This is a "core dump" of potential questions for Midterm 1. This should give you a good idea of the *types* of questions that we will ask on the exam—in particular, there *will* be a series of True/False questions—but the actual exam questions may or may not appear in this handout. This list intentionally includes a few questions that are too long or difficult for exam conditions; most of these are indicated with a *star.

Questions from past exams are labeled with the semester they were used, for example, ⟨⟨*S14*⟩⟩ or ⟨⟨*F14*⟩⟩. (Questions from old exams *might* reappear on this semester's exams, but they might not.) Questions from this semester's homework are labeled ⟨⟨*HW*⟩⟩. Questions from this semester's labs are labeled ⟨⟨*Lab*⟩⟩.

Induction on Strings

Give complete, formal inductive proofs for the following claims. Your proofs must reply on the formal recursive definitions of the relevant string functions, not on intuition. Recall that the concatenation • and length $|\cdot|$ functions are formally defined as follows:

$$w \bullet y := \begin{cases} y & \text{if } w = \varepsilon \\ a \cdot (x \bullet y) & \text{if } w = ax \text{ for some } a \in \Sigma \text{ and } x \in \Sigma^* \end{cases}$$

$$|w| := \begin{cases} 0 & \text{if } w = \varepsilon \\ 1 + |x| & \text{if } w = ax \text{ for some } a \in \Sigma \text{ and } x \in \Sigma^* \end{cases}$$

1. The ***reversal*** $\boldsymbol{w^R}$ of a string w is defined recursively as follows:

$$w^R := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ x^R \bullet a & \text{if } w = ax \text{ for some } a \in \Sigma \text{ and } x \in \Sigma^* \end{cases}$$

   (a) Prove that $(w \bullet x)^R = x^R \bullet w^R$ for all strings w and x. ⟨⟨***lab, F14***⟩⟩
   (b) Prove that $(w^R)^R = w$ for every string w. ⟨⟨***lab***⟩⟩
   (c) Prove that $|w| = |w^R|$ for every string w. ⟨⟨***lab***⟩⟩
   (d) Prove that $(w^n)^R = (w^R)^n$ for every string w and every integer $n \geq 0$.

2. For any string w and any non-negative integer n, let $\boldsymbol{w^n}$ denote the string obtained by concatenating n copies of w; more formally, define

$$w^n := \begin{cases} \varepsilon & \text{if } n = 0 \\ w \bullet w^{n-1} & \text{otherwise} \end{cases}$$

   For example, $(\mathsf{BLAH})^5 = \mathsf{BLAHBLAHBLAHBLAHBLAH}$ and $\varepsilon^{374} = \varepsilon$.

   (a) Prove that $w^m \bullet w^n = w^{m+n}$ for every string w and all non-negative integers n and m.
   (b) Prove that $(w^m)^n = w^{mn}$ for every string w and all non-negative integers n and m.
   (c) Prove that $|w^n| = n|w|$ for every string w and every integer $n \geq 0$.

3. Consider the following pair of mutually recursive functions:

$$evens(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ odds(x) & \text{if } w = ax \end{cases} \qquad odds(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ a \cdot evens(x) & \text{if } w = ax \end{cases}$$

For example, $evens(\texttt{0001101}) = \texttt{010}$ and $odds(\texttt{0001101}) = \texttt{0011}$.

(a) Prove the following identity for all strings w and x:

$$evens(w \bullet x) = \begin{cases} evens(w) \bullet evens(x) & \text{if } |w| \text{ is even,} \\ evens(w) \bullet odds(x) & \text{if } |w| \text{ is odd.} \end{cases}$$

(b) Prove the following identity for all strings w:

$$evens(w^R) = \begin{cases} (evens(w))^R & \text{if } |w| \text{ is odd,} \\ (odds(w))^R & \text{if } |w| \text{ is even.} \end{cases}$$

(c) Prove that $|w| = |evens(w)| + |odds(w)|$ for every string w.

