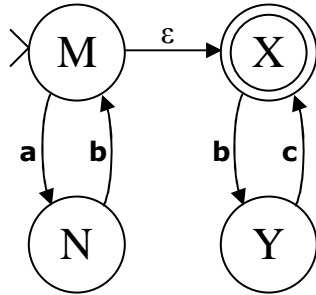


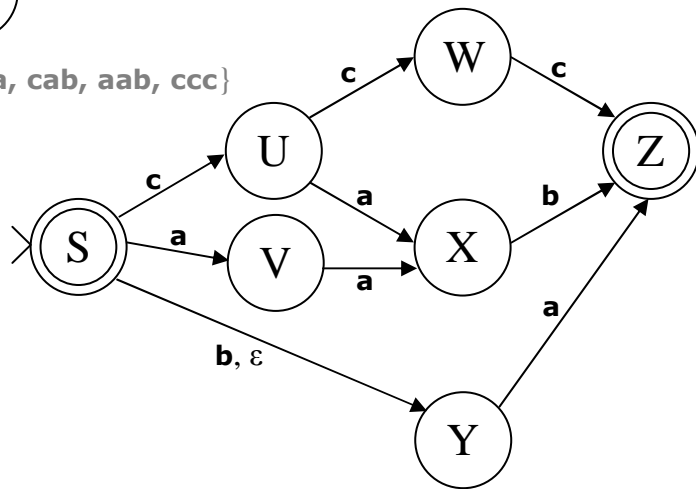
LING 106: Homework 5

1. CONSTRUCTING NFAS

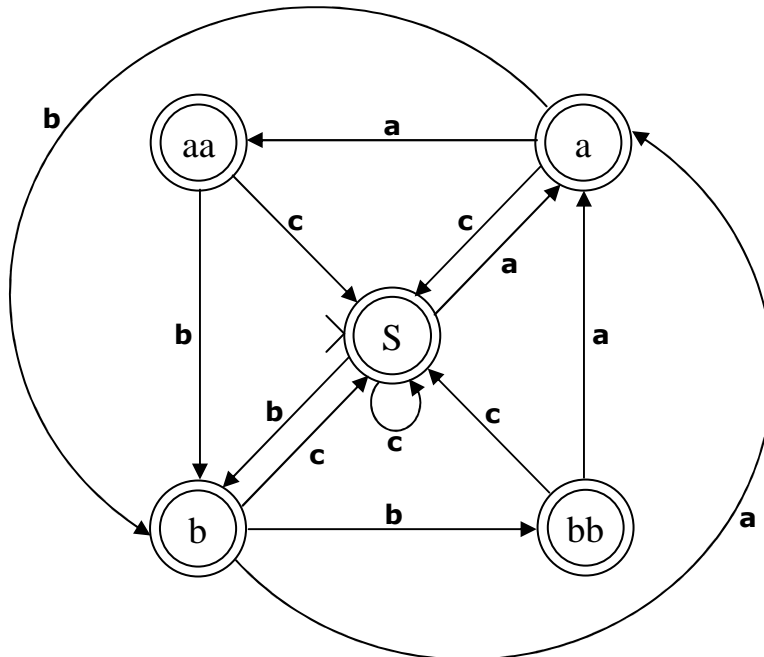
a. The language $(ab)^*(bc)^*$



b. The language $\{\epsilon, a, ba, cab, aab, ccc\}$



c. The language $\{s \mid s \text{ contains neither three } a\text{s in a row nor three } b\text{s in a row}\}$

