^* \mid |w| \text{ is even} \,\}$ is context-free.

(e) TRUE FALSE — The language $\{\, w \in \{0,1\}^* \mid |w| \text{ is even} \,\}$ is regular.

(f) TRUE FALSE — If A and B are context-free languages, then $A \cap B$ must be context-free.

(g) TRUE FALSE — The class of languages recognized by DFAs is closed under complementation.

(h) TRUE FALSE — If A is a language generated by a context-free grammar in Chomsky normal form, then A must be regular.

(i) TRUE FALSE — Every language must be regular or context-free.

(j) TRUE FALSE — Nonregular languages have regular expressions but are not recognized by any DFA.

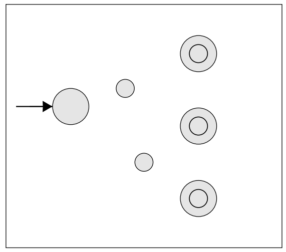
2. **[20 points]** Give short answers to each of the following parts. Each answer should be at most three sentences. Be sure to define any notation that you use.

   (a) Give an NFA with exactly three states for the language having regular expression 1*0*1*1.

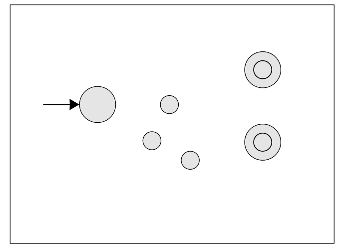
   (b) What does it mean for a context-free grammar $G = (V, \Sigma, R, S)$ to be in Chomsky normal form?

(c) Suppose that language A_1 is recognized by NFA N_1 below, and language A_2 is recognized by NFA N_2 below. Note that the transitions are not drawn in N_1 and N_2. Draw a picture of an NFA for $A_1 \circ A_2$.

N_1

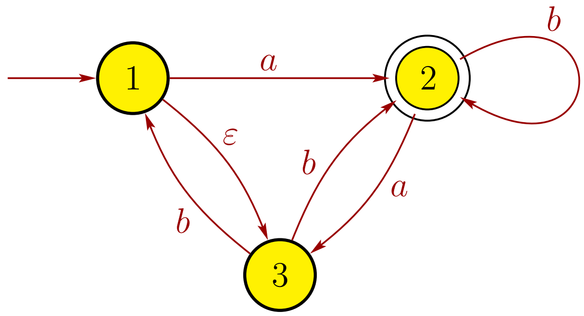


N_2



(d) Suppose that language A has CFG $G_1 = (V_1, \Sigma, R_1, S_1)$. Give a CFG G_2 for A^* in terms of G_1. You do not have to prove the correctness of your CFG G_2, but do not give just an example.

3. **[20 points]** Let N be the following NFA with $\Sigma = \{a, b\}$, and let $C = L(N)$.



(a) List the strings in C in lexicographic order. If C has more than 8 strings, list only the first 8 strings in C, followed by 3 dots.

(b) Give a regular expression for C.

Scratch-work area

4. **[25 points]** Consider the language

$$L = \{\, c^i a^j b^k \mid i, j, k \geq 0, \text{ and } i = j \text{ or } i = k \,\}.$$

(a) Give a context-free grammar G for L. Be sure to specify G as a 4-tuple $G = (V, \Sigma, R, S)$.

(b) Give a PDA for L. You only need to draw the graph.

Scratch-work area

5. [**15 points**] Recall the pumping lemma for regular languages:

Theorem: If L is a regular language, then there exists a pumping length p where, if $s \in L$ with $|s| \geq p$, then there exists strings x, y, z such that $s = xyz$ and

(i) $xy^iz \in L$ for each $i \geq 0$,

(ii) $|y| \geq 1$, and

(iii) $|xy| \leq p$.

Prove that $A = \{ c^{3n}a^nb^{2n} \mid n \geq 0 \}$ is not a regular language.

CS 341, Spring 2007
Solutions for Midterm 2

1. (a) True, see slide 4-54.
   (b) True, see slide 4-54.
   (c) False by Theorem 3.16.
   (d) True, by Theorems 3.13 and 3.16.
   (e) False. Suppose that A is decidable, which implies that it is also Turing-recognizable. But since A is decidable, it is recognized by a Turing machine that never loops.
   (f) False. This is just A_{TM}, which is undecidable by Theorem 4.11.
   (g) False, since $\overline{A_{\mathrm{TM}}}$ is not Turing-recognizable by Corollary 4.23.
   (h) False, by Theorem 5.2.
   (i) True, by Theorem 4.4.
   (j) True, by Theorem 4.8.

2. (a) A language L_1 that is Turing-recognizable has a Turing machine M_1 that may loop forever on a string $w \notin L_1$. A language L_2 that is Turing-decidable has a Turing machine M_2 that always halts.
   (b) The informal notion of an algorithm corresponds exactly to a Turing machine that always halts.

