Name: _____

1 True/False

Answer whether the following statements are true or false and briefly explain your answer.

a) [TRUE/FALSE] For any NFA N, an equivalent NFA N' can be created that has only one accept state. [5 pts]

True. We can create N' as follows:

(a) Add a new state q_{accept} to Q'

(b) Add an ϵ-transition from each state in F_N to q_{accept}

(c) Let $F = \{q_{accept}\}$

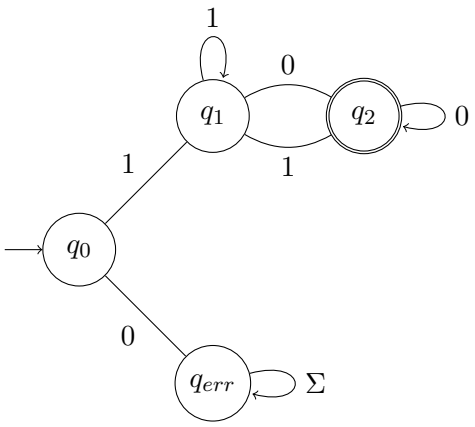
b) [TRUE/FALSE] Converting any context-free grammar to Chomsky normal form will ensure that it is unambiguous. [5 pts]

False. We know that 1) a CFG exists to describe every context-free language, 2) that we can convert any CFG to CNF, and that 3) there exist "inherently ambiguous" languages that can *only* be represented with an ambiguous grammar. Converting a grammar for an inherently ambiguous language to CNF should yield an ambiguous CNF grammar.

2 Finite Automata

Draw a DFA, NFA, or GNFA for each of the following languages.

a) $\{w \in \{0,1\}^* \mid w \text{ begins with a 1 and ends with a 0}\}$ [5 pts]



3 Language Identification

Identify each language below as regular, context-free, or non-context-free. If the language is regular, provide a finite automaton or regular expression for it. If the language is context-free, use the pumping lemma to show that it is *not* regular and provide a context-free grammar or pushdown automaton for it. If the language is not context-free, use the context-free pumping lemma to prove this.

a) $\overline{A}$, where $A = \{0^n 1^n \mid n \geq 0\}$. [10 pts]

$\overline{A}$ is context-free.

Assume that A is regular. Then the pumping lemma holds for A.

Let $s = 0^p 1^p$. Since $|xy| \leq p$, we know that y consists of one or more zeros. Pumping up to $i = 2$ then yields $0^{p+k}1^p$, where $k = |y|$. Since there are more zeros than ones, this string is not in the language. Since the pumping lemma does not hold for A, A is not regular. Since regular languages are closed under complement, $\overline{A}$ is not regular.

We can consider $\overline{A}$ as the union of two languages M and N, where:

$$M = \{0^i 1^j \mid i \neq j\}$$
$$N = \{w \in \{0,1\}^* \mid w \text{ contains a 1 followed by a 0}\}$$

We can create grammars for these languages as follows:

$$S_1 \to 0A \mid B1$$
$$A \to 0A \mid C$$
$$B \to B1 \mid C$$
$$C \to 0C1 \mid \epsilon$$

$$S_2 \to D1D0D$$
$$D \to 0D \mid 1D \mid \epsilon$$

We can now combine these grammars into one that describes the union of M and N:

$$S_0 \to S_1 \mid S_2$$
$$S_1 \to 0A \mid B1$$
$$A \to 0A \mid C$$
$$B \to B1 \mid C$$
$$C \to 0C1 \mid \epsilon$$
$$S_2 \to D1D0D$$
$$D \to 0D \mid 1D \mid \epsilon$$

4 Proofs

a) Show that $\{w \in \{0,1\}^* \mid w \text{ is a binary multiple of } n\}$ is regular for any $n \in \mathbb{N}$. [10 pts]

Proof by Construction

We can create a DFA $D = (Q, \Sigma, \delta, q_{start}, F)$ that decides this language for any arbitrary value of n by having a state for each possible modulus of n. Appending a 0 is equivalent to doubling the current value; appending a 1 is equivalent to doubling the current value and then adding 1.

- $Q = \{q_s\} \cup \{q_i \mid i \in [0, n)\}$
- $\Sigma = \{0, 1\}$
- δ can be described as follows:
  - $\delta(q_s, 0) = q_0$
  - $\delta(q_s, 1) = q_1$
  - For any $i \in [0, n)$:
    - $\delta(q_i, 0) = q_{2i \ \% \ n}$
    - $\delta(q_i, 1) = q_{2i+1 \ \% \ n}$
- $q_{start} = q_s$
- $F = \{q_0\}$

b) Use the context-free pumping lemma to show that $\{a^i b^j c^k \mid 0 \leq i < j < k\}$ is not context-free. [10 pts]

Assume that this language is context-free. Then the context-free pumping lemma holds for it.

Let $s = a^p b^{p+1} c^{p+2}$. Since $|vxy| \leq p$, we can partition the possible values of $uvxyz$ into three cases:

Case 1: Either v or y contains multiple distinct characters

In this case, "pumping up" to $i = 2$ or greater yields a string that is not of the form $a^i b^j c^k$ (characters are out of order)

Case 2: vy contains only one distinct character

- If this character is a, "pump up" to $i = 2$ or greater. We now have at least as many as as bs, so the resulting string is not in the language
- If this character is b or c, "pump down" to $i = 0$. We now have either more more as than bs or more bs than cs, respectively, and the resulting string is not in the language

Case 3: v is composed entirely of one character and y is composed entirely of another

- If these characters are as and bs, "pump up" to $i = 2$ or greater. We now have at least as many bs as cs, and the resulting string is not in the language.
- If these characters are bs and cs, "pump down" to $i = 0$. We now have at least as many as as bs, and the resulting string is not in the language.

Since all of these cases yield a string that is outside the language, the context-free pumping lemma does not hold, and this language is not context-free.