4. Consider the following recursive function:

$$scramble(w) := \begin{cases} w & \text{if } |w| \le 1 \\ ba \bullet scramble(x) & \text{if } w = abx \text{ for some } a, b \in \Sigma \text{ and } x \in \Sigma^* \end{cases}$$

For example, $scramble(\texttt{00 01 10 1}) = \texttt{00 10 01 1}$.

(a) Prove that $|scramble(w)| = |w|$ for every string w.

(b) Prove that $scramble(scramble(w)) = w$ for every string w.

Regular expressions

For each of the following languages over the alphabet $\{0, 1\}$, give an equivalent regular expression.

1. Every string of length at most 3. *[Hint: Don't try to be clever.]*
2. ***⟨⟨lab⟩⟩*** Every string except 010. *[Hint: Don't try to be clever.]*
3. All strings in which every run of consecutive 0s has even length and every run of consecutive 1s has odd length. ***⟨⟨F14⟩⟩***
4. ***⟨⟨hw⟩⟩*** All strings *not* containing the substring 010.
5. All strings containing the substring 10 or the substring 01.
6. All strings containing either the substring 10 or the substring 01, but not both.
7. All strings containing at least two 1s and at least one 0.
8. All strings containing *either* at least two 1s *or* at least one 0.
9. ***⟨⟨lab⟩⟩*** All strings such that *in every prefix*, the number of 0s and the number of 1s differ by at most 1.
10. The set of all strings in $\{0, 1\}^*$ whose length is divisible by 3.
11. ***⟨⟨S14⟩⟩*** The set of all strings in 0^*1^* whose length is divisible by 3.
12. The set of all strings in $\{0, 1\}^*$ in which the number of 1s is divisible by 3.

Direct DFA construction.

Draw or formally describe a DFA that recognizes each of the following languages. If you draw the DFA you may omit transitions to a reject/junk state.

1. Every string of length at most 3.
2. ⟨⟨*lab*⟩⟩ Every string except 010.
3. The language {LONG, LUG, LEGO, LEG, LUG, LOG, LINGO}.
4. The language $\mathtt{MOO}^* + \mathtt{MEOO}^*\mathtt{W}$
5. All strings in which every run of consecutive 0s has even length and every run of consecutive 1s has odd length.⟨⟨*F14*⟩⟩
6. ⟨⟨*lab*⟩⟩ All strings *not* containing the substring 010.
7. All strings containing the substring 10 or the substring 01.
8. All strings containing either the substring 10 or the substring 01, but not both.
9. The set of all strings in $\{0, 1\}^*$ whose length is divisible by 3.
10. ⟨⟨*S14*⟩⟩ The set of all strings in 0^*1^* whose length is divisible by 3.
11. The set of all strings in $\{0, 1\}^*$ in which the number of 1s is divisible by 3.
12. All strings w such that the binary value of w^R is divisible by 5.
13. ⟨⟨*lab*⟩⟩ All strings such that *in every prefix*, the number of 0s and the number of 1s differ by at most 2.

Fooling sets

Prove that each of the following languages is *not* regular.

1. The set of all strings in $\{0,1\}^*$ with more 0s than 1s. ⟨⟨*S14*⟩⟩
2. The set of all strings in $\{0,1\}^*$ with fewer 0s than 1s.
3. The set of all strings in $\{0,1\}^*$ with exactly twice as many 0s as 1s.
4. The set of all strings in $\{0,1\}^*$ with at least twice as many 0s as 1s.
5. $\{0^{2^n} \mid n \geq 0\}$ ⟨⟨*Lab*⟩⟩
6. $\{0^{F_n} \mid n \geq 0\}$, where F_n is the nth Fibonacci number, defined recursively as follows:

$$F_n := \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F_{n-1} + F_{n-2} & \text{otherwise} \end{cases}$$

   [Hint: If $F_i + F_j$ is a Fibonacci number, then either $i = j \pm 1$ or $\min\{i, j\} \leq 2$.]