3. Homework 9, problem 1.

4. This is basically Homework 7, problem 1.

   (a) $q_1 00 \quad \sqcup q_2 0 \quad \sqcup x q_3 \sqcup \quad \sqcup q_5 x \quad q_5 \sqcup x \quad \sqcup q_2 x \quad \sqcup x q_2 \sqcup \quad \sqcup x \sqcup q_{\text{accept}}$

   (b) $q_1 000 \quad \sqcup q_2 00 \quad \sqcup x q_3 0 \quad \sqcup x 0 q_4 \sqcup \quad \sqcup x 0 \sqcup q_{\text{reject}}$

5. Homework 8, problem 1.

6. This is Theorem 5.1, whose proof is given on slide 5-8.

Midterm Exam 2
CS 341: Foundations of Computer Science II — **Spring 2007, day section**
Prof. Marvin K. Nakayama

Print family (or last) name: ______________________________

Print given (or first) name: ______________________________

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

Signature and Date: ______________________________

- This exam has 7 pages in total, numbered 1 to 7. Make sure your exam has all the pages.
- This exam will be 1 hour and 25 minutes in length.
- This is a closed-book, closed-note exam.
- For all problems, follow these instructions:
  1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area or the backs of the sheets to work out your answers before filling in the answer space.
  2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton; CFG stands for context-free grammar; PDA stands for pushdown automaton.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step. Unless you are specifically asked to prove a theorem from the book, you may assume that the theorems in the textbook hold; i.e., you do not have to reprove the theorems in the textbook. When using a theorem from the textbook, make sure you provide enough detail so that it is clear which result you are using; e.g., say something like, "By the theorem that states $S^{**} = S^*$, it follows that ..."

Problem	1	2	3	4	5	Total
Points						

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — If A is regular, then A is Turing-recognizable.

(b) TRUE FALSE — If A is regular, then A is Turing-decidable.

(c) TRUE FALSE — If a language A is recognized by a nondeterministic Turing machine, then A is not Turing-recognizable.

(d) TRUE FALSE — Every 17-tape nondeterministic Turing machine has an equivalent 1-tape deterministic Turing machine.

(e) TRUE FALSE — If A is Turing-recognizable and Turing machine M recognizes A, then there must exist a string $w \notin A$ such that M loops on w.

(f) TRUE FALSE — There is a Turing machine that can take as input $\langle M, w \rangle$, where M is a Turing machine and w is a string, and decide if M accepts w.

(g) TRUE FALSE — We can use the universal Turing machine U to show that $\overline{A_{\text{TM}}}$ is Turing-recognizable.

(h) TRUE FALSE — The language $E_{\text{TM}} = \{ \langle M \rangle \mid M \text{ is a TM with } L(M) = \emptyset \}$ is decidable.

(i) TRUE FALSE — The language $E_{\text{DFA}} = \{ \langle D \rangle \mid D \text{ is a DFA with } L(D) = \emptyset \}$ is decidable.

(j) TRUE FALSE — The language $E_{\text{CFG}} = \{ \langle G \rangle \mid G \text{ is a CFG with } L(G) = \emptyset \}$ is decidable.

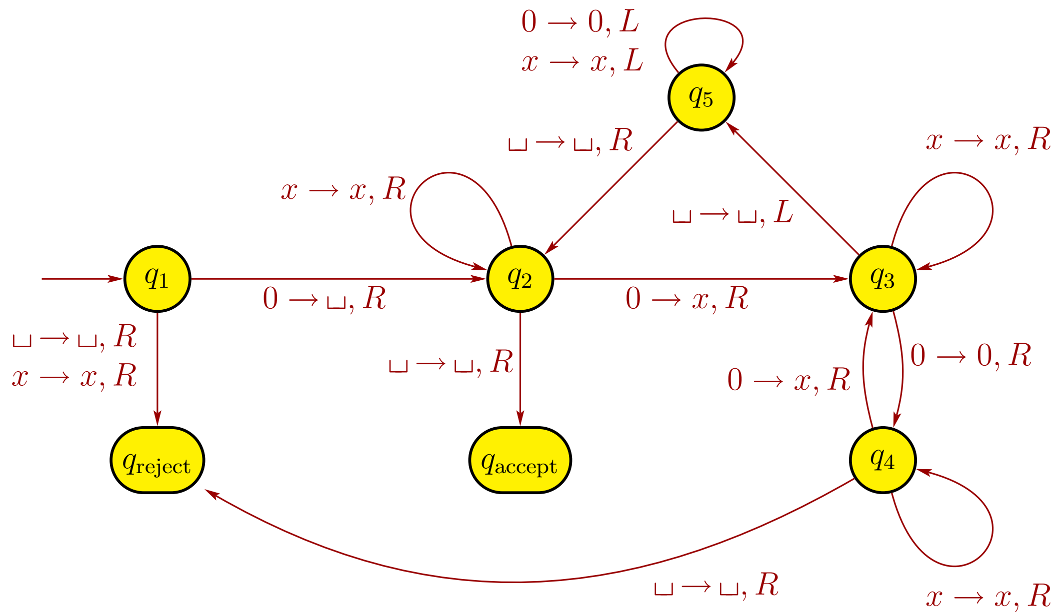
2. **[20 points]** Give a short answer (at most three sentences) for each part below. Be sure to define any notation that you use.

