Notes for October 2nd

Example 1)
Prove that \( C = \{ w : w \text{ has an equal number of 0's and 1's} \} \) is not regular

Proof: Assume that \( C \) is regular and let \( p \) be the pumping length. Choose \( s \) to be \( 0^p1^p \) so \( s \in C \) and \( |s| \geq p \Rightarrow s \) can be split into \( xyz \) such that \( xy^i z, i \geq 0 \in C \).

- If \(|xy| \leq p\), then \( y \) must contain only zeros so \( xyyz \notin C \)

Therefore since \( s \) is a string in \( C \) with \(|s| \geq p\), and we reached a contradiction by assuming \( C \) was regular it follows that \( C \) is not regular.

Example 2)
Prove that \( F = \{ ww : w \in \{0, 1\}^* \} \) is not regular

Proof: Assume that \( F \) is regular and let \( p \) be the pumping length. Choose \( s \) to be \( 0^p1^p0^p1 \) so \( s \in F \) and \(|s| \geq p \Rightarrow s \) can be split into \( xyz \) such that \( xy^i z, i \geq 0 \subseteq F \).

- If \(|xy| \leq p\), \( y \) must only have zeros but \( xyyz \notin F \)

Therefore since \( s \in F \) with \(|s| \geq p\), and we reached a contradiction by assuming \( F \) was regular it follows that \( F \) is not regular.

Context-Free Grammars

- Finite Automata are equivalent and describe regular languages
- Some languages can’t be describe this way
- Regular-Union, concatenation, star
- CF-recursion

CFG are more powerful so \( R \) is a subset of CFG

All programming languages are designed with a CFG

Parser is constructed from the grammar

Ex) \( G = \{ 0^n\#1^n : n \geq 1 \} \)

- \( A \rightarrow 0A1 \)
- \( A \rightarrow B \)
B -> #
Derivation: A -> 0A1 -> 00A11 -> 00B11 -> 00#11
Parse Tree:

- A rule in the grammar consists of a symbol and an arrow pointing to a string containing symbols and terminals

October 4th
We did not take notes on October 4th
Notes from 10/2

**Thm.** $C = \{ w : w \text{ has an equal # of 0's and 1's} \}$ is not regular

**Proof:** Assume that $C$ is regular and let $p$ be the pumping length. Choose $s$ to be $0^p1^p$ so $S \in |S| \geq p$

$\rightarrow$ $S$ can be split into $xyz$ s.t. $xy^iz \in C \forall i \geq 0$

- If $|xy| \leq p$, then $y$ must contain only zeroes. So $xyyz \notin C$

$\therefore$ Since $s$ is a string in $C$ with $|S| \geq p$ and we reached a contraction by assuming $C$ was regular

$\rightarrow$ $C$ is not regular

**Thm.** $F = \{ ww : w \in \{0,1\} \}$ is not regular

**Proof.** Assume $F$ is regular. Let $p$ be the pumping length

- Let $S = 0^p1^p0^p1^p \notin F$ and $|S| \geq p$

Thus $S$ can be split into 3 pieces

- $xyz$ s.t. $xy^iz \in F \forall i \geq 0$
- If $|xy| \leq p$, $y$ must only have 0's
- But $xyyz \notin F$

$\rightarrow$ Thus, since we assumed $F$ was regular and $S \in F$ and $|S| \geq p$, but pumping lemma led to a contradiction, thus $F$ is not a contradiction.

**Thm.** $E = \{ 0^i1^j : i > j \}$ is not regular

**Proof:** Assume that $E$ is regular and let $p$ be the pumping length. Choose $S$ to be $0^p1^p1^p$ so $S \in E$ and $|S| \geq p$. Thus, we can split $S$ into $xyz$ s.t. $xy^iz \in E \forall i \geq 0$.

- If $|xy| \leq p$ so $y$ consists of only 0's
- $|y| > 0$ so $y$ is at least one character

$\rightarrow$ Thus $xy^0z = xz, 0^p1^p \notin E$ pumping down

**Context Free Grammars**

- FA and regex are equivalent and describe regular language
- Some language can't be described this way
- Reg – union, concentration, star
- CF – recursion
- CFG are more powerful so R<CFG
- All P.L. are designed w/ a CFG w/ a grammar
- Parser is constructed from the grammar
- A rule in the grammar consists of a symbol and an arrow pointing to a string containing symbols and terminals

Ex: \[ G = \{ 0^n#^n : n \geq 1 \} \]
A -> O
A -> B
B -> #

Derivation
A -> 0A1 -> 000A1111 -> 000#111