Pumping Lemma (for regular languages)

To show that a language is *not* regular

- For any <u>infinite</u> regular language L,
- there exists a positive integer n_0 such that
- for any $w \in L$ such that $|w| \geq n_0$,
- there exists a decomposition $w = xyz$ where $|xy| \leq n_0$ and $|y| \geq 1$ such that
- for any $i \geq 0$,
- $xy^iz \in L$

Using Pumping Lemma

Play a falsifier: This is how this lemma is used.

- To show that 0^n1^n is *not* regular
- Anticipate any $n_0 \geq 1$
- Choose $0^n1^n \in L$ such that $2n \geq n_0$
- Anticipate any decomposition $w = xyz$:
  - Case 1: $0^a\ \ 0^b\ \ 1^n$ $(b \geq 1)$
  - Case 2: $0^n\ \ 1^c\ \ 1^d$ $(c \geq 1)$
  - Case 3: $0^a\ \ (0^b1^c)\ \ 1^d$ $(b + c \geq 1)$
- Choose $i = 2$
- $0^a0^{2b}1^n \notin L$, $0^n1^{2c}1^d \notin L$, and $0^a(0^b1^c)^21^d \notin L$

$w\$w^R$ is not regular

Play a falsifier

- To show that $w\$w^R$ is *not* regular
- Anticipate any $n_0 \geq 1$
- Choose $w\$w^R \in L$ such that $|w\$w^R| \geq n_0$
- Anticipate any decomposition $w\$w^R = x\underline{y}z$ where $|xy| \leq n_0$ and $|y| \geq 1$:
  - Case 1 (y includes \$): $w_1\ \ (w_2\$w_2^R)\ \ w_1^R$ $[|w_2| \geq 0]$
  - Case 2 (y does not include \$): $w_1\ \ w_2\ \ (w_3\$w^R)$
- Choose $i = 2$
- $w_1(w_2\$w_2^R)^2w_1^R \notin L$, $w_1w_2^2w_3\$w^R \notin L$

ww^R is not regular

Play a falsifier

- To show that ww^R is *not* regular
- Anticipate any $n_0 \geq 1$
- Choose $ww^R \in L$ such that $ww^R = 0^k1uu^R10^k$ where $k > n_0$
- Anticipate any decomposition $ww^R = x\underline{y}z$ where $|xy| \leq n_0$ and $|y| \geq 1$:
  - $0^p\ \ 0^q\ \ (0^r1uu^R10^k)$ where $p + q + r = k$ and $q \geq 1$
- Choose $i = 2$
- $0^p(0^q)^20^r1uu^R10^k \notin L$

Unit C5: Overview

- Distinguish CF and non-CF languages
  - Pumping Lemma (for CF languages)
- Analyze the effects of combining subproblems (regular or CF)
  - Closure properties
- Wrap up Module C

Context-Free? [general]

- HTML/XML
- Processing recursive function calls
- Depth-first tree traverse
- Overhauling (repairing a physical device)
- Sample Problem #20 “respectively”
- Dishwashing
- Teaching a course

Context-Free? [formal]

- $0^n1^n2^n$
- $0^m1^n2^m2^n$
- $\{ww^Rw \mid w^R$ is the reverse of $w \in \{0, 1\}^*\}$
- $\{ww \mid w \in \{0, 1\}^*\}$

Analysis of CF-ness

- To Show that L is CF
  - Proof by existence: Give a CFG or PDA
- To Show that L is *not* CF
  - Need to prove that no CFG (or PDA) can generate the language
  - Demonstrate that some property of CFLs cannot hold

Main Property of CFLs

- *Parenthesis matching*
- Use of a single stack
- Arbitrary nesting is possible
  - 0^n1^n, 0^n1^{n+m}, 0^n1^{2n}
  - $0^n2^m1^n$, $0^n2^m3^m1^n$
  - $0^k2^m0^{(n-k)}1^n$, $0^k2^m3^m0^{(n-k)}1^n$

cf. pumping air into the lung

Pumping Lemma (for CFLs)

