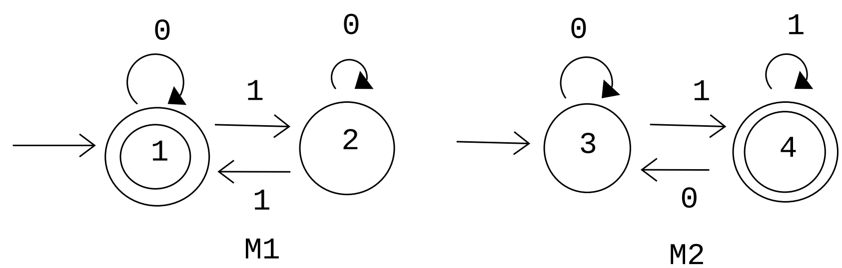
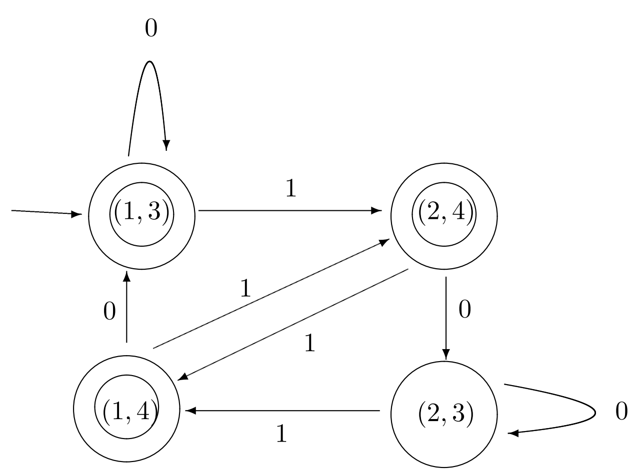
CS 420
First Test Practice Questions Solutions

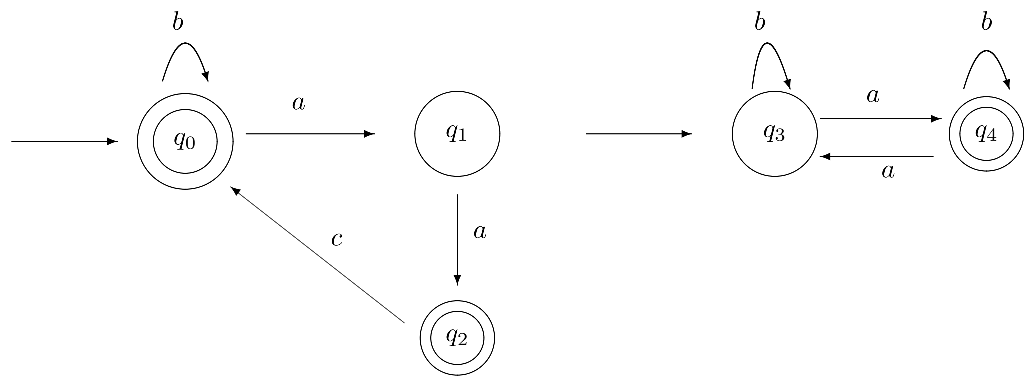
1. Using the method of Theorem 1.25, give a DFA that recognizes the union of the languages recognized by the two DFAs given below. (You must use the method of Theorem 1.25, not the one of Theorem 1.45, which produces an NFA.)



