**Theorem:** 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 \( w = \epsilon \).
  
  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. \( \square \)

**Lemma:** For all strings \( w, y, z : \) \( (w \cdot y) \cdot z = w \cdot (y \cdot z) \)

**Proof:** Let \( w, y, z \) be arbitrary strings.

IH: Assume \( (x \cdot y) \cdot z = x \cdot (y \cdot z) \) for all strings \( x \) shorter than \( w \).

There are two cases:

- \( w = \epsilon \) \( \Rightarrow (w \cdot y) \cdot z = (\epsilon \cdot y) \cdot z \)
  
  \( = y \cdot z \) \( \quad \text{[def \( \cdot \)]} \)
  
  \( = \epsilon \cdot (y \cdot z) \) \( \quad \text{[def \( \cdot \)]} \)
  
  \( = w \cdot (y \cdot z) \) \( \quad \text{[w = \( \epsilon \)]} \)

- \( w = ax \) for some symbol \( a \) and string \( x \)
  
  \( (w \cdot y) \cdot z = ((a \cdot x) \cdot y) \cdot z \) \( \quad \text{[w = ax]} \)
  
  \( = (a \cdot (x \cdot y)) \cdot z \) \( \quad \text{[def \( \cdot \)]} \)
  
  \( = a \cdot (x \cdot (y \cdot z)) \) \( \quad \text{[def \( \cdot \)]} \)
  
  \( = (a \cdot (x \cdot (y \cdot z))) \) \( \quad \text{IH} \)
  
  \( = w \cdot (y \cdot z) \) \( \quad [w = ax] \)

Therefore, \( (w \cdot y) \cdot z = w \cdot (y \cdot z) \)
\textbf{LANGUAGES} = sets of strings over \( \Sigma \)

\( \emptyset \)

\( \{ \varepsilon \} \cup \Sigma^* = \) all strings over \( \Sigma \)

\( \varepsilon \in \{ 0, 1 \}^* \) \( | w \) has even \# of 1s \( \varepsilon \in \{ \varepsilon, 00, 101, \ldots \} \)

\( \{ \text{BMO} \} \)

\( \{ \text{FINN, JAKE, ICEKING} \} \)

\( \varepsilon \in \{ 0, 1 \}^* \) \( w \) is binary for prime \#\( 5 \)

\( L = A \cup B \) \quad \text{All Python programs}

\( L = A \cap B \) \quad \text{All Python programs that \( \infty \) loop}

\( L = \overline{A} = \Sigma^* \setminus A \)

\( L = A \cdot B = \{ x \cdot y \mid x \in A \text{ and } y \in B \} \)

\( \{ \text{FIRST, SECOND, THIRD} \} \cdot \{ \text{BASE, PLACE} \} \)

\( \{ 0 \}^* \cdot \{ 1 \}^* \)

\( \emptyset \cdot L = \emptyset \quad \varepsilon \cdot L = L \)

\( L^* = \text{Kleene star} = \{ \varepsilon \} \cup \varepsilon L \cup \varepsilon L \cup \ldots \)

\( w \in L^* \iff w = \varepsilon \text{ or } w = x \gamma \text{ for some } x \in L \quad \gamma \in L^* \)

Is \( L^* \) always infinite?

\( \emptyset^* = \{ \varepsilon \} \cup \emptyset \cup \emptyset \cup \ldots = \{ \varepsilon \} \)

\( \{ 0 \}^* = \{ \varepsilon \} \cup \{ \varepsilon \} \cdot \{ 0 \} \cup \ldots = \{ \varepsilon \} \)
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) $\emptyset \cdot A = A \cdot \emptyset = \emptyset$.
(d) $\{\varepsilon\} \cdot A = A \cdot \{\varepsilon\} = A$.
(e) $(A \cdot B) \cdot C = A \cdot (B \cdot C)$.
(f) $A \cdot (B \cup C) = (A \cdot B) \cup (A \cdot C)$.
(g) $(A \cup B) \cdot C = (A \cdot C) \cup (B \cdot C)$.

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

(a) $L^* = \{\varepsilon\} \cup L^+ = L^* \cdot L^* = (L \cup \{\varepsilon\})^* = (L \setminus \{\varepsilon\})^* = \{\varepsilon\} \cup L \cup (L^* \cdot L^*)$.
(b) $L^+ = L \cdot L^* = L^* \cdot L = L^+ \cdot L^* = L^* \cdot L^+ = L \cup (L^* \cdot L^*)$.
(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 $L = A \cdot L \cup B$, we have $A^* \cdot B \subseteq L$. Moreover, if $A$ does not contain the empty string, then $L = A \cdot L \cup B$ if and only if $L = A^* \cdot B$.

Regular languages

$L$ is regular means either

- if $\varepsilon w3$
- else $A \cup B$

Sequencing

if $A \cdot B$

while

else

Regular expressions

$0 + 10^*$

$= \varepsilon 03 \cup (\varepsilon 15 \cdot (\varepsilon 05)^*)$
Alternating 0s and 1s

Good: 0, 1, 0, 101, 010101, 01010, ...

Bad: 11, 0100, 01101, ...

\[
\begin{align*}
\varepsilon + 0 \cdot (10)^* (1+\varepsilon) + 1 \cdot (01)^* (0+\varepsilon) &= (0 + \varepsilon)(10)^* (1 + \varepsilon)
\end{align*}
\]
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 $R = \emptyset$.
  
  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 $R = S \cdot T$ 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.