*7. $\{0^{n^3} \mid n \geq 0\}$

8. $\{x\#y \mid x, y \in \{0,1\}^* \text{ and } \#(0,x) = \#(1,y)\}$
9. $\{xx^c \mid x \in \{0,1\}^*\}$, where x^c is the *complement* of x, obtained by replacing every 0 in x with a 1 and vice versa. For example, $0001101^c = 1110010$.
10. The language of properly balanced strings of parentheses, described by the context-free grammar $S \rightarrow \varepsilon \mid SS \mid (S)$. ⟨⟨*Lab*⟩⟩
11. $\{(01)^n(10)^n \mid n \geq 0\}$
12. $\{(01)^m(10)^n \mid n \geq m \geq 0\}$
13. $\{w\#x\#y \mid w, x, y \in \Sigma^* \text{ and } w, x, y \text{ are not all equal}\}$

Regular or Not?

For each of the following languages, either prove that the language is regular (by describing a DFA, NFA, or regular expression), or prove that the language is not regular (using a fooling set argument). Unless otherwise specified, all

1. ⟨⟨*F14*⟩⟩ The set of all strings in $\{0,1\}^*$ in which the substrings 01 and 10 appear the same number of times. (For example, the substrings 01 and 01 each appear three times in the string 1100001101101.)

2. ⟨⟨*F14*⟩⟩ The set of all strings in $\{0,1\}^*$ in which the substrings 00 and 11 appear the same number of times. (For example, the substrings 00 and 11 each appear three times in the string 1100001101101.)

3. ⟨⟨*F14*⟩⟩ $\{www \mid w \in \Sigma^*\}$

4. ⟨⟨*F14*⟩⟩ $\{wxw \mid w, x \in \Sigma^*\}$

5. The set of all strings in $\{0,1\}^*$ such that in every prefix, the number of 0s is greater than the number of 1s.

6. The set of all strings in $\{0,1\}^*$ such that in every *non-empty* prefix, the number of 0s is greater than the number of 1s.

7. $\{0^m 1^n \mid 0 \le m - n \le 374\}$

8. $\{0^m 1^n \mid 0 \le m + n \le 374\}$

9. The language generated by the following context-free grammar:

$$S \to 0A1 \mid \varepsilon$$
$$A \to 1S0 \mid \varepsilon$$

10. The language generated by the following context-free grammar:

$$S \to 0S1 \mid 1S0 \mid \varepsilon$$

11. $\{w\#x \mid w, x \in \{0,1\}^* \text{ and no substring of } w \text{ is also a substring of } x\}$

12. $\{w\#x \mid w, x \in \{0,1\}^* \text{ and no } \textit{non-empty} \text{ substring of } w \text{ is also a substring of } x\}$

*13. $\{w\#x \mid w, x \in \{0,1\}^* \text{ and } \textit{every} \text{ non-empty substring of } w \text{ is also a substring of } x\}$

14. $\{w\#x \mid w, x \in \{0,1\}^* \text{ and } w \text{ is a substring of } x\}$

15. $\{w\#x \mid w, x \in \{0,1\}^* \text{ and } w \text{ is a proper substring of } x\}$

16. $\{xy \mid \#(0,x) = \#(1,y) \text{ and } \#(1,x) = \#(0,y)\}$

*17. $\{xy \mid \#(0,x) = \#(1,y) \text{ or } \#(1,x) = \#(0,y)\}$

Product/Subset Constructions

For each of the following languages $L \subseteq \{0, 1\}^*$, formally describe a DFA $M = (Q, \{0, 1\}, s, A, \delta)$ that recognizes L. ***Do not attempt to draw the DFA.*** Instead, give a complete, precise, and self-contained description of the state set Q, the start state s, the accepting state A, and the transition function δ. Do ***not*** just describe several smaller DFAs and write "product construction!"

