Summary and References

Chapter 10
Languages and Machines

SD Languages

D Languages

Context-Free Languages

Regular Languages

FSMs

PDAs

Turing Machines
Well I'm back at work again. I passed your class with a B. I was glad it was over. Thanks!

But, today at work, I decided I was going to teach myself how to use a new performance monitoring tool. So the first thing to do is go read the documentation on the tool, right? You'll never believe what the FIRST paragraph said....

"...In order to understand the complexities of (the tool), imagine a finite state machine. The vertices are the possible states in which a thread can exist. Edges between the vertices represent (functions) to change from one state to another. Now this FSM will have to be non- deterministic because..." blah, blah, blah.....

I couldn't believe it! But thanks to you, I could visualize it and understood completely. My sincere thanks!!! See you around. :)
A Regular Expression

Given the following DFSM $M$:

Give a regular expression for $\neg L(M)$
Is $L$ Regular?

$L = \{w \in \{Y, N\}^* : w \text{ contains at least two } Y \text{'s and at most two } N \text{'s}\}$
Is $L$ Regular?

$L = \{w : w$ is a natural number whose digits appear in a non-decreasing order without leading zeros$\}$
Is $L$ Regular?

$L = \{w \in \{a, b, c, d\}^*: \text{if there are any } a \text{'s in } w, \text{ then there is at least one } c, \text{ and if there are any } b \text{'s, then there is at least one } d\}$
Is $L$ Regular?

$L = \{w \in \{a, b\}^* : \text{for each prefix } x \text{ of } w, \#_a(x) \geq \#_b(x)\}$
Is $L$ Regular?

$\Sigma = \{0 - 9, *, =\}$

$L = \{w = a*b=c: \quad a, b, c \in \{0-9\}^+ \text{ and } \text{int}(a) \times \text{int}(b) = \text{int}(c)\}$
Is $L$ Regular?

$L = \{a^m b^n : |n - j| = 2\}$
Is $L$ Regular?

$L = \{a^n b^m : 0 \leq n, 0 \leq m, \text{ and } n + m \text{ is even} \}$
Is $L$ Regular?

$L = \{w \in \{a, b, c\}^* : (|w| \text{ is even}) \rightarrow (w \text{ contains an even number of } a\text{'s})\}$
Is $L$ Regular?

$L = \{a^i b^j : i, j \geq 0 \text{ and } i \neq j\}$
Is $L$ Regular?

$L = \{w \in \{a, b, c\}^* : \text{in each prefix } x \text{ of } w, \newline \quad \#_a(x) = \#_b(x) = \#_c(x)\}\}$

$L = \{w \in \{a, b, c\}^* : \exists \text{ some prefix } x \text{ of } w, \newline \quad \#_a(x) = \#_b(x) = \#_c(x)\}\}$

$L = \{w \in \{a, b, c\}^* : \exists \text{ some prefix } x \text{ of } w, \newline \quad x \neq \varepsilon \text{ and } \#_a(x) = \#_b(x) = \#_c(x)\}\}$
Is $L$ Regular?

$L = \{uww^Rv : u, v, w \in \{a, b\}^+\}$
Is $L$ Regular?

$L = \{xyzy^Rx : x, y, z \in \{a, b\}^+\}$
Is $L$ Regular?

$$L = \{ w = x_1^n : x \in \{a \cup b \cup c\}^* $$
and $|x| = n$
and $\#_b(x) = n/2 \}$$
Is $L$ Regular?

Let $L = \{w \in \{a, b\}^* : \#_a(w) = \#_b(w)\}$.

Is $L$ regular?

Is $L^*$ regular?
Is $L$ Regular?

Let $L = \{a^p : p \text{ is prime}\}$.

Is $L$ regular?

Is $L^*$ regular?
True or False

(a) There are uncountably many non-regular languages over \( \Sigma = \{a, b\} \).

(b) \( L_1 = L_2 \) if and only if \( L_1^* = L_2^* \).

(c) \( (\emptyset \cup \emptyset^*) \cap (\overline{\emptyset} - (\emptyset\emptyset^*)) = \emptyset \)

(d) Every infinite language is the complement of a finite language.
True or False

(a) The union of an infinite number of regular languages is necessarily regular.

(b) The intersection of an infinite number of regular languages is necessarily regular?
The Function *Middle*

For any language $L$ over $\Sigma$, we can define the function $middle(L)$ as follows:

$$middle(L) = \{ x : \exists y, z \in \Sigma^* (yxz \in L) \}$$

If $L$ is regular, is $middle(L)$ necessarily regular?
Consider any function $f(L_1) = L_2$, where $L_1$ and $L_2$ are both languages over the alphabet $\Sigma = \{0,1\}$. We say that function $f$ is **nice** iff $L_2$ is regular iff $L_1$ is regular.

Examples:

- $f(L) = L^R$ is nice.

- $f(L)$ replaces all 1’s with 0’s (leaving 0’s unchanged)

- $f(L) = L \cup 0^*$
Are these Functions Nice?

\( f(L) \) is the language formed from \( L \) by changing every 0 to 1 and every 1 to 0 (simultaneously).

\( f(L) \) is the language formed by appending 11 to the end of every string in \( L \).

\( f(L) \) is the language formed from \( L \) by throwing away \( \varepsilon \), and deleting the first positions of all other strings.

middle(\( L \))