To show that a language is *not* CF

- For any CFL L,
- there exists a positive integer n_0 such that
- for any $w \in L$ such that $|w| \geq n_0$,
- there exists a decomposition $w = abcde$ where $|bcd| \leq n_0$ and $|bd| \geq 1$ such that
- for any $i \geq 0$,
- $ab^icd^ie \in L$

$0^n1^n2^n$ is *not* CF

Play a falsifier

- To show that $0^n1^n2^n$ is *not* CF
- Anticipate any $n_0 \geq 1$
- Choose $0^n1^n2^n \in L$ such that $|0^n1^n2^n| \geq 3n_0$
- Anticipate any decomposition $0^n1^n2^n = a\underline{b}c\underline{d}e$ where $|\underline{bcd}| \leq n_0$ and $|\underline{bd}| \geq 1$:
  - Case 1: $\underline{b}c\underline{d}$ spans part of 0^n1^n
  - Case 2: $\underline{b}c\underline{d}$ spans part of 1^n2^n
- Choose $i = 2$
- At least one of 0 and 1 is pumped: i.e., $0^{n+1}1^n2^n$ or $0^n1^{n+1}2^n$ or $0^{n+1}1^{n+1}2^n$

Group Exercise 1

Context-free? Justify.

A. $\{0^i1^j2^k \mid i = j + k\}$

B. $\{0^i1^j2^k \mid i < j < k\}$

Combining Subproblems

- Compilers
  - Lexer: Regular
  - Parser: Context-free
  - Semantic analysis: Decidable (ignore the context-sensitive level)
  - Compiler as a whole: ?
- Bird flocking
  - Each bird (per frame): Regular
  - Entire simulation: ?

Dealing with Subproblems

- Serial connection ~ Concatenation
- Parallel/alternative connection ~ Union
- Repetition ~ Kleene closure
- Simultaneous requirements ~ Intersection
- Negation ~ Complement

Closure

- A class C is Closed under binary operation op.
  - The result of computing C_1 op C_2, where $C_1, C_2 \in C$, is still in C.
- A class C is Closed under unary operation op.
  - The result of computing op C_1, where $C_1 \in C$, is still in C.

Closure: Regular

- Concatenation
- Union
- Kleene closure
- Complement
- Intersection

Closure: Context-Free

- Concatenation
- Union
- Kleene closure
- Intersection
- Complement

Group Exercise 2

Group Exercise 2

CFLs closed under the following operations?

A. Intersection

B. Complement

Intersection of CFLs

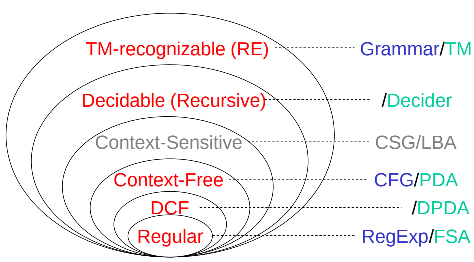
- $L_1 = 0^n 1^n 2^i \in$ CFL
- $L_2 = 0^i 1^n 2^n \in$ CFL
- $L_1 \cap L_2 = 0^n 1^n 2^n \notin$ CFL

Now, is CFLs closed under complement?

Unit Summary

- CF vs. non-CF
  - To show CF: Give a CFG or PDA
  - To show not CF: Use the Pumping Lemma
- Closure: Regular languages
  - Closed under concatenation, union, Kleene closure, complement, and intersection
- Closure: CFLs
  - Closed under concatenation, union, and Kleene closure
  - Not closed under complement or intersection

Extended Chomsky Hierarchy



Examples

- Pumping
  - Gas (automobile)
  - Air (human lung)
- Computer environment
  - Occurrence of arbitrary events
  - Modularity, recursion, interrupt, etc.
- Baby (newborn)
  - Limb movement
  - Remember things after sleep?

Module C Summary

- Chomsky hierarchy and properties
  - Regular languages: mostly closed and extremely fast to process
  - CFLs: partially closed and potential processing drawbacks
    - E.g., compilers only deal with DCFLs
  - Beyond: *more* closed but easily undecidable
- Distinguishing classes
  - Constructive vs. Pumping Lemma