CKY (11/12)

CS 585, Fall 2015
Introduction to Natural Language Processing
http://people.cs.umass.edu/~brenocon/inlp2015/

Brendan O’Connor
College of Information and Computer Sciences
University of Massachusetts Amherst
CKY

Grammar
Adj -> yummy
NP -> foods
NP -> store
NP -> NP NP
NP -> Adj NP

For cell [i,j] (loop through them bottom-up)
For possible splitpoint k=(i+1)..<j-1):
  For every B in [i,k] and C in [k,j],
    If exists rule A -> B C,
      add A to cell [i,j]  (Recognizer)
    ... or ...
      add (A,B,C, k) to cell [i,j]  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.
For cell \([i,j]\) (loop through them bottom-up):
For possible splitpoint \(k=(i+1) .. (j-1)\):
For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
If exists rule \(A \rightarrow B \; C\),
\add A to cell \([i,j]\) \((\text{Recognizer})\)
... or ...
\add (A,B,C, k) to cell \([i,j]\) \((\text{Parser})\)
For cell [i,j] (loop through them bottom-up)
   For possible splitpoint k=(i+1)..(j-1):
      For every B in [i,k] and C in [k,j],
         If exists rule A -> B C,
            add A to cell [i,j] (Recognizer)
         ... or ...
            add (A,B,C, k) to cell [i,j] (Parser)
**Grammar**

- Adj -> yummy
- NP -> foods
- NP -> store
- NP -> NP NP
- NP -> Adj NP

---

**CKY**

For cell \([i,j]\) (loop through them bottom-up)

For possible splitpoint \(k = (i+1)\ldots(j-1)\):

- For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
  - If exists rule \(A \rightarrow B\ C\),
    - **add** \(A\) to cell \([i,j]\)  (**Recognizer**)  
      - or ...
    - **add** \((A,B,C,\ k)\) to cell \([i,j]\)  (**Parser**)
**Grammar**
- Adj -> yummy
- NP -> foods
- NP -> store
- NP -> NP NP
- NP -> Adj NP

For cell \([i,j]\) (loop through them bottom-up)
For possible splitpoint \(k=(i+1)\)..<\(j-1\):
    For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
        If exists rule \(A -> B C\),
            add \(A\) to cell \([i,j]\) (Recognizer)
            ... or ...
        add \((A,B,C, k)\) to cell \([i,j]\) (Parser)

**Recognizer**: per span, record list of possible nonterminals

**Parser**: per span, record possible ways the nonterminal was constructed.
For cell \([i,j]\) (loop through them bottom-up):
For possible splitpoint \(k=(i+1)\ldots(j-1)\):
For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
If exists rule \(A \rightarrow B C\),
\begin{align*}
&\text{add } A \text{ to cell } [i,j] \quad (\text{Recognizer}) \\
&\quad \text{... or ...} \\
&\quad \text{add } (A,B,C, k) \text{ to cell } [i,j] \quad (\text{Parser})
\end{align*}
For cell $[i,j]$ (loop through them bottom-up)

For possible splitpoint $k = (i+1)..(j-1)$:

For every $B$ in $[i,k]$ and $C$ in $[k,j]$,

If exists rule $A \rightarrow B \; C$,

add $A$ to cell $[i,j] \quad \text{(Recognizer)}$

... or ...

add $(A,B,C, \; k)$ to cell $[i,j] \quad \text{(Parser)}$
**Grammar**
- Adj -> yummy
- NP -> foods
- NP -> store
- NP -> NP NP
- NP -> Adj NP

For cell [i,j] (loop through them bottom-up)
For possible splitpoint $k=(i+1)\ldots(j-1)$:
For every B in [i,k] and C in [k,j],
If exists rule $A \rightarrow B\; C$,
- **add** $A$ to cell [i,j]  (**Recognizer**)
- ... or ...
- **add** $(A,B,C,\; k)$ to cell [i,j]  (**Parser**)

**Recognizer:** per span, record list of possible nonterminals

**Parser:** per span, record possible ways the nonterminal was constructed.
For cell \([i,j]\) (loop through them bottom-up)

For possible splitpoint \(k=(i+1)\ldots(j-1)\):

For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),

If exists rule \(A \rightarrow B C\),

\[
\text{add } A \text{ to cell } [i,j] \quad (\text{Recognizer})
\]

... or ...

\[
\text{add } (A,B,C, k) \text{ to cell } [i,j] \quad (\text{Parser})
\]
For cell $[i,j]$ (loop through them bottom-up)
For possible splitpoint $k=(i+1)..(j-1)$:
For every $B$ in $[i,k]$ and $C$ in $[k,j]$,
If exists rule $A \rightarrow B \ C$,
\[\text{add } A \text{ to cell } [i,j] \quad \text{(Recognizer)}\]
... or ...
\[\text{add } (A,B,C, k) \text{ to cell } [i,j] \quad \text{(Parser)}\]

**Grammar**
- Adj $\rightarrow$ yummy
- NP $\rightarrow$ foods
- NP $\rightarrow$ store
- NP $\rightarrow$ NP NP
- NP $\rightarrow$ Adj NP

