Lemma: For all strings $w,y,z : (wyz)z = w \cdot (yoz)$

Proof: Let $w,y,z$ be arbitrary strings. Assume for all strings $x$ shorter than $w$ that $(xoy)z = x \cdot (yoz)$. 

Case: $w=\varepsilon$

$(wyz)z = (\varepsilon y)z$

$= yz$

$= \varepsilon \cdot (yoz)$

$= w \cdot (yoz)$

$= \varepsilon$

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

$(wyz)z = (\alpha x y)z$

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

$= (\alpha \cdot \varepsilon)z$

$= \alpha \cdot (\varepsilon z)$

$= \alpha \cdot ((x y)z)$

$= \alpha \cdot (x(yoz))$

$= \alpha \cdot x \cdot (yoz)$

$= w \cdot (yoz)$

Therefore $(wyz)z = w \cdot (yoz)$
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.

\[
\begin{align*}
\text{LANGUAGE} &= \text{set of strings} \\
\text{All binary strings} &= \Sigma^* \\
\emptyset &= \text{empty set} \\
\Sigma^* &= \text{all strings over } \Sigma \\
\exists \geq 3 \\
\forall w \in \Sigma^* \mid \#1s \geq w = \#0s \text{ in } w \\
001101 & \\
\exists \text{JAKE, FINN, FIONNA, CAKES} \\
\text{All Python programs} \\
\text{All Python programs that do loop} \\
\end{align*}
\]
Kleene star/closure

\[ L^* = \text{concatenations of any strings in } L \]
\[ = \varepsilon L \cup L \cdot L \cup L \cdot L \cdot L \cup \ldots \]
\[ = \varepsilon \cup L \cdot L^* \]

\[ w \in L^* \iff w = \varepsilon \text{ or } w = x \cdot y \text{ for some } x \in L \text{ and } y \in L^* \]

\[ \varepsilon 01 \varepsilon^* = \varepsilon, 01, 0101, 010101, \ldots \]

Is \( L^* \) always infinite? \( \emptyset^* = \varepsilon \varepsilon^* \)
\( \varepsilon 01 \varepsilon^* = \varepsilon 01 \varepsilon^* \)

Kleene's regular languages

\( L \) is regular \( \iff \)
\[ \begin{cases} L = \emptyset & \text{For some } w \in \varepsilon^* \\ L = \varepsilon w^3 & \text{branching} \\ L = A \cup B & \text{reg langs } A, B \\ L = A \cdot B & \text{reg langs } A, B \\ L = A^* & \text{reg lang } A \\ \end{cases} \]

While

```
if
else
```

regular expression = \( \emptyset \)
\( A + B \)
\( AB \)
\( A^* \)

\[ \emptyset + 10^* = \varepsilon 01 \varepsilon^* \cup (\varepsilon 10 \varepsilon^* \cdot \varepsilon 01 \varepsilon^*) \]
\[ = \varepsilon 0, 1, 10, 100, 1000, 10000, \ldots \]
Alternating O's and I's = never 00 or 11

Good: \[ \varepsilon, 0, 1, 10, 01, 01010, 10101010, \ldots \]

Bad: \[ 00, 11, 00000, 11001, 1101011, \ldots \]

\[
\left( 1 + \varepsilon \right) (01)^{*} (0 + \varepsilon)
\]

**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) \( \emptyset \cdot A = A \cdot \emptyset = \emptyset \).

(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) \( L^+ = L \cdot L^* = L^* \cdot L = L^+ \cdot L^* = L^* \cdot L^+ = L \cup (L \cdot L^*) = L \cup (L^+ \cdot L^+) \).

(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 \( L = A^* \cdot B \).
A regular expression tree for $0^*0 + 0^*1(10^*1 + 01^*0)^*10^*$

**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 $R = \emptyset$.

  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.