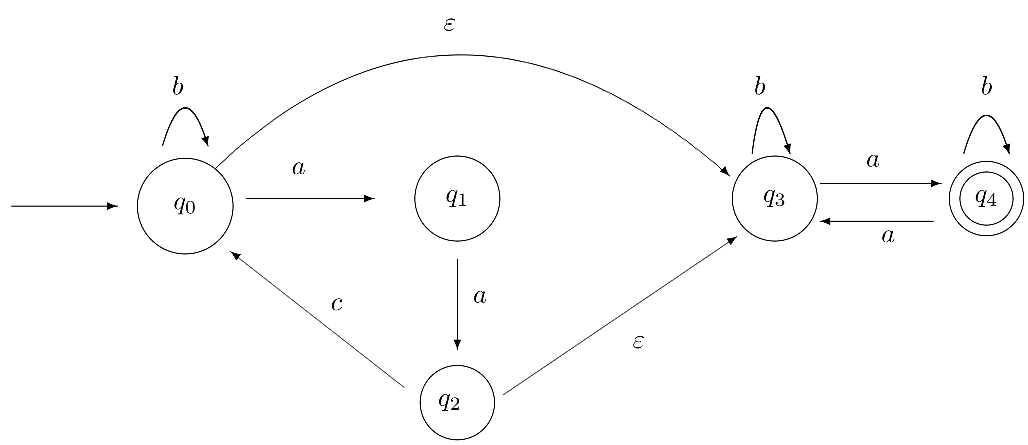
Answer:



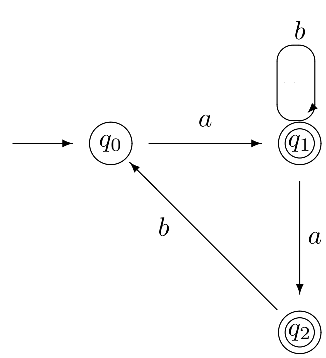
2. Using the method from class, give an NFA that recognizes the concatenation of the language recognized by the first automaton below with the language recognized by the second automaton below:



Answer:



3. Using the method from class, give a nondeterministic finite automaton that recognizes the star of the language recognized by the following NFA.



Answer:

See Figure 1.

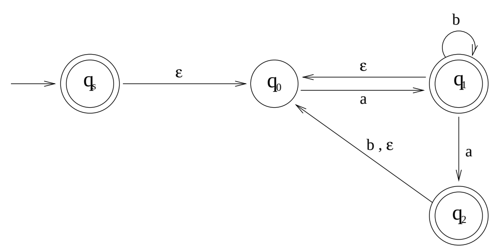
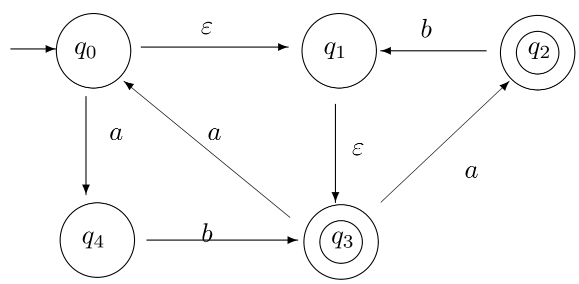


Figure 1: Solution to Problem 3

4. Let N be the NFA given by



and let M be the DFA equivalent to N obtained by the method from class. Answer the following questions about N and M. (Put your answers in the blank spaces.)

(a) Does N accept the string ε? yes

(b) Does N accept the string b? no

(c) Does N accept the string ab? yes

(d) Does N accept the string abb? no

(e) What is the start state of M? $\{q_0, q_1, q_3\}$

(f) Is $\emptyset$ an accept state or a reject state of M? reject

(g) Is $\{q_1, q_3\}$ an accept state or a reject state of M? accept

(h) What state does M go to from state $\{q_0, q_1, q_3\}$ reading an a?

$\{q_0, q_1, q_2, q_3, q_4\}$

(i) What state does M go to from state $\{q_0, q_1, q_3\}$ reading a b?

$\emptyset$

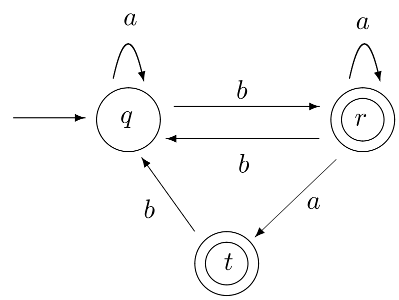
(j) What state does M go to from state $\{q_0, q_1, q_2, q_3\}$ reading a b?

