Admin: DRINK WATER!!

- Changing sections
- Homework — Gradescope, 1 sub per group per # problem
- Homework parties
  - Thu 5–8pm Siebel 0216 ← ≤93
  - Sat 2–5 pm Siebel 1404 this week, DCL 1320 after that ← ≤200

Lemma: For all strings w, y, z : $(w\bullet y)\bullet z = w\bullet(y\bullet z)$

Proof: Let w, y, z be arbitrary strings

Assume for all strings x shorter than w. that $(x\bullet y)\bullet z = x\bullet(y\bullet z)$

Case: $w=\varepsilon$

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

Case: $w = ax$ for some $a\in\Sigma$ and string x

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

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

Therefore $(w\bullet y)\bullet z = w\bullet(y\bullet z)$

Proof: Let w be an arbitrary string.
Assume, for every string x such that $|x| < |w|$, that x is perfectly cromulent.
There are two cases to consider.

- Suppose $w = \varepsilon$.

  Therefore, w is perfectly cromulent.

- Suppose $w = ax$ for some symbol a and string x.
  The induction hypothesis implies that x is perfectly cromulent.

  Therefore, w is perfectly cromulent.

In both cases, we conclude that w is perfectly cromulent. □

LANGUAGE = set of strings　　　Alphabet Σ

All binary strings　$\{0,1\}^*$　　$\Sigma^* \leftarrow$ all strings over Σ

$\emptyset$　empty set　　boeuf　bõse　øl

$\{\varepsilon\}$

$\cancel{\varepsilon}$

$\{w \in \{0,1\}^* \mid \#1\text{s in } w = \#0\text{s in } w\}$

$\{001101\}$

{JAKE, FINN, FIONNA, CAKE}

All Python programs

All Python programs that ∞ loop

$L = A \cup B$

$L = A \cap B$

$L = A \setminus B$

$L = A \oplus B$

$\overline{L} = \Sigma^* \setminus L$

$L = A \bullet B = \{x \bullet y \mid x \in A \text{ and } y \in B\}$

{FIRST, SECOND, THIRD} • {PLACE, BASE}

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

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

Kleene star/closure

$L^* =$ concatenations of any # strings in L

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

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

$$\boxed{w \in L^* \iff w = \varepsilon \text{ or } w = x \bullet y \text{ for some } x \in L \text{ and } y \in L^*}$$

$\{01\}^* = \{\varepsilon, 01, 0101, 010101, \ldots\}$

Is L^* always infinite?

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

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

Kleene's regular languages

L is regular $\iff$

- $L = \emptyset$
- $L = \{w\}$ for some $w \in \Sigma^*$
- branching: $L = A \cup B$ reg lang A, B
- sequencing: $L = A \bullet B$ reg langs A, B
- repetition: $L = A^*$ reg lang A

while

if

else

regular expression $=$ $\emptyset$, w, $A + B$, AB, A^*

$0 + 10^* = \{0\} \cup (\{1\} \bullet \{0\}^*)$

$= \{0, 1, 10, 100, 1000, 10000, \ldots\}$

Alternating 0's and 1's = never 00 or 11

Good: $\boxed{\varepsilon}$, 0, 1, 10, $\boxed{01}$, $\boxed{0101}$0, 1$\boxed{01010101}$0, ...

Bad: 00, 11, 00000, 11001, 1101011, ...

$$\underline{(1+\varepsilon)}(01)^*\underline{(0+\varepsilon)}$$

Lemma 2.1. *The following identities hold for all languages A, B, and C:*

(a) $A \cup B = B \cup A$.

(b) $(A \cup B) \cup C = A \cup (B \cup C)$.

(c) $\varnothing \bullet A = A \bullet \varnothing = \varnothing$.

(d) $\{\varepsilon\} \bullet A = A \bullet \{\varepsilon\} = A$.

(e) $(A \bullet B) \bullet C = A \bullet (B \bullet C)$.

(f) $A \bullet (B \cup C) = (A \bullet B) \cup (A \bullet C)$.

(g) $(A \cup B) \bullet C = (A \bullet C) \cup (B \bullet C)$.

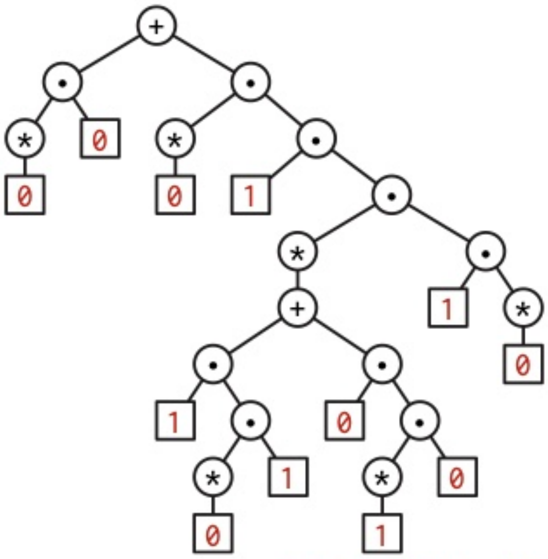
Lemma 2.2. *The following identities hold for every language L:*

(a) $L^* = \{\varepsilon\} \cup L^+ = L^* \bullet L^* = (L \cup \{\varepsilon\})^* = (L \setminus \{\varepsilon\})^* = \{\varepsilon\} \cup L \cup (L \bullet L^+)$.

(b) $L^+ = L \bullet L^* = L^* \bullet L = L^+ \bullet L^* = L^* \bullet L^+ = L \cup (L \bullet L^+) = L \cup (L^+ \bullet L^+)$.

(c) $L^+ = L^*$ *if and only if* $\varepsilon \in L$.

Lemma 2.3 (Arden's Rule). *For any languages A, B, and L such that* $L = A \bullet L \cup B$, *we have* $A^* \bullet B \subseteq L$. *Moreover, if A does not contain the empty string, then* $L = A \bullet L \cup B$ *if and only if* $L = A^* \bullet B$.



A regular expression tree for $\mathtt{0}^*\mathtt{0} + \mathtt{0}^*\mathtt{1}(\mathtt{10}^*\mathtt{1} + \mathtt{01}^*\mathtt{0})^*\mathtt{10}^*$

Proof: Let R be an arbitrary regular expression.
Assume that **every regular expression smaller than R** is perfectly cromulent.
There are five cases to consider.

- Suppose $R = \varnothing$.

  Therefore, R is perfectly cromulent.
- Suppose R is a single string.

  Therefore, R is perfectly cromulent.
- Suppose $R = S + T$ for some regular expressions S and T.
  The induction hypothesis implies that S and T are perfectly cromulent.

  Therefore, R is perfectly cromulent.
- Suppose $R = S \bullet T$ for some regular expressions S and T.
  The induction hypothesis implies that S and T are perfectly cromulent.

  Therefore, R is perfectly cromulent.
- Suppose $R = S^*$ for some regular expression. S.
  The induction hypothesis implies that S is perfectly cromulent.

  Therefore, R is perfectly cromulent.

In all cases, we conclude that w is perfectly cromulent. □