1. ⟨⟨*S14*⟩⟩ All strings that satisfy *all* of the following conditions:
   (a) the number of 0s is even
   (b) the number of 1s is divisible by 3
   (c) the total length is divisible by 5
2. All strings that satisfy *at least one* of the following conditions: ...
3. All strings that satisfy *exactly one* of the following conditions: ...
4. All strings that satisfy *exactly two* of the following conditions: ...
5. All strings that satisfy *an odd number of* of the following conditions: ...
6. Other possible conditions:
   (a) The number of 0s in w is odd.
   (b) The number of 1s in w is not divisible by 5.
   (c) The length $|w|$ is divisible by 7.
   (d) The binary value of w is divisible by 7.
   (e) The binary value of w^R is not divisible by 7.
   (f) w contains the substring 00
   (g) w does not contain the substring 11
   (h) ww does not contain the substring 101

NFA Construction

Let L be an arbitrary regular language $\Sigma = \{0, 1\}$. Prove that each of the following languages over $\{0, 1\}$ is regular. "Describe" does not necessarily mean "draw".

1. All strings where the 374th symbol from the end is 0.

2. All strings that satisfy *at least one* of the following conditions:

    (a) The number of 0s is even

    (b) The number of 1s is divisible by 3

    (c) The total length is divisible by 5

3. ⟨⟨*lab*⟩⟩ All strings such that *in every prefix*, the number of 0s and the number of 1s differ by at most 2.

4. ⟨⟨*lab*⟩⟩ All strings such that *in every substring*, the number of 0s and the number of 1s differ by at most 2.

Regular Language Transformations

Let L be an arbitrary regular language over the alphabet $\Sigma = \{\mathtt{0}, \mathtt{1}\}$. Prove that each of the following languages over $\{\mathtt{0}, \mathtt{1}\}$ is regular. "Describe" does not necessarily mean "draw".

1. $L^c := \{ w^c \mid w \in L \}$, where w^c is the complement of w, defined recursively as follows:

$$w^c := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ \mathtt{1} \cdot x^c & \text{if } w = \mathtt{0}x \text{ for some string } x \\ \mathtt{0} \cdot x^c & \text{if } w = \mathtt{1}x \text{ for some string } x \end{cases}$$

   For example, $\mathtt{0001101}^c = \mathtt{1110010}$.

2. $\textsc{OneInFront}(L) := \{\mathtt{1}x \mid x \in L\}$

3. $\textsc{OnlyOnes}(L) := \left\{ \mathtt{1}^{\#(\mathtt{1},w)} \mid w \in L \right\}$

4. $\textsc{OnlyOnes}^{-1}(L) := \left\{ w \mid \mathtt{1}^{\#(\mathtt{1},w)} \in L \right\}$

5. $\textsc{MissingFirst}(L) := \{w \in \Sigma^* \mid aw \in L \text{ for some symbol } a \in \Sigma\}$

6. $\textsc{Prefixes}(L) := \{x \mid xy \in L \text{ for some string } x \in \Sigma^*\}$

7. $\textsc{Suffixes}(L) := \{y \mid xy \in L \text{ for some string } y \in \Sigma^*\}$

8. ***⟨⟨lab, F14⟩⟩*** $\textsc{Evens}(L) := \{evens(w) \mid w \in L\}$, where the functions *evens* and *odds* are recursively defined as follows:

$$evens(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ odds(x) & \text{if } w = ax \end{cases} \qquad odds(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ a \cdot evens(x) & \text{if } w = ax \end{cases}$$

   For example, $evens(\mathtt{0001101}) = \mathtt{010}$ and $odds(\mathtt{0001101}) = \mathtt{0011}$.

9. ***⟨⟨lab, F14⟩⟩*** $\textsc{Evens}^{-1}(L) := \{w \mid evens(w) \in L\}$, where the functions *evens* and *odds* are recursively defined as above.

10. $\textsc{Shuffle}(L) := \{ shuffle(w, x) \mid w, x \in L \}$, where the function *shuffle* is defined recursively as follows:

$$shuffle(w, x) := \begin{cases} x & \text{if } w = \varepsilon \\ a \cdot shuffle(x, y) & \text{if } w = ay \text{ for some } a \in \Sigma \text{ and some } y \in \Sigma^* \end{cases}$$

    For example, $shuffle(\mathtt{0001101}, \mathtt{1111}) = \mathtt{0^{1}0^{1}0^{1}1^{1}101}$

11. $\textsc{Scramble}(L) := \{ scramble(w) \mid w \in L \}$, where the function *scramble* is defined recursively as follows:

$$scramble(w) := \begin{cases} w & \text{if } |w| \le 1 \\ ba \cdot scramble(x) & \text{if } w = abx \text{ for some } a, b \in \Sigma \text{ and } x \in \Sigma^* \end{cases}$$

    For example, $scramble(\mathtt{00\,01\,10\,1}) = \mathtt{00\,10\,01\,1}$.