$\underline{\{q_1, q_3\}}$

(k) What state does M go to from state $\{q_0, q_1, q_2, q_3, q_4\}$ reading an a?

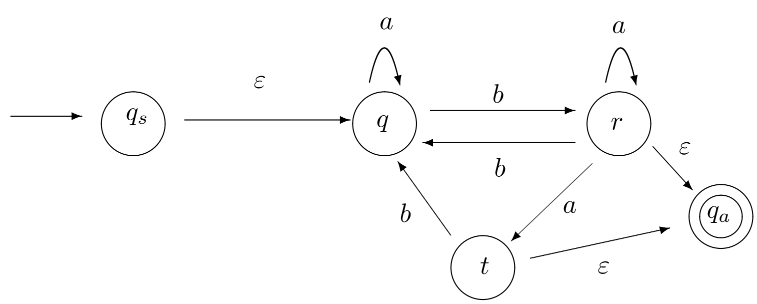
$\underline{\{q_0, q_1, q_2, q_3, q_4\}}$

5. (a) Let M be the NFA given below.

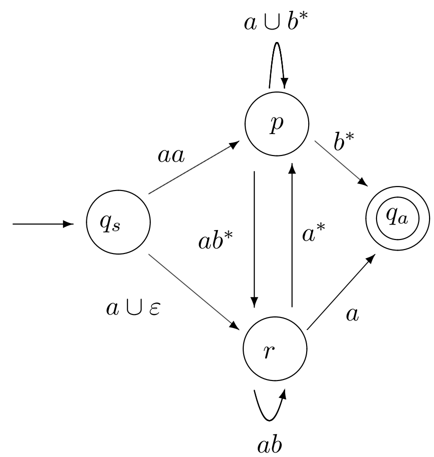


If you want to transform M into an equivalent regular expression, you have to first modify M into a GNFA M' *before* you start eliminating states. Give the state diagram of this GNFA M'. (You are *not* being asked to eliminate any states. Just give the GNFA M' that you first transform M into.)

Answer:

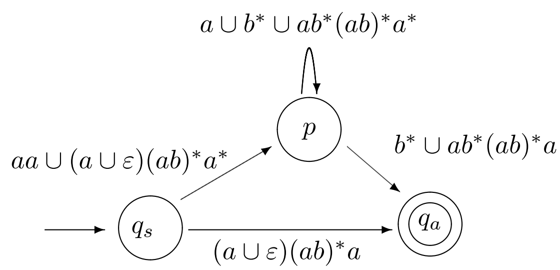


(b) Suppose that at some point while transforming an NFA into a regular expression you have the following GNFA.

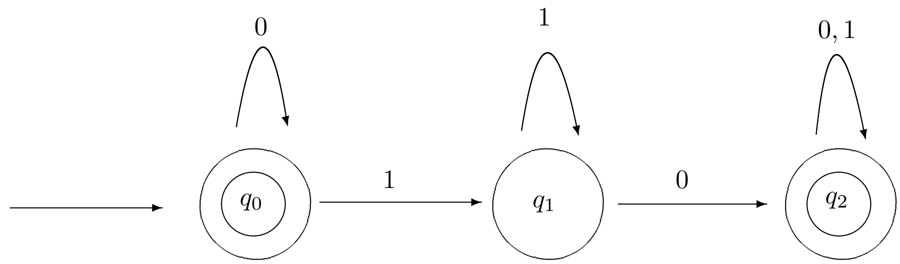


Show the GNFA you would get from this one by eliminating state r. (You are not being asked to convert the GNFA into a regular expression. Just eliminate the state r.)

