First a note on the "pumping lemma" and logic: Version 1b (2x2) Theorem 1b

**If A is a regular language, then**

- **There exists** an integer \( p > 0 \) such that 
  - **For all** \( s \in A \) with \( |s| \geq p \) 
    - **There exists** a breakdown \( S = x \cdot y \cdot z \) such that 
      - \( |x| \cdot |y| \leq p \), \( y \neq \varepsilon \), and 
      - **For all** \( i \geq 0 \), 
        - \( xy^iz \in A \).

**Proof Script:**

Let any \( p > 0 \) be given. 

- **A1:** \( A_1 = \{ \text{W} \cdot W^R : W \in \{a,b\}^* \} = \{ \text{palindrome} \} \)
- **A2:** \( A_2 = \{ \text{W} \cdot W : \text{W} \in \{a,b\}^* \} = \{ \text{square} \} \)

**Case 1:** \( A_1 \)

- Let \( x = a^m \cdot y = a^n \), \( m, n \geq 0 \)
- Consider any breakdown \( S = x \cdot y \cdot z \) with \( |x| \cdot |y| \leq p \), \( y \neq \varepsilon \)
  - Take \( i = 0 \). 

**Case 2:** \( A_2 \)

- Let \( x = a^m b, y = a^n b \), \( m, n \geq 0 \)
- Consider any breakdown \( S = x \cdot y \cdot z \) with \( |x| \cdot |y| \leq p \), \( y \neq \varepsilon \)
  - Take \( i = 0 \). 

Hence \( A \) is not regular, by the Pumping Lemma.
Context-Free Grammars:

\[ A_1 = \{ W \cdot WR = WE \{ a, b \}^* \} \]

\[ A_1 = \{ \varepsilon, aa, bb, aaaa, abba, baba, bbbb, aaaa, aabbaaa, \ldots \} \]

A CFG \( G \) s.t. \( L(G) = A_1 \):

\( S \) means "I stand for some EVENPAL x."

\( S \rightarrow aSa \mid bSb \mid \varepsilon \)

Derivation:

\[ S \rightarrow aSa \rightarrow aaaSaa \rightarrow aabSbbaa \rightarrow aabbaa \]

\[ S \rightarrow aSa \rightarrow aaaSaa \rightarrow aabSbbaa \rightarrow aabbaa \]

Inductive Defn:

Basis: \( \varepsilon \) is an "EVENPAL"

Induction: If \( x \) is an EVENPAL, then

- \( axa \) is also an EVENPAL.
- If \( x \) is an EVENPAL, then \( bxb \) is also an EVENPAL.

Catchall Clause: The only EVENPALS are the ones that arise via these three rules.

\[ A_2 = \{ WW = WE \{ a, b \}^* \} \]

Can we design a CFG \( G \) such that

\( G \) generates exactly those strings in \( A_2 \), i.e. \( L(G) = A_2 \)?

Answer: No! \( \varepsilon, a, a, a, abab, abbab, ababbb \).

CFG \( \equiv \) BNF as taught in CSE 305.
Formal Definition: A context-free grammar (CFG) is a 4-tuple \( G = (V, \Sigma, R, S) \) where \( \Sigma \) is the terminal alphabet, \( V \) is a finite set of variables, \( S \in V \) is the start variable, and \( R \) is a finite set of rules of the form \( A \to X \) where \( A \in V \) and \( X \in (V \cup \Sigma)^* \).

Before we have \( \Sigma = \{a, b, c\} \),

\[ \Sigma = \{a, b, c\}, \quad R = \{S \to \varepsilon, S \to aSa, S \to bSb\}. \]

Given two strings \( X, Y \in (V \cup \Sigma)^* \), we write \( X \Rightarrow Y \) if \( X \) can be broken down as \( X \Rightarrow UVW \) such that for some rule \( A \to Z \in R, \ Y = UZW \).

We write \( S \Rightarrow^* X \) if there are \( Y_1, \ldots, Y_k \) s.t. \( X = VAW \)

\[ S \Rightarrow S \Rightarrow Y_1 \Rightarrow Y_2 \Rightarrow \cdots \Rightarrow Y_k \Rightarrow X, \text{ and } \text{shall } L(G) = \{x \in \Sigma^* : S \Rightarrow^* x \}. \]

Variables can also be written as XML-style tags.

\[ S \to \langle Noun Phrase \rangle \langle Verb Phrase \rangle \mid \ldots \]
\[ \langle Noun Phrase \rangle \to \langle Complex Noun \rangle \mid \langle Complex Noun \rangle \langle Prep Phrase \rangle \mid \cdots \]
\[ \langle Complex Noun \rangle \to \langle Article \rangle \langle Simple Noun \rangle \mid \cdots \]
\[ \langle Article \rangle \to \text{ the } \mid \text{ a/an} \]
\[ \langle Simple Noun \rangle \to \text{ cat } | \text{ dog } | \text{ hat} \]
\[ \langle Prep Phrase \rangle \to \langle Preposition \rangle \langle Complex Noun \rangle \]
\[ \langle Noun Phrase \rangle \Rightarrow \text{ the cat in the hat}. \]

[End of Week 6]