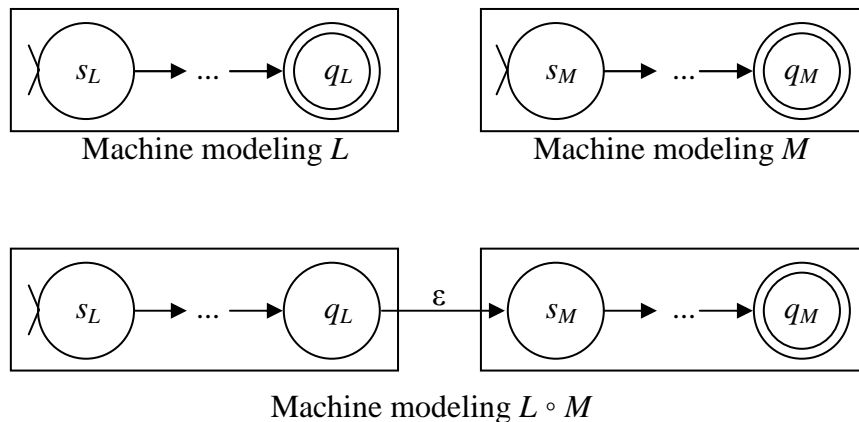


2. TWO PROOFS

a. Prove that regular languages are closed under concatenation.

- Given: L and M are regular.
- Thus: L is modeled by a FSA $\langle Q_L, \Sigma_L, \delta_L, s_L, F_L \rangle$, and M is modeled by $\langle Q_M, \Sigma_M, \delta_M, s_M, F_M \rangle$.
- Then the language LM is modeled by:
 - $\Sigma_{LM} = \Sigma_L = \Sigma_M$
 - $Q_{LM} = Q_L \cup Q_M$
 - $s_{LM} = s_L$
 - $F_{LM} = F_M$
 - δ_{LM} is defined as follows:
 - $\delta_{LM}(\langle q, x \rangle) = \delta_M(\langle q, x \rangle)$ for all $\langle q, x \rangle$ in $Q_M \times (\Sigma_M \cup \{\varepsilon\})$
 - $\delta_{LM}(\langle q, \varepsilon \rangle) = \delta_L(\langle q, \varepsilon \rangle) \cup \{s_M\}$ for all q in F_L
 - $\delta_{LM}(\langle q, x \rangle) = \delta_L(\langle q, x \rangle)$ for all other $\langle q, x \rangle$ in $Q_L \times (\Sigma_L \cup \{\varepsilon\})$

This machine has the same alphabet as the two original machines; it starts where L starts and ends where M ends, using all the same transitions; with the addition of an empty transition from anywhere L can end to the start of M .



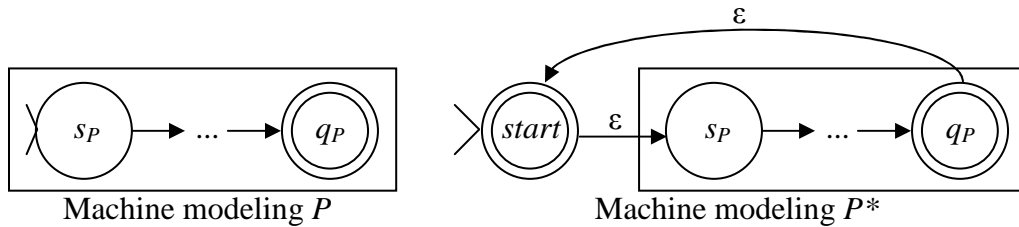
This is a FSA, and it accepts any string that starts with a string L accepts and then goes to a string M accepts. This is $L \circ M$.

- Thus, there is a FSA that models $L \circ M$. So $L \circ M$ is regular.

b. Prove that regular languages are closed under the star operator.

- Given: P is regular.
- Thus: P is modeled by a FSA $\langle Q_P, \Sigma_P, \delta_P, s_P, F_P \rangle$.
- Then the language P^* is modeled by:
 - $\Sigma_{P^*} = \Sigma_P$
 - $Q_{P^*} = Q_P \cup \{start\}$
 - $s_{P^*} = start$
 - $F_{P^*} = F_P \cup \{start\}$
 - δ_{P^*} is defined as follows:
 - $\delta_{P^*}(\langle q, \epsilon \rangle) = \delta_P(\langle q, \epsilon \rangle) \cup \{start\}$ for all q in F_P
 - $\delta_{P^*}(\langle q, x \rangle) = \delta_P(\langle q, x \rangle)$ for all other $\langle q, x \rangle$ in $Q_L \times (\Sigma_L \cup \{\epsilon\})$
 - $\delta_{P^*}(\langle start, \epsilon \rangle) = \{s_P\}$
 - $\delta_{P^*}(\langle start, x \rangle) = \emptyset$ for all x in Σ_P

This machine accepts the empty string. It also accepts any string in P , after which the machine can optionally go back to its start and add another string in P .