Answer:



6. Convert the DFA given below into an equivalent right regular grammar.



Answer:

$$\begin{aligned} A_0 &\rightarrow 0A_0|1A_1|\varepsilon \\ A_1 &\rightarrow 0A_2|1A_1 \\ A_2 &\rightarrow 0A_2|1A_2|\varepsilon \end{aligned}$$

7. Convert the right regular grammar given below into an equivalent NFA.

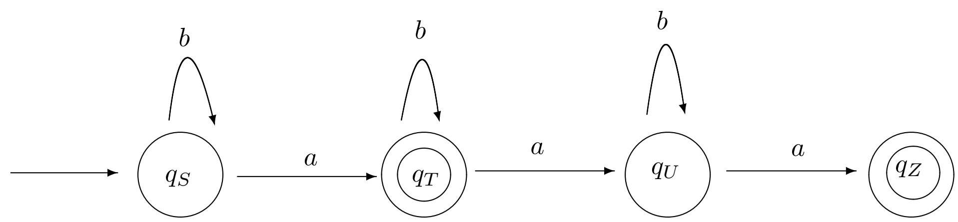
$$\begin{aligned} S &\rightarrow bS|aT \\ T &\rightarrow bT|aU|\varepsilon \\ U &\rightarrow bU|a \end{aligned}$$

Answer:

We first have to replace the rule $U \rightarrow a$, giving the grammar:

$$\begin{aligned} S &\rightarrow bS|aT \\ T &\rightarrow bT|aU|\varepsilon \\ U &\rightarrow bU|aZ \\ Z &\rightarrow \varepsilon \end{aligned}$$

Then we get the NFA:



8. (a) Suppose that you are in the second step of putting a grammar into Chomsky normal form (eliminate ε-rules). You have already eliminated $X \to \varepsilon$ and the current grammar is:

$$\begin{array}{rcl} S_0 & \to & S \\ S & \to & \varepsilon|SaS|SXY \\ X & \to & YS|S|a \\ Y & \to & SS|a \end{array}$$

Show the grammar that you get by eliminating $S \to \varepsilon$. (You are not being asked to eliminate all ε- rules, just $S \to \varepsilon$.)

$$\begin{array}{rcl} S_0 & \to & \underline{S|\varepsilon} \\ S & \to & \underline{SaS|SXY|aS|Sa|a|XY} \\ X & \to & \underline{YS|S|a|Y} \\ Y & \to & \underline{SS|a|S|\varepsilon} \end{array}$$

(b) Suppose that you are in the third step of putting a grammar into Chomsky normal form (eliminate unit rules). You have already eliminated $S \to V$ and the current grammar is:

$$\begin{array}{rcl} S_0 & \to & S \\ S & \to & aTb|U|VaV|b \\ U & \to & aUb|V|W|a \\ V & \to & VaV|b \\ W & \to & ab \end{array}$$

Show the grammar that you get by eliminating $S \to U$. (You are not being asked to eliminate all unit rules, just $S \to U$.)

$$\begin{array}{rcl} S_0 & \to & \underline{S} \\ S & \to & \underline{aTb|VaV|b|aUb|W|a} \\ U & \to & \underline{aUb|V|W|a} \\ V & \to & \underline{VaV|b} \\ W & \to & \underline{ab} \end{array}$$

(c) Suppose that in the fourth and fifth steps of converting a grammar into Chomsky normal form, you have a rule $A \to BCab$. Show the collection of rules of the form $X \to YZ$ and $X \to c$ that you use to replace this one rule.

$\underline{A \to BX, X \to CY, Y \to Z_1Z_2, Z_1 \to a, Z_2 \to b}$

9. Let G be the grammar

$$\begin{array}{rcl} S & \to & aB|bA|\varepsilon \\ A & \to & aS|bAA \\ B & \to & aBB|bS \end{array}$$

(We have discussed this grammar in class.)

(a) Give a leftmost derivation in G for the string *bbaaab*.

(b) Using the method from class, give a PDA that recognizes the language generated by G.

