Session 14

Context-free Grammars

There are languages that are non-regular

- $Pal = \{w \mid w = w^R\} \subseteq \{0, 1\}^*$
- Pal is not regular:
  - The pumping Lemma:
    - Let n be the associated constant
    - Let $w = 0^n10^n$: $|w| = 2n + 1 > n$
    - If Pal is regular $w = xyz$, such that $|xy| \leq n$ and $|y| > 0$; y is a sequence of 0's at the end of the first group: $x = 0^i$ and $y = 0^j$, such that $i \geq 0$, $j > 0$, $i + j = n$ so $|xy| = |0^i0^j| = |0^n| \leq n$ and $|y| = j > 0$.
    - Let $m = 0$:
      - $xy^mz = xz = 0^i10^n \notin Pal$ as $i < n$
- Pal cannot be represented through a RE or a FA

Recursive definition of languages

- Recursive definition of a language:
  - Define composite strings of in the language as a function of more simple strings in the language
- Recursive definition of Pal
  - Basis: Λ, 0 and 1 $\in$ Pal
  - Induction: if $w \in Pal$ then $1w1$ and $0w0$

Recursive definition of languages

- A CFG is a notation to express this kind of recursive definitions
  - Variables represent classes of strings (i.e. grammatical categories and languages)
  - Constants represent the lexical symbols in Σ
  - Production rules of the form

    $\alpha \rightarrow \beta$

    α can be rewritten as β in any context

Recursive definition of languages

- Recursive definition of Pal
  - Basis: Λ, 0 and 1 $\in$ Pal
  - Induction: if $w \in Pal$ then $1w1$ and $0w0$

- The grammar of Pal:
  1. $P \rightarrow \Lambda$
  2. $P \rightarrow 0$
  3. $P \rightarrow 1$
  4. $P \rightarrow 0P0$
  5. $P \rightarrow 1P1$

$0110 \in Pal$

1.	$P \Rightarrow 0P0$	by 4
2.	$\Rightarrow 01P10$	by 5
3.	$\Rightarrow 01\Lambda10$	by 1
	$= 0110$	

Recursive definition of a language: Example 2

- Recursive definition of L_{exp} (Non-regular)
  - Basis: a
  - Induction: if $w \in L$ then $w + w \mid w * w \mid (w)$

- The CFG:
  1. $E \rightarrow a$
  2. $E \rightarrow E + E$
  3. $E \rightarrow E * E$
  4. $E \rightarrow (E)$

$a + (a * a) \in L$:

1.	$E \Rightarrow E + E$	by 2
2.	$\Rightarrow a + E$	by 1
3.	$\Rightarrow a + (E)$	by 4
4.	$\Rightarrow a + (E * E)$	by 3
5.	$\Rightarrow a + (a * E)$	by 1
6.	$\Rightarrow a + (a * a)$	by 1

Formal definition of CFG

- A context-free grammar (CFG) is a 4-tuple

  $G = (V, \Sigma, S, P)$,

  where:
  - V is a set of variables (non-terminal symbols, syntactic categories, types of strings)
  - Σ is the alphabet (terminal or lexical symbols)
  - $S \in V$ is the start symbol (sentence, program)
  - P is a set of grammar rules or productions of the form:

    $A \rightarrow \gamma$ (the productions of A)

    where

    $A \in V$ is the head of the production

    "$\rightarrow$" is the production symbol

    $\gamma \in \{V \cup \Sigma\}^*$ is the body of the production

Formal definition:examples

- CFG for *Pal*
  - $G_{pal} = (\{P\}, \{0, 1\}, P,$

    $\{P \rightarrow \Lambda, P \rightarrow 0, P \rightarrow 1, P \rightarrow 0P0, P \rightarrow 1P1\})$
  - Compact notation for P: $P \rightarrow \Lambda \mid 0 \mid 1 \mid 0P0 \mid 1P1$
- CFG for 0^n1^n
  - $G_{pal} = (\{P\}, \{0, 1\}, P, \{P \rightarrow \Lambda, P \rightarrow 0P1\})$
- CFG for L_{exp}
  - $G_{exp} = (\{E\}, \{+, *, (,), a\}, E, P)$

    Where $P = \{E \rightarrow E + E \mid E * E \mid (E) \mid a\}$

Productions

- If α is a string of the form

  $\alpha_1 A \alpha_2$

  and there is a production of form

  $A \rightarrow \gamma$

  then α can by substituted or rewritten by β of form

  $\alpha_1 \gamma \alpha_2$
- We say that α derives β or β is derived from α in one step in G:

  $\alpha \Rightarrow_G \beta$
- Why context-free?
  - Substitution can be performed regardless the form of α_1 and α_2

Derivations of CFG

- Derivation in *Pal*:
  - $P \Rightarrow_G 0P0 \Rightarrow_G 01P10 \Rightarrow_G 01\Lambda10 = 0110$
- Derivation in $L = 0^n1^n$
  - $P \Rightarrow 0P1 \Rightarrow 00P11 \Rightarrow 00\Lambda11 = 0011$
- If it is clear what is G, we just write "$\Rightarrow$"

Derivations of CFG

- *-derivation: derivations in zero or more steps in a grammar G:

  $\alpha \Rightarrow^*_G \beta$
  - either $\alpha = \beta$
  - or there is a $k \geq 1$ and strings $\alpha_0, \alpha_1, \ldots \alpha_k$, with $\alpha_0 = \alpha$ and $\alpha_k = \beta$ so that $\alpha_i \Rightarrow_G \alpha_{i+1}$ for every i such that $0 \leq i \leq k - 1$
- Examples:
  - $P \Rightarrow^*_{Pal} 0110$
  - $P \Rightarrow^*_L 0011$

How many derivations are there?

