Office hours:

Emily : Thur 4-5pm (conceptual) Siebel basement

Fri 11am-noon (general)

Ruta : 1:30-2:30 Thur (general)

Homework parties: Sat 2-4

Sun 2-4

Mon 6-8

Lemma: For any two strings w and z,

$$|w \bullet z| = |w| + |z|$$

Think big!

$(w = a \cdot x) \qquad (z = b \cdot y)$ induct on w!

$$|w \bullet z| = |(a \cdot x) \bullet z| = 1 + |x \bullet z|$$

Proof: Let w and z be two arbitrary strings.

Assume $|x \bullet z| = |x| + |z|$ for every string x such that $|x| < |w|$.

There are two cases to consider.

Suppose $w = \epsilon$.

$$
\begin{aligned}
|w \bullet z| &= |\epsilon \bullet z| && w = \epsilon \\
&= |z| && \text{def. } \bullet \\
&= 0 + |z| && \text{arithmetic} \\
&= |\epsilon| + |z| && \text{def. } |\cdot| \\
&= |w| + |z| && w = \epsilon
\end{aligned}
$$

Suppose $w = a \cdot x$ for some symbol a & string x.

$$|w \bullet z| = |(a \cdot x) \bullet z| \qquad w = a \cdot x$$
$$= |a \cdot (x \bullet z)| \qquad \text{def. } \bullet$$
$$= 1 + |x \bullet z| \qquad \text{def. } |\cdot|$$
$$= 1 + |x| + |z| \qquad \text{IH}$$
$$= |a \cdot x| + |z| \qquad \text{def. } |\cdot|$$
$$= |w| + |z| \qquad w = a \cdot x$$

Lemma: $(w \bullet y) \bullet z = w \bullet (y \bullet z)$

A formal language is a set of strings from some common alphabet.

Σ^*: all strings with alphabet Σ

Language $L \subseteq \Sigma^*$.

Ex: $\emptyset$ empty set (not a string)

$\{\varepsilon\}$ not empty, but has empty string

$\{0,1\}^*$ binary strings

{BABA, KIKI, FOFO, JIJI}

from $\{0,1\}^*$ with an odd # 1's

The set of all valid Python

programs,

If L_1 and L_2 are languages, so is

$L_1 \cup L_2 \quad L_1 \setminus L_2$

$L_1 \cap L_2 \quad L_1 \oplus L_2$ ← symmetric diff.

$\overline{L_1} := \Sigma^* \setminus L_1$ (compliment)

The <u>concatenation</u> of languages L_1 and L_2 is

$$L_1 \bullet L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\}$$

Ex: $\{SUPER, SPIDER, BAT\} \bullet \{MAN, WOMAN\}$

has size 6.

Ex: $\emptyset \bullet L = \emptyset = L \bullet \emptyset$

Ex: $\{\varepsilon\} \bullet L = L = L \bullet \{\varepsilon\}$

Kleene closure (kleene star) of language L is

"clay-knee"

$$L^* = \{\varepsilon\} \cup L \cup L \bullet L \cup L \bullet L \bullet L \cup \ldots$$

i.e. smallest solution to

$$L^* = \{\varepsilon\} \cup L \bullet L^*$$

i.e. $w \in L^*$ iff $w = \varepsilon$ or $w = xy$ with $x \in L$ and $y \in L^*$.

Ex: $\{0, 11\}^* = \{\varepsilon, 0, 11, 00, 011, 1111, 000, \ldots$

Ex: $\emptyset^* = \{\varepsilon\}$

$\{\varepsilon\}^* = \{\varepsilon\}$

A language L is <u>regular</u> if one of

$L = \emptyset$

L contains one string (could be ε)

L is the union of two regular languages

L is the concatenation of two reg. languages.

L is the Kleene closure of a reg. language.

Not regular: $\{0^n 1^n \mid n \geq 0\}$

Regular expressions:	Language
$\emptyset$	: $\emptyset$
w	: $\{w\}$
$A + B$	: A's language $\cup$ B's lang.
AB	: A's $\bullet$ B's
A^*	: (A's lang.)*

simpler regular expressions

$* > \text{concatenation} > +$

can add parantheses

Ex: $0+10^* = \{0\} \cup (\{1\} \circ \{0\}^*)$

$= \{0, 1, 10, 100, 1000, \ldots\}$

R: a regular expression

It <u>represents</u> language $L(R)$

w <u>matches</u> R is $w \in L(R)$

Ex: Even length binary strings

$$((0+1)(0+1))^*$$

Ex: Binary strings of alternating 0's & 1's

$$(\epsilon+1)(01)^*(\epsilon+0)$$