Answer:

(a) $S \Rightarrow bA \Rightarrow bbAA \Rightarrow bbaSA \Rightarrow bbaA \Rightarrow bbaaS \Rightarrow bbaaaB \Rightarrow bbaaabS \Rightarrow bbaaab$

(b) See Figure 2.

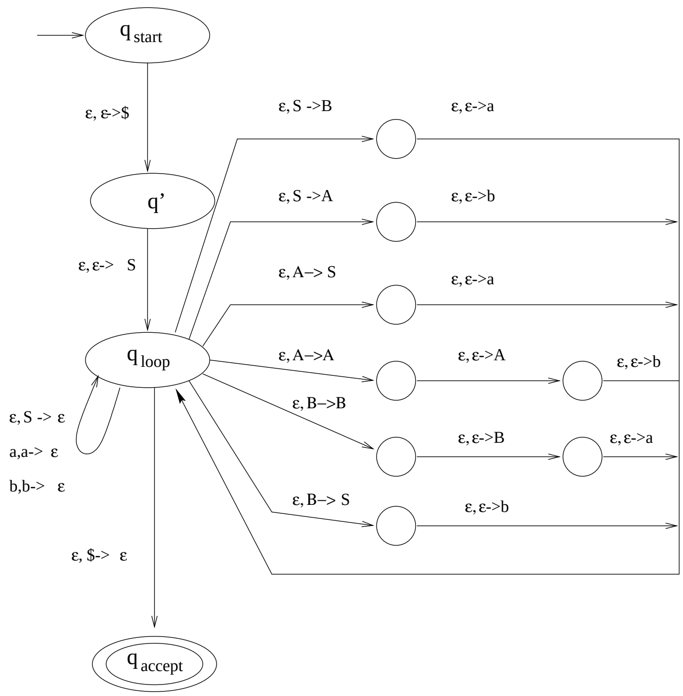
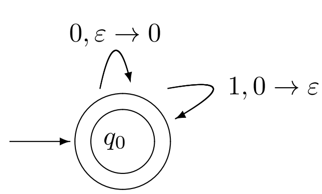


Figure 2: Solution to Problem 7b

10. Let M be the PDA given by:



(a) Does M accept the string ε? Yes

(b) Does M accept the string 01001? Yes

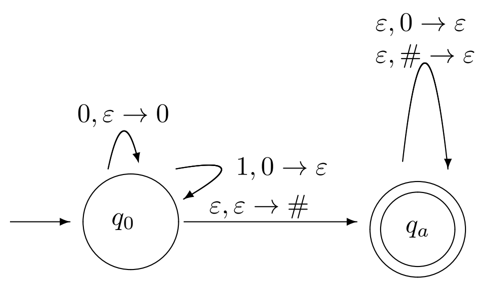
(c) Does M accept the string 01101? No

(d) Describe in words the language recognized by M.
[Hint: You may find the concept of "prefix of a string" as given in Problem 1.32 of the text to be useful.]

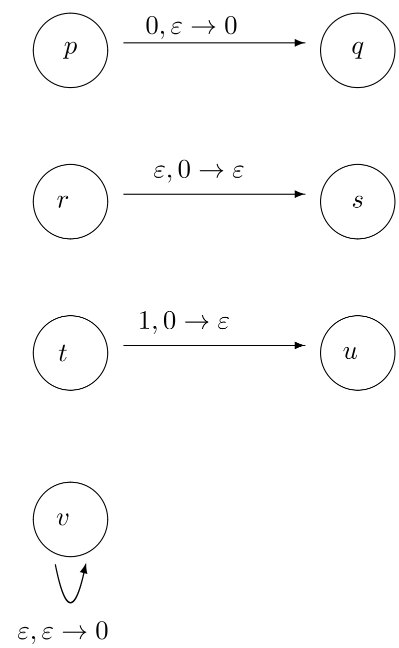
Answer: The set of binary strings w such that each prefix of w has at least as many 0s as 1s.