Context-Free Grammars

Construct context-free grammars for each of the following languages, and give a brief explanation of how your grammar works, including the language of each non-terminal.

1. All strings in $\{0,1\}^*$ whose length is divisible by 5.
2. All strings in which the substrings 00 and 11 appear the same number of times.
3. All strings in which the substrings 01 and 01 appear the same number of times.
4. $\{0^n 1^{2n} \mid n \geq 0\}$
5. $\{0^m 1^n \mid n \neq 2m\}$
6. $\{0^i 1^j 2^{i+j} \mid i, j \geq 0\}$
7. $\{0^{i+j} \# 0^j \# 0^i \mid i, j \geq 0\}$
8. $\{0^i 1^j 2^k \mid j \neq i + k\}$
9. $\left\{w \# 0^{\#(0,w)} \,\middle|\, w \in \{0,1\}^*\right\}$
10. $\{0^i 1^j 2^k \mid i = j \text{ or } j = k \text{ or } i = k\}$
11. $\{0^i 1^j 2^k \mid i \neq j \text{ or } j \neq k\}$
12. $\{0^{2i} 1^{i+j} 2^{2j} \mid i, j \geq 0\}$
13. $\left\{x \# y^R \,\middle|\, x, y \in \{0,1\}^* \text{ and } x \neq y\right\}$
14. All strings in $\{0,1\}^*$ that are *not* palindromes.
15. $\{0,1\}^* \setminus \{ww \mid w \in \{0,1\}^*\}$ **⟨⟨*lab*⟩⟩**
16. $\left\{0^n 1^{an+b} \,\middle|\, n \geq 0\right\}$, where a and b are arbitrary natural numbers.
17. $\left\{0^n 1^{an-b} \,\middle|\, n \geq b/a\right\}$, where a and b are arbitrary natural numbers.

Context-Free Grammar Proofs

Each of the following questions describes a language L and a context-free grammar G, and asks you to prove that $L = L(G)$. As always, you must separately prove $L \subseteq L(G)$ and $L(G) \subseteq L$; each proof will proceed by induction.

1. Prove that the following grammar generates the language $\{0^n 1^n \mid n \geq 0\}$.

$$S \to 0S1 \mid \varepsilon$$

2. Prove that the following grammar generates the language $\{0^m 1^n \mid m \leq n\}$.

$$S \to 0S1 \mid \varepsilon \mid S1$$

3. Prove that the following grammar generates the language $\{0^m 1^n \mid n \leq 2m \text{ and } m \leq 2n\}$.

$$S \to 00S1 \mid 0S11 \mid 0S1 \mid \varepsilon$$

4. Prove that the following grammar generates the language $\{0^m 1^n \mid n \leq 2m \text{ and } m \leq 2n\}$.

$$S \to 00S1 \mid 0S11 \mid 0011 \mid 01 \mid \varepsilon$$

5. Prove that the following grammar generates the language $\{0^m 1^n \mid n \leq 2m \text{ and } m \leq 2n\}$.

$$\begin{aligned} S &\to A \mid B \\ A &\to 00A1 \mid 0A1 \mid \varepsilon \\ B &\to 0B11 \mid 0B1 \mid \varepsilon \end{aligned}$$

6. Prove that the following grammar generates the language $\{0^m 1^n \mid n \leq 2m \text{ and } m \leq 2n\}$.

$$\begin{aligned} S &\to A \mid B \\ A &\to 00A1 \mid C \\ B &\to 0B11 \mid C \\ C &\to 0C1 \mid \varepsilon \end{aligned}$$

7. Prove that the following grammar generates the language $\left\{0^m{+}0^n{=}0^{m+n} \;\middle|\; m, n \geq 0\right\}$.