This is the language P^* .

- Thus, there is a FSA that models P^* . So P^* is regular.

b. Prove that regular languages are closed under the star operator.

- Given: P is regular.
- Thus: P is modeled by a FSA $\langle Q_P, \Sigma_P, \delta_P, s_P, F_P \rangle$.
- Then the language P^* is modeled by:
 - $\Sigma_{P^*} = \Sigma_P$
 - $Q_{P^*} = Q_P$
 - $s_{P^*} = s_P$
 - $F_{P^*} = s_P$
 - δ_{P^*} is defined as follows:
 - $\delta_{P^*}(\langle q, \epsilon \rangle) = \delta_P(\langle q, \epsilon \rangle) \cup \{s_P\}$ for all q in F_P
 - $\delta_{P^*}(\langle q, x \rangle) = \delta_P(\langle q, x \rangle)$ for all other $\langle q, x \rangle$ in $Q_L \times (\Sigma_L \cup \{\epsilon\})$

This machine accepts the empty string. It also accepts any string in P , after which the machine can optionally go back to its start and add another string in P .



This is the language P^* .

- Thus, there is a FSA that models P^* . So P^* is regular.

3. THE PUMPING LEMMA

a. Given $L = \{s \mid s \text{ contains at most three } \mathbf{1s}\}$, show that the following strings in L obey the pumping lemma, with $p = 4$: **10101**, **1110**, **111**.

- For **10101**: let $x = \mathbf{1}$, $y = \mathbf{0}$, $z = \mathbf{101}$. Then $xz = \mathbf{1101} \in L$, $xyyz = \mathbf{100101} \in L$, $xyyyz = \mathbf{1000101} \in L$, and so forth. Repeating y won't change the number of **1s**, so the resulting string will still be in L . Thus, this string has a cutting which gives a y , where $y \neq \varepsilon$ and $|xy| = 2 \leq p$. Therefore, it obeys the Pumping Lemma.
- For **1110**: let $x = \mathbf{111}$, $y = \mathbf{0}$, $z = \varepsilon$. Then $xz = \mathbf{111} \in L$, $xyyz = \mathbf{11100} \in L$, and so on—once again, repeating y won't change the number of **1s**, so the resulting string will still be in L ; and $y \neq \varepsilon$ and $|xy| = 4 \leq p$. Therefore, it obeys the Pumping Lemma.
- The length of **111** is 3, which is less than the pumping length p . The Lemma states that all strings of length at least p have a certain property; a string of length less than p can't contradict this. Therefore, it obeys the Pumping Lemma.

b. Given $\Sigma = \{\mathbf{a}, \mathbf{b}\}$ and $L = \{s \mid s \text{ contains at most fifteen } \mathbf{as} \text{ and at most fifteen } \mathbf{bs}\}$, explain why L obeys the pumping lemma.

- Note that L is a finite language—it contains no strings over length 30, and as such has a finite number of strings. (If I did the math right, it contains 601,080,389.) That might cause some concern, because you can't take a substring out of any of these strings and repeat it endlessly; at some point, you'll get a string of at least 31 symbols.

The Pumping Lemma, however, guarantees the following for a regular language: **there exists a p such that, for any string of length at least p** , a particular set of facts hold. Therefore, take $p = 31$ (or 225, or 1000, or anything else greater than 30). It's certainly the case that every string of length 31 in L satisfies the particular requirements of the lemma (you certainly can't find a string that long in L that doesn't satisfy them).

c. Prove that $L = \mathbf{xy}^n\mathbf{xy}^n\mathbf{y}^n$, where $n \geq 0$, is not regular.

Suppose that L is regular. Then there is a length p such that all strings in L of that length have a valid cutting into x , y , and z .

Then, in particular, the string $\mathbf{xy}^{p-1}\mathbf{xy}^{p-1}\mathbf{y}^{p-1}$, whatever p might be, has a valid cutting, in which y is found in the first p characters, i.e. \mathbf{xy}^{p-1} . However, in that cutting, y will have...

