CS 536 Announcements for Monday, March 6, 2023

Last Time
• review for Midterm 1

Today
• approaches to parsing
• bottom-up parsing
• CFG transformations
  • removing useless non-terminals
  • Chomsky normal form (CNF)
• CYK algorithm

Next Time
• wrap up CYK
• classes of grammars
• top-down parsing

Parsing: two approaches

Top-down / "goal driven"
• start at start nonterminal
• grow parse tree downward until entire sequence is matched

Bottom-up / "data driven"
• start with terminals (sequence)
• generate ever larger subtrees until get to single tree whose root is the start nonterminal

Example:

CFG:  
expr \rightarrow \text{expr} + \text{term} | \text{term}  
  
  \text{term} \rightarrow \text{term} * \text{ID} | \text{ID}  

Derive:  
\text{ID} + \text{ID}
Cocke – Younger – Kasami (CYK) algorithm

- Works bottom-up
- Time complexity: $O(n^3)$
- Requires grammar to be in Chomsky Normal Form

Chomsky Normal Form (CNF)

- All rules must be in one of two forms
  - $x \rightarrow T$
  - $x \rightarrow a b$
- Only rule allowed to derive epsilon is the start symbol $s$

Why CNF is helpful?

- Nonterminals in pairs
- Nonterminals (except start) can't derive epsilon

CYK: Dynamic Programming

$x \rightarrow T$

$x \rightarrow a b$
Running CYK

Track every viable subtree from leaf to root.
All subspans for a sequence (string) with 6 terminals
CYK Example

\[
\begin{align*}
  f &\rightarrow iw \\
  f &\rightarrow iy \\
  w &\rightarrow lx \\
  x &\rightarrow nr \\
  y &\rightarrow lr \\
  n &\rightarrow ID \\
  n &\rightarrow iz \\
  z &\rightarrow cn \\
  i &\rightarrow ID \\
  l &\rightarrow ( \\
  r &\rightarrow ) \\
  c &\rightarrow ,
\end{align*}
\]
Eliminating useless nonterminals

Avoid unnecessary work – remove useless rules
1. If a nonterminal cannot derive a sequence of terminal symbols, then it is useless
2. If a nonterminal cannot be derived from the start symbol, then it is useless

Nonterminals that cannot derive a sequence of terminal symbols
mark all terminal symbols
repeat
   if all symbols on the RHS of a production are marked
      mark the LHS nonterminal
until no more nonterminals can be marked

Example
s → x | y
x → ()
y → (y y)

Nonterminals that cannot be derived from the start symbol
mark the start symbol
repeat
   if the LHS of a production is marked
      mark all RHS nonterminals
until no more nonterminals can be marked

Example
s → a b
a → + | − | ε
b → digit | b digit
c → . b
Chomsky Normal Form

Four steps
- eliminate epsilon productions
- eliminate unit productions
- fix productions with terminal on RHS
- fix productions with > 2 nonterminals on RHS

Eliminate (most) epsilon productions
If nonterminal A immediately derives epsilon

Example 1
\[
f \rightarrow \text{ID}(a)
\]
\[
a \rightarrow \varepsilon
\]
\[
a \rightarrow n
\]
\[
n \rightarrow \text{ID}
\]
\[
n \rightarrow \text{ID}, n
\]

Example 2
\[
x \rightarrow a X a Y a
\]
\[
a \rightarrow \varepsilon
\]
\[
a \rightarrow Z
\]
Chomsky Normal Form (cont.)

Eliminate unit productions
Productions of the form $a \rightarrow b$ are called unit productions

Example

\[
\begin{align*}
f & \rightarrow \text{ID} \ ( \ a \ ) \\
f & \rightarrow \text{ID} \ ( \ ) \\
a & \rightarrow n \\
n & \rightarrow \text{ID} \\
n & \rightarrow \text{ID} \ , \ n
\end{align*}
\]

Fix RHS terminals
For productions with terminals and something else on the RHS

Example

\[
\begin{align*}
f & \rightarrow \text{ID} \ ( \ n \ ) \\
f & \rightarrow \text{ID} \ ( \ ) \\
n & \rightarrow \text{ID} \\
n & \rightarrow \text{ID} \ , \ n
\end{align*}
\]
Chomsky Normal Form (cont.)

Fix RHS nonterminals
For productions with > 2 nonterminals on the RHS

Example