$$\begin{aligned} S &\to {+}A \mid 0S0 \\ A &\to {=} \mid 0A0 \end{aligned}$$

8. Prove that the following grammar generates the language $\left\{0^{2i} 1^{i+j} 0^{2j} \;\middle|\; i, j \geq 0\right\}$.

$$\begin{aligned} S &\to AB \\ A &\to 00S1 \mid \varepsilon \\ B &\to 1S00 \mid \varepsilon \end{aligned}$$

9. *⟨⟨hw⟩⟩* Prove that the following grammar generates the language of all binary strings w such that $\#(0, w) = \#(1, w)$.

$$S \to \varepsilon \mid 0S1S \mid 1S0S$$

True or False (sanity check)

For each statement below, check "True" if the statement is ***always*** true and "False" otherwise. Each correct answer is worth 1 point; each incorrect answer is worth −½ point; checking "I don't know" is worth ¼ point; and flipping a coin is (on average) worth ¼ point.

Read each statement *very* carefully. Some of these are deliberately subtle. On the other hand, you should not spend more than two minutes on any single statement.

Definitions

1. Every language is regular.
2. For all languages L, if L is regular then L can be represented by a regular expression.
3. For all languages L, if L is not regular then L cannot be represented by a regular expression.
4. For all languages L, if L can be represented by a regular expression then L is regular.
5. For all languages L, if L cannot be represented by a regular expression then L is not regular.
6. For all languages L, if there is a DFA that accepts every string in L, then L is regular.
7. For all languages L, if there is a DFA that accepts every string not in L, then L is not regular.
8. For all languages L, if there is a DFA that rejects every string not in L, then L is regular.
9. For all languages L, if for every string $w \in L$ there is a DFA that accepts w, then L is regular. ⟨⟨*S14*⟩⟩
10. For all languages L, if for every string $w \notin L$ there is a DFA that rejects w, then L is regular.
11. For all languages L, if some DFA recognizes L, then some NFA also recognizes L.
12. For all languages L, if some NFA recognizes L, then some DFA also recognizes L.

Closure Properties

1. For all regular languages L and L', the language $L \cap L'$ is regular.
2. For all regular languages L and L', the language $L \cup L'$ is regular.
3. For all regular languages L, the language L^* is regular.
4. For all regular languages A, B, and C, the language $(A \cup B) \setminus C$ is regular.
5. For all languages $L \subseteq \Sigma^*$, if L is regular, then $\Sigma^* \setminus L$ is regular.
6. For all languages $L \subseteq \Sigma^*$, if L is regular, then $\Sigma^* \setminus L$ is not regular.
7. For all languages $L \subseteq \Sigma^*$, if L is not regular, then $\Sigma^* \setminus L$ is regular.
8. For all languages $L \subseteq \Sigma^*$, if L is not regular, then $\Sigma^* \setminus L$ is not regular.
9. ⟨⟨*S14*⟩⟩ For all languages L and L', the language $L \cap L'$ is regular.

