Announcements

- HW2 due now
- HW3 is up
  feel free to use the web
due on Sunday
Recap: CFGs and CNF

Chomsky Normal Form:

Each rule in the grammar is either:

- $A \rightarrow BC$ \textcolor{red}{2 nonterminals where neither B or C is the start variable, or both are nonterminals}

- $A \rightarrow a$ \underline{where a is a terminal}

- No useless symbols
Why do we care?

1. Makes structure of parsing nice.

2. Computable: \(O(n^2)\) time to convert to CNF

3. Given CNF, can compute a parse tree in \(O(n^3)\) time (using dynamic programming).
Ex: Simple CFG

\[ S_0 \rightarrow A\, B \]

\[ A \rightarrow a\, A \mid \varepsilon \]

\[ B \rightarrow b\, A\, b\, A\, B \mid \varepsilon \]

\[ a^* \left( b\, a^*\, b\, a^* \right)^* \]

Convert:
1. useless states?
   - No
Eliminate ε-transitions

\[ S_0 \rightarrow AB \]

\[ A \rightarrow aA / \epsilon \]

\[ B \rightarrow bAbA \epsilon \]

\[ S_0 \rightarrow AB \mid A \mid B \mid \epsilon \]

\[ A \rightarrow aA \mid a \]

\[ B \rightarrow bAbA \mid bAbA / bb / bbB \]

\[ bAb \mid bbA \mid bbAB \mid bAbB \]
Eliminate unit pairs: \((S, A) + (S, B)\)

\[
\begin{align*}
S_0 & \rightarrow AB | A | B | \varepsilon \\
A & \rightarrow aA | a \\
B & \rightarrow bAbAB | bAbAbA | bb | bbbB \\
& \quad bAb | bbaA | bbbAB | bAbB
\end{align*}
\]
a: Get rid of non-single terminals
(eg. $X \rightarrow y^2$ becomes $X \rightarrow YZ + Y \rightarrow Y$)

$S_0 \rightarrow AB | aA | a$
$| bAb | bAB | bbb | bbb | bbb B$
$| bAb | bbb | bbb B | bbb B | bbb B | bbb B | bbb B | bbb B | bbb B | bbb B | bbb B |

$A \rightarrow XA | a$
$| bAb | bbb | bbb | bbb B$

$\overline{B} \rightarrow YAYAB |
| YAYA | bbb | bbb B$

$\overline{X} \rightarrow a$

$\overline{Y} \rightarrow b$

(replace any a with X
+ any b with Y)
46: Eliminate 3 or more non-terminal transitions

\[ B \rightarrow YAYAB \quad \text{(too long)} \]

\[ \begin{align*}
B & \rightarrow YM_1 \\
M_1 & \rightarrow AM_2 \\
M_2 & \rightarrow YM_3 \\
M_3 & \rightarrow AB
\end{align*} \]
Now, parsing. Consider \( \text{abbaababab} \).

In language?

How to parse:

1. \( S_0 \rightarrow AB \rightarrow aAB \rightarrow aB \rightarrow aabbAB \)

\[ S_o \rightarrow AB \]

\[ A \rightarrow aA / \varepsilon \]

\[ B \rightarrow bA bA bA \varepsilon \]

In CNF, would be deg 2.
Parse tree: (prev page)
CYK algorithm: build a table

Given a word \( w = w_1 w_2 w_3 w_4 \ldots w_k \), we'll look at all possible substrings \( w_i \), \( w_{ij} \), and look at how they can be parsed.

We'll build a table from the bottom up.
Ex: \[ S \rightarrow AB \mid BC \]
\[ A \rightarrow BA \mid a \]
\[ B \rightarrow CC \mid b \]
\[ C \rightarrow AB \mid a \]

Compute valid parse tree for ‘baaba’

Complexity n
Running times:
Say we have $n$ rules.
Converting to CNF: $O(n^2)$
Finding unit pairs
Running CYK: $O(n^3)$
Other parsing algorithms

CYK is still pretty slow, especially for large programming languages.

After it was developed, a lot of work was put into figuring out what grammars could have faster algorithms.

Two big (and useful) classes have linear time parsers: LL and LR.