- *Exp* is a CFG
  - $G_{exp} = (\{E\}, \{+, *, (,), a\}, E, P)$

    where $P = \{E \rightarrow E + E \mid E * E \mid (E) \mid a\}$
- A derivation of $a + (a * a) \in Exp$
  - $E \Rightarrow E + E \Rightarrow a + E \Rightarrow a + (E) \Rightarrow a + (E * E)$

    $\Rightarrow a + (E * a) \Rightarrow a + (a * a)$
  - There can be many ways to derive a string!
  - Are they all equivalent?

Leftmost and right most derivations

- Leftmost derivation: $\Rightarrow_{lm}$
  - $E \Rightarrow E + E \Rightarrow a + E \Rightarrow a + (E) \Rightarrow a + (E * E)$
    $\Rightarrow a + (a * E) \Rightarrow a + (a * a)$
  - $E \Rightarrow^*_{lm} a + (a * a)$

- Rightmost derivation: $\Rightarrow_{rm}$
  - $E \Rightarrow E + E \Rightarrow E + (E) \Rightarrow E + (E * E) \Rightarrow E + (E * a)$
    $\Rightarrow E + (a * a) \Rightarrow a + (a * a)$
  - $E \Rightarrow^*_{rm} a + (a * a)$

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Syntactic structure

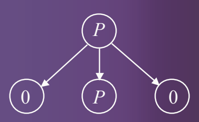
- Derivation in *Pal*:
  - P

Productions: $P \rightarrow \Lambda \mid 0 \mid 1 \mid 0P0 \mid 1P1$

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Syntactic structure

- Derivation in *Pal*:
  - $P \Rightarrow_G 0P0$

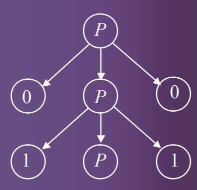


Productions: $P \rightarrow \Lambda \mid 0 \mid 1 \mid 0P0 \mid 1P1$

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Syntactic structure

- Derivation in *Pal*:
  - $P \Rightarrow_G 0P0 \Rightarrow_G 01P10$

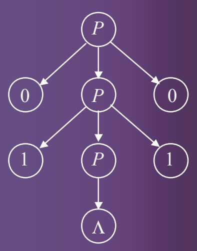


Productions: $P \rightarrow \Lambda \mid 0 \mid 1 \mid 0P0 \mid 1P1$

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Syntactic structure

- Derivation in *Pal*:
  - $P \Rightarrow_G 0P0 \Rightarrow_G 01P10 \Rightarrow_G 01\Lambda10$

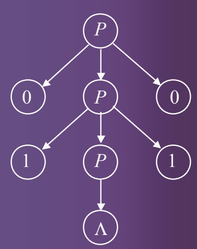


Productions: $P \rightarrow \Lambda \mid 0 \mid 1 \mid 0P0 \mid 1P1$

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Syntactic structure

- Derivation in *Pal*:
  - $P \Rightarrow_G 0P0 \Rightarrow_G 01P10 \Rightarrow_G 01\Lambda10 = 0110$

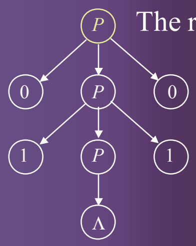


Productions: $P \rightarrow \Lambda \mid 0 \mid 1 \mid 0P0 \mid 1P1$

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Syntactic structure

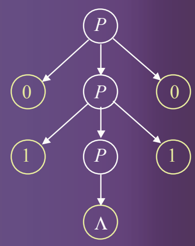
- Derivation in *Pal*:
  - $P \Rightarrow_G 0P0 \Rightarrow_G 01P10 \Rightarrow_G 01\Lambda10 = 0110$



The root

Syntactic structure

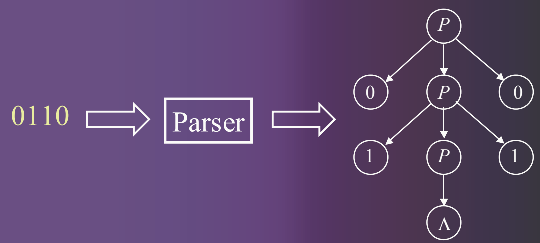
- Derivation in *Pal*:
  - $P \Rightarrow_G 0P0 \Rightarrow_G 01P10 \Rightarrow_G 01\Lambda10 = 0110$



The yield

Parsing

- Parsing: Recursive inference to obtain the syntactic structure of a string in the language



Derivations and Parse Trees

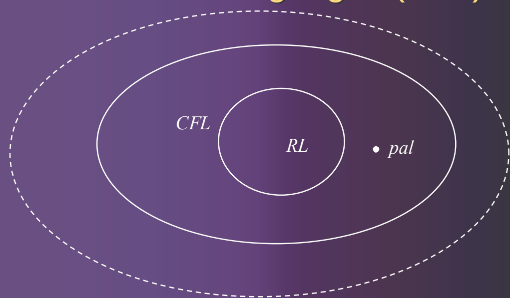
- If there is a derivation there is a recursive inference
- If there is a recursive inference there is a parse tree
- If there is a parse three there are leftmost and rightmost derivations
- If there are leftmost and rightmost derivations there is a derivation!

The language of a grammar

- Let $G = (V, \Sigma, S, P)$ be a CFG. The language generated by G is:

  $L(G) = \{x \in \Sigma^* \mid S \Rightarrow^*_G x\}$
- A language L is a *context-free language* (*CFL*) if there is a CFG G so that $L = L(G)$
- Sentential forms: derivations from the start symbol
- $L(G)$ consists of the sentential forms in Σ^*
  - Derivations from the start symbol that have no variables

Context-free Languages (CFL)



- All *RL* are *CFL* but not all *CFL* are *RL*
- There are also languages which are not *CFL*