(e) If you want to convert the PDA M into a context-free grammar, then the first thing you have to do is transform M into another PDA M'. Give this PDA M'. (You are *not* being asked to transform M into a CFG. Just give the PDA M'.)

Answer:



(f) Suppose you are converting a PDA into a CFG and the only moves in the PDA that involve the stack symbol 0 are the following:

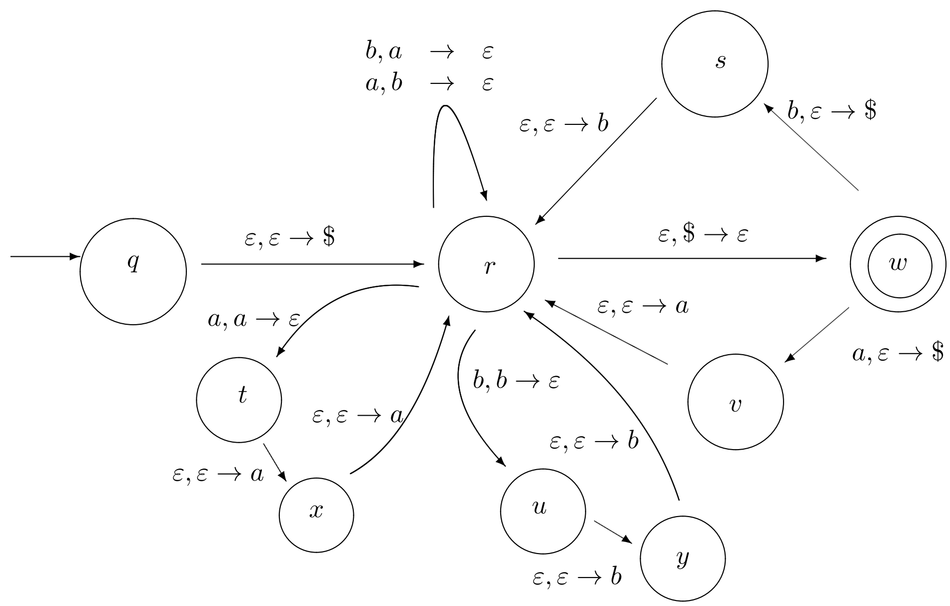


The grammar will contain four rules corresponding to the different ways to push and pop a 0. What are these four rules?

Answer:

$$
\begin{aligned}
A_{ps} &\rightarrow 0A_{qr} \\
A_{pu} &\rightarrow 0A_{qt}1 \\
A_{vs} &\rightarrow A_{vr} \\
A_{vu} &\rightarrow A_{vt}1
\end{aligned}
$$

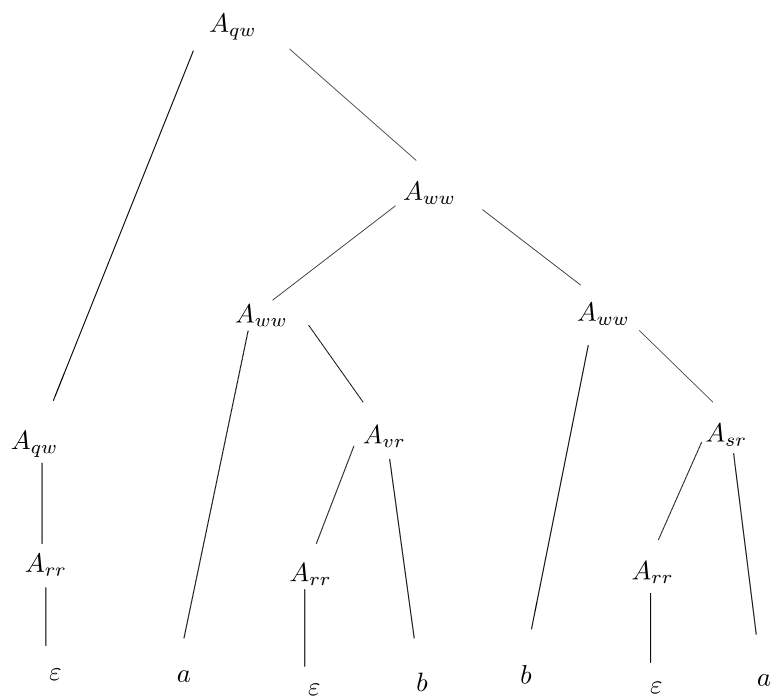
11. Let P be the following PDA. (We discussed a PDA very similar to P in class. P recognizes the language $\{w \in \{a,b\}^* | w$ contains the same number of a's and b's$\}$.)