- ...only \mathbf{ys} in it. In that case, repeating y will make the string start with \mathbf{x} followed by some larger number of \mathbf{ys} , which won't be the same as the number of \mathbf{ys} and \mathbf{xs} later in the string. So that can't be a valid cutting.
- ... \mathbf{xs} and \mathbf{ys} . The only \mathbf{x} in the first p characters is the starting one, so pumping y even once will give $[\mathbf{xy}\dots\mathbf{y}][\mathbf{xy}\dots\mathbf{y}]\mathbf{y}\dots\mathbf{yxy}^{p-1}\mathbf{y}^{p-1}$, which has too many separate substrings of \mathbf{xs} to be in L , so that can't be a valid cutting.
- ...only \mathbf{xs} in it; but again, there's only one \mathbf{x} in the first p characters, and repeating that will give a string that starts with more than one \mathbf{x} , which isn't a string in L . So that, too, can't be a valid cutting.

The remaining possibility is that y contains neither \mathbf{ys} nor \mathbf{xs} in it, but in that case $y = \epsilon$, which is also not a valid cutting. Therefore, there is a string in L that, no matter what the pumping length is, cannot be pumped. If our assumption that L is regular were correct, this would contradict the Pumping Lemma. Therefore, that assumption is wrong and L is not regular.

4. SET THEORY REVISITED

Given the set of strings B :

$$B = \{\mathbf{y}, \mathbf{zy}, \mathbf{yzy}, \mathbf{zyzy}, \mathbf{yzyzy}, \mathbf{zyzyzy}, \mathbf{yzyzyzy}, \mathbf{zyzyzyzy}, \dots\}$$

PART ONE: Express B in predicate notation.

$$B = \{s \mid s \text{ alternates } \mathbf{z} \text{ and } \mathbf{y}, \text{ and ends with } \mathbf{y}\}$$

PART TWO: Express B in recursive notation.

$$\begin{aligned} &\mathbf{y} \in B, \\ &\mathbf{zy} \in B, \\ &\text{if } a \in B, \text{ then } a \circ \mathbf{zy} \in B, \\ &\text{Nothing else is in } B. \end{aligned}$$

5. FUNCTIONS

5.1. First Question

- \mathbb{N} = the set of “counting numbers”, $\{0, 1, 2, 3, 4, 5, \dots\}$
- L = the set of all people currently living
- P = the set of all people, living or dead

ONE: $f \subseteq \mathbb{N} \times \mathbb{N}$, where $f(\langle x, y \rangle)$ if $x = y$.

f is a function, because each number is equal only to itself: each number is thus used once, and only once. For the same reason, the function is also **one-to-one** and **onto**.

TWO: $g \subseteq \mathbb{N} \times \mathbb{N}$, where $g(\langle x, y \rangle)$ if $y = x^2$.

f is a function, because every number in \mathbb{N} has exactly one square in \mathbb{N} . It's **one-to-one**, because nothing is the square of more than one integer (since negative numbers aren't included); but it's **into**, not onto, because some numbers (e.g., 2) aren't the square of any counting number.

THREE: $h \subseteq \mathbb{N} \times \mathbb{N}$, where $g(\langle x, y \rangle)$ if $x = y^2$.

h is not a function. There is not an ordered pair for each element of the domain (e.g., there's no ordered pair starting with "2").

FOUR: $m \subseteq (\mathbb{N} \times \mathbb{N}) \times \mathbb{N}$, where $m(\langle \langle x, y \rangle, z \rangle)$ if $x + y = z$

m is a function: for any ordered pair $\langle x, y \rangle$, there's only one result of adding them together. It's **onto** (because every natural number n will be mapped to by some pair, e.g. $\langle n, 0 \rangle$), but it's **not one-to-one** (because different pairs produce the same sum).

FIVE: $\varphi \subseteq L \times P$, where $\varphi(\langle x, y \rangle)$ if y is the mother of x

φ is *not* a function: my friends A and L are each the mother of all three of their daughters. Thus, φ maps some individuals x to more than one y .

