THM. EVERY STRING IS PERFECTLY CROMULENT.

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. □

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 $(x \bullet y) \bullet z = x \bullet (y \bullet z)$ for all x s.t. $|x| < |w|$, all y, all z.

Case 1: $w = \varepsilon$

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

Case 2: $w = a \cdot x$ for some symbol a, string x.

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

$$= a\cdot(x\bullet(y\cdot z)) \quad \text{[by IH]}$$

$$= a\cdot x \bullet (y\cdot z) \quad \text{[def } \bullet\text{]}$$

$$= w\bullet(y\cdot z) \quad [w = a\cdot x]$$

In both cases, $(w\bullet y)\bullet z = w\bullet(y\cdot z)$

LANGUAGES

Language: set of strings over an alphabet

eg, $\Sigma = \{0,1\}$

Examples of languages:

$\phi \rightarrow$ EMPTY SET (no strings)

$\{\varepsilon\} \rightarrow$ set containing the empty string

~~ε~~ NOT a language

Σ^* : All strings over Σ.

S^* : All strings formed by concatenating symbols from set S.

Kleene

Σ^5 : All strings of length 5 formed by concatenating symbols from Σ.

$\Sigma = \{A, B, C \dots Z\}$

$\{XYZ\}$ is a language

$L = L_1 \cup L_2$ "

$L = L_1 \cap L_2$ "

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

$L =$ All python programs

$L = A \cdot B = \{x \cdot y \mid x \in A, y \in B\}$

$\{OVER, UNDER\} \cdot \{EAT, PAY\}$

$\{0\}^* \cdot \{1\}^*$

$\phi \cdot L = \phi$ $\{\varepsilon\} \cdot L = L$

Also a language $L^* = \{\varepsilon\} \cup L \cup L \cdot L \cup \dots\dots$

$w \in L^* \iff w = \varepsilon$ OR $w = x \cdot y$
where $x \in L, y \in L^*$

Is L^* always infinite?

What is L^* when $L = \phi$? $L^* = \{\varepsilon\} \cup \phi \cup \phi \cdot \phi$
$= \{\varepsilon\}$

What about $L = \{\varepsilon\}$? $L^* = \{\varepsilon\} \cup \{\varepsilon\} \cdot \{\varepsilon\} \dots = \{\varepsilon\}$

$L = \{1\} \Rightarrow L^*$ is infinite.

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

$L^+ = L \cup L \cdot L \cup L \cdot L \cdot L$ ....

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

(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* $L = A^* \bullet B$.

REGULAR LANGUAGES

L is regular means

either * $L = \phi$

or * $L = \{w\}$ for some string w

if-then-else or * $L = A \cup B$ for regular A, B

sequence of lines of code or * $L = A \bullet B$ for regular A, B

loop or * $L = A^*$ for regular A

REGULAR EXPRESSIONS

$0 + 10^*$

$= \boxed{\{0\} \cup \{1\} \cdot \{0\}^*}$

$0 \quad \overset{or}{=} \cup \quad 1 \cdot \varepsilon$

$\cup \; 1 \cdot 0$

$\cup \; 1 \cdot 00$

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

~~101~~

Eg: The language of alternating 0s and 1s.

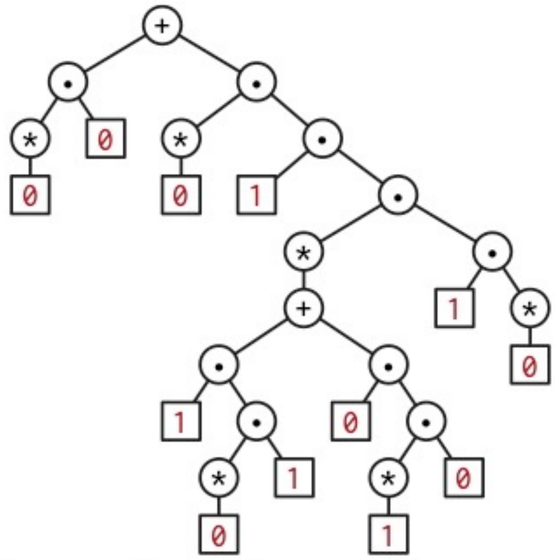
strings in language

ε, 0101, 0, 1, 10, 101, ...

strings not in language

11, 0010, 01101

Regular Expression: $(\varepsilon + 1)\{01\}^*(\varepsilon + 0)$



A regular expression tree for $0^*0 + 0^*1(10^*1 + 01^*0)^*10^*$

THM: Every regular expression is perfectly cromulent.

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. □