GPS 1 : Mon Aug 31
HW 1 : Tue Sep 1

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

Lemma: $(w \bullet y) \bullet z = w \bullet (y \bullet z)$ for all string w, y, z.

Proof: Let $w, y, \& z$ be arbitrary strings.
Assume, for every string x s.t. $|x| < |w|$,

such that

$$(x \bullet y) \bullet z = x \bullet (y \bullet z)$$

There are two cases,
Suppose $w = \varepsilon$. Then,

$$
\begin{aligned}
(w \bullet y) \bullet z &= (\varepsilon \bullet y) \bullet z && (w = \varepsilon) \\
&= y \bullet z && (\text{def. } \bullet) \\
&= \varepsilon \bullet (y \bullet z) && (\text{def. } \bullet) \\
&= w \bullet (y \bullet z) && (w = \varepsilon)
\end{aligned}
$$

Suppose $w = ax$ for some symbol a, string x

$$
\begin{aligned}
(w \bullet y) \bullet z &= ((ax) \bullet y) \bullet z && (w = ax) \\
&= (a \cdot (x \bullet y)) \bullet z && (\text{def. } \bullet) \\
&= a \cdot ((x \bullet y) \bullet z) && (\text{def. } \bullet) \\
&= a \cdot (x \bullet (y \bullet z)) && (IH)
\end{aligned}
$$

$$= (ax) \bullet (y \bullet z) \qquad (\text{def. } \bullet)$$

$$= w \bullet (y \bullet z) \qquad (w = ax)$$

Both cases have $(w \bullet y) \bullet z = w \bullet (y \bullet z)$

$\square$

formal language (or **language**) is a set of strings over a common alphabet Σ

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

- $\{\varepsilon\}$ set of only the empty string
- Σ^*: all strings over Σ
- $\{0,1\}^*$: all binary strings
- set of all strings in $\{0,1\}^*$ with an odd number of 1s.
- set of all syntactically valid Python programs (over ASCII)

If A & B are languages over Σ, so are

$A \cup B := \{w \mid w \in A \text{ or } w \in B\}$

$A \cap B :=$ " and "

$A \setminus B :=$ in A but not B

$A \oplus B :=$ w in exactly one of A or B

$\bar{A} := \Sigma^* \setminus A$ = complement of A.

Concatenation of A & B is

$A \bullet B := \{x \bullet y \mid x \in A \text{ \& } y \in B\}$

$\{SUPER, SPIDER\} \bullet \{MAN, WOMAN\}$
$= \{SUPERMAN, SUPERWOMAN,$

SPIDERMAN, SPIDERWOMAN}

language

$\emptyset \bullet L = L \bullet \emptyset = \emptyset$

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

Kleene closure (Kleene star) of L:

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

· or - the smallest solution to $L^* = \{\varepsilon\} \cup L \bullet L^*$.

· or - $w \in L^*$ if and only if

$w = \varepsilon$ or

$w = x \bullet y$ where $x \in L$ & $y \in L^*$

Ex: $\{0,1\}^* =$ all binary strings

Ex: $\{0,11\}^* = \{\epsilon, 0, 11, 011, 00, 110, 1111, \ldots\}$

$\emptyset^* = \{\epsilon\} = \{\epsilon\}^*$

$\{0\}^* = \{\epsilon, 0, 00, 000, 0000, 00000, \ldots\}$

A language L is <u>regular</u> if and only if L satisfies one of the following:

- $L = \emptyset$
- L contains exactly one string (may be ϵ)
- L is the union of two regular languages
- L is the concatenation "
- L is the kleene closure of a regular language

Ex: $\{0^n 1^n \mid n \geq 0\}$ is <u>not</u> regular.

Regular expression: a notation to describe regular languages

$\emptyset$: empty language

w: language $\{w\}$

$R_A + R_B$: $A \cup B$

regular expressions for A & B

$R_A R_B$: $A \circ B$

R_A^* : A^*

Add parantheses as necessary.

$* \gg$ concat $\gg +$.

Ex: $0+10^*$ <u>represents</u> $\{0\} \cup (\{1\} \cdot \{0\}^*)$

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

each of these <u>matches</u> the regular expression $0+10^*$

$10^* \neq (10)^*$

these are <u>not</u> regexes

$\swarrow$ fixed integer

you <u>may</u> write ex. $(10)^3$ to mean 101010.

Ex: Even length binary strings:

members of $\{0,1\}$: $(0+1)$

pairs: $(0+1)(0+1)$

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

Ex: Binary strings alternating 0s & 1s.

i.e. no substrings 00 or 11

$$\cancel{1(01)^*0} \quad \text{oops}$$

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