10. ⟨⟨*F14*⟩⟩ For all languages L and L', the language $L \cup L'$ is regular.
11. For all languages L, the language L^* is regular. ⟨⟨*F14*⟩⟩
12. For all languages L, if L^* is regular, then L is regular.
13. For all languages A, B, and C, the language $(A \cup B) \setminus C$ is regular.
14. For all languages L, if L is finite, then L is regular.
15. For all languages L and L', if L and L' are finite, then $L \cup L'$ is regular.
16. For all languages L and L', if L and L' are finite, then $L \cap L'$ is regular.
17. For all languages L, if L contains a finite number of strings, then L is regular.
18. For all languages $L \subseteq \Sigma^*$, if L contains infinitely many strings in Σ^*, then L is not regular.
19. ⟨⟨*S14*⟩⟩ For all languages $L \subseteq \Sigma^*$, if L contains all but a finite number of strings of Σ^*, then L is regular.
20. For all languages $L \subseteq \{0, 1\}^*$, if L contains a finite number of strings in 0^*, then L is regular.
21. For all languages $L \subseteq \{0, 1\}^*$, if L contains a all but a finite number of strings in 0^*, then L is regular.
22. If L and L' are not regular, then $L \cap L'$ is not regular.
23. If L and L' are not regular, then $L \cup L'$ is not regular.
24. If L is regular and $L \cup L'$ is regular, then L' is regular. ⟨⟨*S14*⟩⟩
25. If L is regular and $L \cup L'$ is not regular, then L' is not regular. ⟨⟨*S14*⟩⟩
26. If L is not regular and $L \cup L'$ is regular, then L' is regular.
27. If L is regular and $L \cap L'$ is regular, then L' is regular.
28. If L is regular and $L \cap L'$ is not regular, then L' is not regular.
29. If L is regular and L' is finite, then $L \cup L'$ is regular. ⟨⟨*S14*⟩⟩
30. If L is regular and L' is finite, then $L \cap L'$ is regular.
31. If L is regular and $L \cap L'$ is finite, then L' is regular.
32. If L is regular and $L \cap L' = \varnothing$, then L' is not regular.
33. If L is regular and L' is not regular, then $L \cap L' = \varnothing$.
34. If $L \subseteq L'$ and L is regular, then L' is regular.
35. If $L \subseteq L'$ and L' is regular, then L is regular. ⟨⟨*F14*⟩⟩
36. If $L \subseteq L'$ and L is not regular, then L' is not regular.

37. If $L \subseteq L'$ and L' is not regular, then L is not regular. ⟨⟨*F14*⟩⟩

38. For all languages $L \subseteq \Sigma^*$, if L cannot be described by a regular expression, then some DFA accepts $\Sigma^* \setminus L$.

39. For all languages $L \subseteq \Sigma^*$, if no DFA accepts L, then the complement $\Sigma^* \setminus L$ can be described by a regular expression.

40. Every context-free language is regular. ⟨⟨*F14*⟩⟩

41. Every regular language is context-free.

Equivalence Classes. Recall that for any language $L \subset \Sigma^*$, two strings $x, y \in \Sigma^*$ are equivalent with respect to L if and only if, for every string $z \in \Sigma^*$, either both xz and yz are in L, or neither xz nor yz is in L. We denote this equivalence by $x \equiv_L y$.

1. For all languages L, if L is regular, then $\equiv_L$ has finitely many equivalence classes.

2. For all languages L, if L is not regular, then $\equiv_L$ has infinitely many equivalence classes. ⟨⟨*S14*⟩⟩

3. For all languages L, if $\equiv_L$ has finitely many equivalence classes, then L is regular.

4. For all languages L, if $\equiv_L$ has infinitely many equivalence classes, then L is not regular.

5. For all regular languages L, each equivalence class of $\equiv_L$ is a regular language.

*6. For all languages L, each equivalence class of $\equiv_L$ is a regular language.

Fooling Sets

1. For all languages L, if L has an infinite fooling set, then L is not regular.

2. For all languages L, if L has an finite fooling set, then L is regular.

3. For all languages L, if L does not have an infinite fooling set, then L is regular.

4. For all languages L, if L is not regular, then L has an infinite fooling set.

5. For all languages L, if L is regular, then L has no infinite fooling set.

6. For all languages L, if L is not regular, then L has no finite fooling set. ⟨⟨*F14*⟩⟩

Specific Languages (Gut Check). Do *not* construct complete DFAs, NFAs, regular expressions, or fooling-set arguments for these languages. You don't have time.

1. $\{0^i 1^j 2^k \mid i + j - k = 374\}$ is regular. ⟨⟨*S14*⟩⟩

2. $\{0^i 1^j 2^k \mid i + j - k \leq 374\}$ is regular.

3. $\{0^i 1^j 2^k \mid i + j + k = 374\}$ is regular.

4. $\{0^i 1^j 2^k \mid i + j + k > 374\}$ is regular.