(a) Show the accepting computation of P on the input string *abba* by filling in the chart below showing the state, tape contents and stack contents after each step. There are 10 steps in the computation and the initial state, tape and stack are filled in.

State	Tape	Stack
q	$abba$	ε
r	$abba$	$\$$
w	$abba$	ε
v	bba	$\$$
r	bba	$a\$$
r	ba	$\$$
w	ba	ε
s	a	$\$$
r	a	$b\$$
r	ε	$\$$
w	ε	ε

(b) The PDA P is "special". Let G be the grammar you obtain from P using the method of Lemma 2.27. Give a parse tree for the string *abba* in G.
[You are not being asked to write out the whole grammar G. You only have to give a parse tree for *abba*. Even if you find this part hard, you should do Part (a), which should be easier.]


A_{qw}
A_{ww}
A_{ww}
A_{ww}
A_{qw}
A_{vr}
A_{sr}
A_{rr}
A_{rr}
A_{rr}
ε
a
ε
b
b
ε
a

12. Classify each of the following languages as being i) regular, or ii) context-free but not regular, or iii) not context-free. Justify your answers.

(a) $\{w\#u | w, u \in \{0,1\}^* \text{ and } |w| = |u|\}$
Classification: ii
Explanation:

The language is context-free because it is generated by the following context-free grammar:

$$\begin{aligned} S &\rightarrow DSD | \# \\ D &\rightarrow 0|1 \end{aligned}$$

We use the Pumping Lemma to show that the language is not regular. We call the language A. Given $p \geq 1$, choose $s = 0^p\#0^p$. Then, $s \in A$ and $|s| \geq p$. Given x, y, z with $s = xyz$, $|xy| \leq p$ and $|y| > 0$, we must have $y = 0^k$ for some $k > 0$. Choose $i = 2$. Then, $xy^iz = 0^{p+k}\#0^p$ which is not in A.

(b) $\{w \in \{0,1\}^* | w \text{ contains an odd number of 0's and exactly one 1}\}$
Classification: i
Explanation:

The language is given by the regular expression:

$$0(00)^*1(00)^* \cup (00)^*10(00)^*$$

(It is also not hard to give an NFA for the language.)

(c) $\{w \in \{0,1,2\}^* | w \text{ contains more 0's than 1's and more 0's than 2's}\}$
Classification: iii
Explanation:

Call the language C. We use the Pumping Lemma for context-free languages to show that C is not context-free. Given $p \geq 1$, choose $s = 0^{p+1}1^p2^p$. Then, $s \in C$ and $|s| \geq p$. Given u, v, x, y, z with $s = uvxyz$, $|vxy| \leq p$ and $|vy| > 0$, we choose i as follows:

Case 1: Either v or y contains a 0. Then choose $i = 0$. In uxz, the number of 0's is reduced, but the number of 2's stays the same (since $|vxy| \leq p$). Thus, uxz has at least as many 2's as 0's, so is not in C.

Case 2: Neither v nor y contains a 0. Then, choose $i = 2$. In $uvvxyyz$, either the number of 1's or the number of 2's (or both) is increased, while the number of 0's stays the same. This means that $uvvxyyz$ is not in C.