If you read "mother" as "woman who contributed genetic material", however, φ is a function, as far as I understand the current limits of biology. This function is **neither one-to-one** (more than one person may have the same mother) **nor onto** (not everyone in the world is a mother).

SIX: $\alpha \subseteq \mathbb{N} \times L$, where $\alpha(\langle x, y \rangle)$ if x is the age of y , in years (rounded off)

α is not a function: "19", for instance, maps to anyone who's 19 years old.

SEVEN: $\beta \subseteq L \times \mathbb{N}$, where $\beta(\langle x, y \rangle)$ if y is the age of x , in years (rounded off)

β is a function: every living person has an age. It's **neither one-to-one** (more than one person may have the same age) **nor onto** (no number over 150 is mapped to).

6. THE δ FUNCTION

ONE: $\delta(\langle q_0, \mathbf{0} \rangle) = q_1$
 $\delta(\langle q_1, \mathbf{1} \rangle) = q_0$
 $\delta(\langle q_0, \mathbf{1} \rangle) = q_1$
 $\delta(\langle q_1, \mathbf{0} \rangle) = \text{dead}$

Not a transition function: if δ maps something to *dead*, then *dead* is a state, and the machine must have transitions for $\langle \text{dead}, \mathbf{0} \rangle$ and $\langle \text{dead}, \mathbf{1} \rangle$ as well. An easy change is to add $\delta(\langle \text{dead}, \mathbf{0} \rangle) = \text{dead}$ and $\delta(\langle \text{dead}, \mathbf{1} \rangle) = \text{dead}$.

TWO: $\delta(\mathbf{0}) = q_1$
 $\delta(\mathbf{1}) = q_0$
 $\delta(\mathbf{2}) = q_1$
 $\delta(\mathbf{3}) = q_0$

Not remotely a transition function: this isn't a mapping from state/symbol pairs to states, it's a mapping from symbols to states.

THREE: $\delta(\langle q_0, \mathbf{a} \rangle) = q_0$
 $\delta(\langle q_1, \mathbf{a} \rangle) = q_0$
 $\delta(\langle q_1, \mathbf{x} \rangle) = q_0$
 $\delta(\langle q_0, \mathbf{x} \rangle) = q_1$

This can be a transition function.

FOUR: $\delta(\langle \text{start}, \# \rangle) = \text{whoosh}$
 $\delta(\langle \text{start}, \$ \rangle) = \text{tweet}$
 $\delta(\langle \text{start}, \& \rangle) = \text{start}$
 $\delta(\langle \text{whoosh}, \# \rangle) = \text{whoosh}$
 $\delta(\langle \text{whoosh}, \$ \rangle) = \text{start}$
 $\delta(\langle \text{tweet}, \# \rangle) = \text{tweet}$
 $\delta(\langle \text{tweet}, \& \rangle) = \text{start}$

Not a transition function: the *whoosh* state needs a **&** transition, and the *tweet* state needs a **\$** transition. Adding these is easy: $\delta(\langle \text{whoosh}, \& \rangle) = \text{start}$, and $\delta(\langle \text{tweet}, \$ \rangle) = \text{start}$, for instance.

FIVE: $\delta(\langle q_0, \mathbf{t} \rangle) = q_0$
 $\delta(\langle q_0, \mathbf{h} \rangle) = q_1$
 $\delta(\langle q_0, \mathbf{e} \rangle) = q_2$
 $\delta(\langle q_1, \mathbf{t} \rangle) = q_2$
 $\delta(\langle q_1, \mathbf{h} \rangle) = q_1$
 $\delta(\langle q_1, \mathbf{e} \rangle) = q_1$
 $\delta(\langle q_1, \mathbf{t} \rangle) = q_1$
 $\delta(\langle q_2, \mathbf{h} \rangle) = q_0$
 $\delta(\langle q_2, \mathbf{e} \rangle) = q_2$

Not a transition function: $\delta(\langle q_1, \mathbf{t} \rangle)$ has two values, and $\delta(\langle q_2, \mathbf{t} \rangle)$ has none. It can be made into a transition function by changing $\delta(\langle q_1, \mathbf{t} \rangle) = q_1$ to $\delta(\langle q_2, \mathbf{t} \rangle) = q_1$.