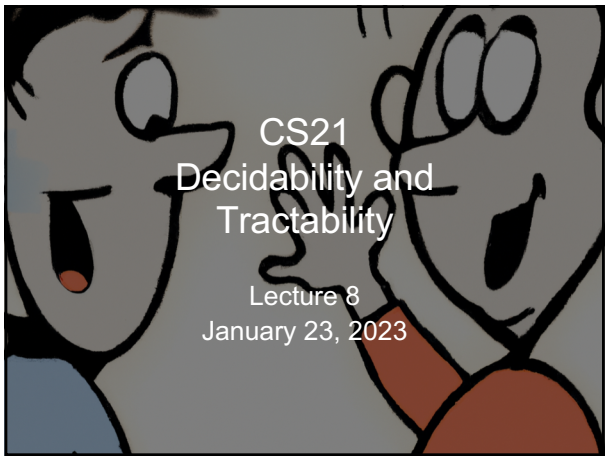


CS21 Decidability and Tractability

Lecture 8
January 23, 2023

1

Outline

- non-Context-Free languages via the CFL Pumping Lemma
- deciding CFLs
- deterministic PDAs

2

Pumping Lemma for CFLs

CFL Pumping Lemma: Let L be a CFL. There exists an integer p ("pumping length") for which every $w \in L$ with $|w| \geq p$ can be written as

$w = uvxyz$ such that

1. for every $i \geq 0$, $uv^ixy^iz \in L$, and
2. $|vy| > 0$, and
3. $|vxy| \leq p$.

3

CFL Pumping Lemma Example

Theorem: the following language is not context-free:

$L = \{a^nb^nc^n : n \geq 0\}$.

- Proof:
  - let p be the pumping length for L
  - choose $w = a^pb^pc^p$

    w = aaaa…abbbb…bcccc…c
  - w = uvxyz, with $|vy| > 0$ and $|vxy| \leq p$.

4

CFL Pumping Lemma Example

– possibilities:

w = aaaa…aaabbb…bbcccc…c
u v x y z

(if v, y each contain only one type of symbol, then pumping on them produces a string not in the language)

5

CFL Pumping Lemma Example

– possibilities:

w = aaaa…abbbb…bccccc…c
u v x y z

(if v or y contain more than one type of symbol, then pumping on them might produce a string with equal numbers of a's, b's, and c's – if vy contains equal numbers of a's, b's, and c's. But they will be out of order.)

6

CFL Pumping Lemma Example

Theorem: the following language is not context-free:

$L = \{xx : x \in \{0,1\}^*\}$.

- Proof:
  - let p be the pumping length for L
  - try w = 0^p10^p1
  - can this be pumped?

7

CFL Pumping Lemma Example

$L = \{xx : x \in \{0,1\}^*\}$.

- try w = $0^{2p}1^{2p}0^{2p}1^{2p}$
- w = uvxyz, with $|vy| > 0$ and $|vxy| \leq p$.
- case: vxy in first half.
  - then $uv^2xy^2z = 0??...?1??...?$
- case: vxy in second half.
  - then $uv^2xy^2z = ??...?0??...?1$
- case: vxy straddles midpoint
  - then $uv^0xy^0z = uxz = 0^{2p}1^i0^j1^{2p}$ with $i \neq 2p$ or $j \neq 2p$

8

Pumping Lemma for CFLs

CFL Pumping Lemma: Let L be a CFL. There exists an integer p ("pumping length") for which every $w \in L$ with $|w| \geq p$ can be written as

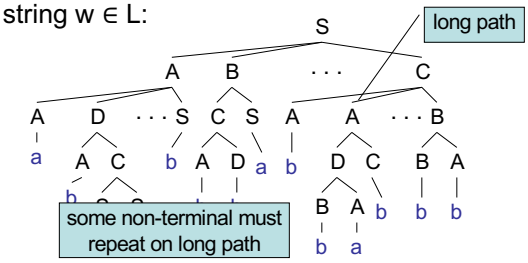
$w = uvxyz$ such that

1. for every $i \geq 0$, $uv^ixy^iz \in L$, and
2. $|vy| > 0$, and
3. $|vxy| \leq p$.

9

CFL Pumping Lemma

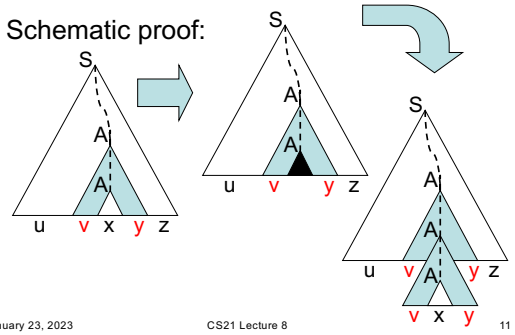
Proof: consider a parse tree for a very long string $w \in L$:



10

CFL Pumping Lemma

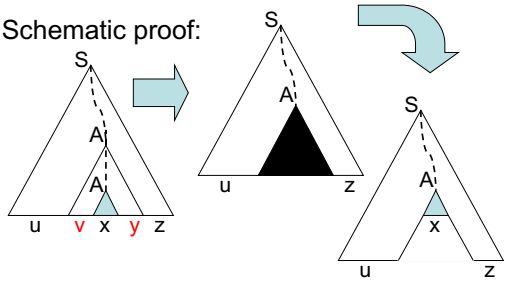
- Schematic proof:



11

CFL Pumping Lemma

- Schematic proof:



12

CFL Pumping Lemma

– how large should pumping length p be?
– need to ensure other conditions:

$|vy| > 0$ $|vxy| \le p$

– b = max # symbols on rhs of any production (assume $b \ge 2$)
– if parse tree has height ≤ h, then string generated has length ≤ b^h (so length > b^h implies height > h)

13

CFL Pumping Lemma

– let m be the # of nonterminals in the grammar
– to ensure path of length at least m+2, require

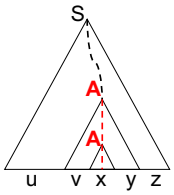
$|w| \ge \mathbf{p} = b^{m+2}$

– since $|w| > b^{m+1}$, any parse tree for w has height > m+1
– let T be the smallest parse tree for w
– longest root-leaf path must consist of ≥ m+1 non-terminals and 1 terminal.

14

CFL Pumping Lemma

– must be a repeated non-terminal **A** on long path
– select a repetition among the lowest m+1 non-terminals on path.
– pictures show that for every i ≥ 0, $uv^ixy^iz \in L$



– is |vy| > 0 ?
  • smallest parse tree T ensures
– is |vxy| ≤ p?
  • red path has length ≤ m+2, so ≤ b^{m+2} = p leaves

15

Chomsky Normal Form

- Useful to deal only with CFGs in a simple normal form
- Most common: Chomsky Normal Form (CNF)
- Definition: every production has form

$A \to BC$ or $S \to \varepsilon$ or $A \to a$

where A, B, C are any non-terminals (and B, C are not S) and a is any terminal.

16

Chomsky Normal Form

Theorem: Every CFL is generated by a CFG in Chomsky Normal Form.

Proof: exercise or in book…

17

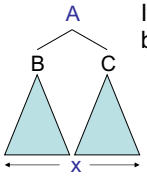
Deciding CFLs

- Useful to have an efficient algorithm to decide whether string x is in given CFL
  – e.g. programming language often described by CFG. Determine if string is valid program.
- If CFL recognized by deterministic PDA, just simulate the PDA.
  – but not all CFLs are (homework)…
- Can simulate NPDA, but this takes exponential time in the worst case.

18

Deciding CFLs

- Convert CFG into Chomsky Normal Form
- parse tree for string x generated by nonterminal A:



If $A \rightarrow^k x$ $(k > 1)$ then there must be a way to split x:

$x = yz$

- $A \rightarrow BC$ is a production and
- $B \rightarrow^i y$ and $C \Rightarrow^j z$ for $i, j < k$

19

Deciding CFLs

- An algorithm:

IsGenerated(x, A)

if |x| = 1, then return YES if $A \rightarrow x$ is a production, else return NO
for all n-1 ways of splitting x = yz
for all ≤ m productions of form $A \rightarrow BC$
if IsGenerated(y, B) and IsGenerated(z, C), return YES
return NO

- worst case running time?

20

Deciding CFLs

- worst case running time exp(n)
- Idea: avoid recursive calls
  - build table of YES/NO answers to calls to IsGenerated, in order of length of substring
  - example of general algorithmic strategy called dynamic programming
  - notation: x[i,j] = substring of x from i to j
  - table: T(i, j) contains

{A: A nonterminal such that $A \rightarrow^* x[i,j]$}

21

Deciding CFLs

IsGenerated($x = x_1x_2x_3\ldots x_n$, G)

for i = 1 to n
T[i, i] = {A: "$A \rightarrow x_i$" is a production in G}
for k = 1 to n - 1
for i = 1 to n - k
for k splittings x[i, i+k] = x[i,i+j]x[i+j+1, i+k]
T[i, i+k] = {A: "$A \rightarrow BC$" is a production in G and $B \in T[i,i+j]$ and $C \in T[i+j+1,i+k]$ }
output "YES" if $S \in T[1, n]$, else output "NO"

22

Deciding CFLs

IsGenerated($x = x_1x_2x_3\ldots x_n$, G)

for i = 1 to n
T[i, i] = {A: "$A \rightarrow x_i$" is a production in G}
for k = 1 to n - 1
for i = 1 to n - k
for k splittings x[i, i+k] = x[i,i+j]x[i+j+1, i+k]
T[i, i+k] = {A: "$A \rightarrow BC$" is a production in G and $B \in T[i,i+j]$ and $C \in T[i+j+1,i+k]$ }
output "YES" if $S \in T[1, n]$, else output "NO"

O(nm) steps

$O(n^3m^3)$ steps

23