   (a) Explain the difference between Turing-recognizable and Turing-decidable.

   (b) What is the Church-Turing thesis?

3. **[10 points]** Let $\mathcal{B}$ be the set of infinite binary sequences. Show that $\mathcal{B}$ is uncountable.

4. **[20 points]** The Turing machine M below recognizes the language $A = \{ 0^{2^n} \mid n \geq 0 \}$.



In each of the parts below, give the sequence of configurations that M enters when started on the indicated input string.

(a) 00

(b) 000

Scratch-work area

5. **[15 points]** Consider the problem of testing whether a DFA and a regular expression are equivalent. Express this problem as a language and show that it is decidable. For this problem, you can apply any theorem that we went over in class without proving it, but make sure that you give enough details so that it is clear what theorem you are using (e.g., say something like, "By the theorem that says every context-free language has a CFG in Chomsky normal form, we can show that ....")

6. **[15 points]** Recall that

$$HALT_{\text{TM}} = \{ \langle M, w \rangle \mid M \text{ is a Turing machine that halts on input } w \}.$$

Prove that $HALT_{\text{TM}}$ is undecidable by showing that A_{TM} reduces to $HALT_{\text{TM}}$, where

$$A_{\text{TM}} = \{ \langle M, w \rangle \mid M \text{ is a Turing machine that accepts input } w \}.$$

Review for the Final Exam

A. Regular Languages
- DFAs, NFAs, ε-NFAs   You should be able to convert any of the others to a DFA.
- Regular Expressions. It is fairly easy to convert a regular expression to a DFA. It is possible but harder to convert a DFA to a regular expression.
- The Pumping Lemma: If $|w| > p$ then $w=xyz$ where $|xy|<=p$, y is not empty, and xy^iz is in the language for all $i>= 0$.
- Properties of Regular Languages: Unions, Intersections, Differences and Complements of regular languages are regular.

B. Context-Free Languages
- Grammars
- PDAs
- To show that grammars generate the same languages as PDAs we found algorithms to convert a grammar to a PDA (easy) and to convert a PDA to a grammar (hard).
- Chomsky Normal Form and the algorithm for finding a CNF grammar equivalent to a given grammar.
- The Pumping Lemma for Context-Free languages: If $|z| > p$ then $z=uvwxy$ where $|vwx|<=p$, v and x aren't both empty, and uv^iwx^iy is in the language for all $i>= 0$.
- Properties of CF Languages: Unions and concatenations of CF languages are CF. Intersections and Complements of CF languages are not necessarily CF.

C. Turing Machines
- Simple TMs, multli-track, multi-tape and non-deterministic TMs
- Church's Thesis: TMs embody our notion of an algorithm

D. Decidability
- Recursive languages, Recursively enumerable languages, Decidable problems, Recognizable problems
- The diagonal language $\mathcal{L}_d$={M | M does not accept its own encoding} is not RE but its complement is RE.
- The universal language $\mathcal{L}_u$ = {(M,w) | M accepts w} is RE but not Recursive. The complement of $\mathcal{L}_u$ is not RE.
- The halting language $\mathcal{L}_{halt}$ = {(M.w) | M halts on input w} is RE but not recursive.
- Rice's Theorem: Any nontrivial property of context-free languages is undecidable.
- Reductions: To show that a language $\mathcal{L}$ is not RE or is not Recursive, reduce a language $\mathcal{L}$* that you know is not RE or is not Recursive to it. This means showing that if you a recognizer or decider for $\mathcal{L}$, then you would also have one for $\mathcal{L}$*

E. NP-Completeness

- $\mathscr{P}$ is the class of problems that can be solved deterministically in polynomial time
- $\mathscr{NP}$ is the class of problems that can be solved non-deterministically in polynomial time, which usually means that a solution can be verified deterministically in polynomial time.
- Cook's (or Cook-Levin) Theorem: SAT is NP-Complete.
- You should know what all of this means, but I am unlikely to ask you to prove that a specific language is NP-Complete.

Show that the set of Turing Machines that accept only a finite number of strings is not Recursively Enumerable.

Proof. We reduce the complement of the halting language to this. Suppose we have an (M, w) pair. Build M' so that (a) M' accepts all strings of length 2 or less, and (b) if |x| > 2 M'(x) simulates M on w for |x| steps. If M halts on w within |x| steps M' accepts x; otherwise M' rejects x. If M does halt within n steps M' accepts all x with |x| >= n, which is an infinite set. If M does not halt on w M' accepts only the strings of length 2 or less, which is a finite set. If we could recognize if M' accepts only a finite set of string we could also recognize if M does not halt on w. But we can't.