5. $\{0^i 1^j \mid i < 374 < j\}$ is regular. ⟨⟨*S14*⟩⟩
6. $\left\{0^m 1^n \;\middle|\; 0 \le m + n \le 374\right\}$ is regular. ⟨⟨*F14*⟩⟩
7. $\left\{0^m 1^n \;\middle|\; 0 \le m - n \le 374\right\}$ is regular. ⟨⟨*F14*⟩⟩
8. $\{0^i 1^j \mid (i - j)$ is divisible by 374$\}$ is regular. ⟨⟨*S14*⟩⟩
9. $\{0^i 1^j \mid (i + j)$ is divisible by 374$\}$ is regular.
10. $\left\{0^{n^2} \;\middle|\; n \ge 0\right\}$ is regular.
11. $\left\{0^{37n+4} \;\middle|\; n \ge 0\right\}$ is regular.
12. $\left\{0^n 1 0^n \;\middle|\; n \ge 0\right\}$ is regular.
13. $\left\{0^m 1 0^n \;\middle|\; m \ge 0 \text{ and } n \ge 0\right\}$ is regular.
14. $\{w \in \{0,1\}^* \mid |w|$ is divisible by 374$\}$ is regular.
15. $\{w \in \{0,1\}^* \mid w$ represents a integer divisible by 374 in binary$\}$ is regular.
16. $\{w \in \{0,1\}^* \mid w$ represents a integer divisible by 374 in base 473$\}$ is regular.
17. $\left\{w \in \{0,1\}^* \;\middle|\; |\#(0,w) - \#(1,w)| < 374\right\}$ is regular.
18. $\left\{w \in \{0,1\}^* \;\middle|\; |\#(0,x) - \#(1,x)| < 374 \text{ for every prefix } x \text{ of } w\right\}$ is regular.
19. $\left\{w \in \{0,1\}^* \;\middle|\; |\#(0,x) - \#(1,x)| < 374 \text{ for every substring } x \text{ of } w\right\}$ is regular.
20. $\left\{w 0^{\#(0,w)} \;\middle|\; w \in \{0,1\}^*\right\}$ is regular.
21. $\left\{w 0^{\#(0,w) \bmod 374} \;\middle|\; w \in \{0,1\}^*\right\}$ is regular.

Automata Transformations

1. Let M be a DFA over the alphabet Σ. Let M' be identical to M, except that accepting states in M are non-accepting in M' and vice versa. Each string in Σ^* is accepted by exactly one of M and M'.
2. Let M be an NFA over the alphabet Σ. Let M' be identical to M, except that accepting states in M are non-accepting in M' and vice versa. Each string in Σ^* is accepted by exactly one of M and M'.
3. If a language L is recognized by a DFA with n states, then the complementary language $\Sigma^* \setminus L$ is recognized by a DFA with at most $n + 1$ states.
4. If a language L is recognized by an NFA with n states, then the complementary language $\Sigma^* \setminus L$ is recognized by a NFA with at most $n + 1$ states.
5. If a language L is recognized by a DFA with n states, then L^* is recognized by a DFA with at most $n + 1$ states.
6. If a language L is recognized by an NFA with n states, then L^* is also recognized by a NFA with at most $n + 1$ states.

Language Transformations

1. For every regular language L, the language $\{w^R \mid w \in L\}$ is also regular.
2. For every language L, if the language $\{w^R \mid w \in L\}$ is regular, then L is also regular. **⟨⟨*F14*⟩⟩**
3. For every language L, if the language $\{w^R \mid w \in L\}$ is not regular, then L is also not regular. **⟨⟨*F14*⟩⟩**
4. For every regular language L, the language $\{w \mid ww^R \in L\}$ is also regular. **⟨⟨*hw*⟩⟩**
5. For every regular language L, the language $\{ww^R \mid w \in L\}$ is also regular.
6. For every language L, if the language $\{w \mid ww^R \in L\}$ is regular, then L is also regular. *[Hint: Consider the language $L = \{0^n 1^n \mid n \geq 0\}$.]*
7. For every regular language L, the language $\{0^{|w|} \mid w \in L\}$ is also regular.
8. For every language L, if the language $\{0^{|w|} \mid w \in L\}$ is regular, then L is also regular.