**Recognizer:** per span, record list of possible nonterminals

**Parser:** per span, record possible ways the nonterminal was constructed.
For cell \([i,j]\) (loop through them bottom-up)
  For possible splitpoint \(k=(i+1)\ldots(j-1)\):
    For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
      If exists rule \(A \rightarrow B \ C\),
      \textit{add} \(A\) to cell \([i,j]\) \(\textup{(Recognizer)}\)
    \ldots or \ldots
  
  \textit{add} \((A,B,C, \ k)\) to cell \([i,j]\) \(\textup{(Parser)}\)

**Recognizer:** per span, record list of possible nonterminals

**Parser:** per span, record possible ways the nonterminal was constructed.
For cell [i,j] (loop through them bottom-up)
For possible splitpoint k=(i+1)..(j-1):
For every B in [i,k] and C in [k,j],
If exists rule A -> B C,

\[\text{add } A \text{ to cell } [i,j] \quad (\text{Recognizer})\]

... or ...

\[\text{add } (A,B,C, \ k) \text{ to cell } [i,j] \quad (\text{Parser})\]
For cell \([i,j]\) (loop through them bottom-up)
For possible splitpoint \(k=(i+1)\ldots(j-1)\):
For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
If exists rule \(A \rightarrow B \ C\),
\textit{add} \(A\) to cell \([i,j]\) \(\textit{(Recognizer)}\)
... or ...
\textit{add} \((A,B,C,\ k)\) to cell \([i,j]\) \(\textit{(Parser)}\)

\textit{Recognizer:} per span, record list of possible nonterminals

\textit{Parser:} per span, record possible ways the nonterminal was constructed.

\begin{itemize}
  \item Grammar
  \begin{itemize}
    \item Adj \(\rightarrow\) yummy
    \item NP \(\rightarrow\) foods
    \item NP \(\rightarrow\) store
    \item NP \(\rightarrow\) NP NP
    \item NP \(\rightarrow\) Adj NP
  \end{itemize}
\end{itemize}
For cell \([i,j]\) (loop through them bottom-up)
For possible splitpoint \(k=(i+1)\ldots(j-1)\):
For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
If exists rule \(A \rightarrow B C\),
\(\text{add } A\) to cell \([i,j]\)  \(\text{(Recognizer)}\)
\(...\text{ or }...\)
\(\text{add } (A,B,C, k)\) to cell \([i,j]\)  \(\text{(Parser)}\)
For cell \([i,j]\) (loop through them bottom-up)

For possible splitpoint \(k=(i+1) \ldots (j-1)\):

For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),

If exists rule \(A \rightarrow B C\),

add \(A\) to cell \([i,j]\) \((\text{Recognizer})\)

... or ...

add \((A,B,C,k)\) to cell \([i,j]\) \((\text{Parser})\)
**For cell** [i,j] (loop through them bottom-up)  
**For possible splitpoint** k=(i+1)..(j-1):  
**For every** B in [i,k] and C in [k,j],  
If exists rule A -> B C,  
*add A to cell* [i,j]  
*(Recognizer)*  
... or ...  
*add (A,B,C, k) to cell* [i,j]  
*(Parser)*
For cell \([i,j]\) (loop through them bottom-up)
For possible splitpoint \(k=(i+1)\ldots(j-1)\):
For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
If exists rule \(A \to B \ C\),
add \(A\) to cell \([i,j]\) \((\text{Recognizer})\)
... or ...
add \((A,B,C, \ k)\) to cell \([i,j]\) \((\text{Parser})\)
How do we fill in $C(1,2)$?

For cell $[i,j]$
For possible splitpoint $k=(i+1)..(j-1)$:
For every $B$ in $[i,k]$ and $C$ in $[k,j]$,
If exists rule $A \rightarrow B \ C$,
$\textit{add} \ A$ to cell $[i,j]$

Computational Complexity?
For cell \([i,j]\)
For possible splitpoint \(k=(i+1)\ldots(j-1)\):
For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
If exists rule \(A \rightarrow B\ C\),
\(\text{add}\ A\ \text{to cell}\ [i,j]\)

How do we fill in \(C(1,2)\)?
Put together \(C(1,1)\) and \(C(2,2)\).
For cell \([i,j]\)
For possible splitpoint \(k=(i+1)\ldots(j-1)\):
For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
If exists rule \(A \rightarrow B C\), add \(A\) to cell \([i,j]\)

How do we fill in \(C(1,3)\)?

Computational Complexity?
How do we fill in $C(1,3)$?

One way …

For cell $[i,j]$
For possible splitpoint $k = (i+1) .. (j-1)$:
  For every $B$ in $[i,k]$ and $C$ in $[k,j]$,
  If exists rule $A \rightarrow B C$,
  add $A$ to cell $[i,j]$

Computational Complexity?
For cell \([i,j]\)
For possible splitpoint \(k=(i+1)\ldots(j-1)\):
For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
If exists rule \(A \rightarrow B C\),
\(\text{add} \ A\) to cell \([i,j]\)

How do we fill in \(C(1,3)\)?

One way …
Another way.

Computational Complexity?
For cell \([i,j]\)
For possible splitpoint \(k=(i+1)..(j-1)\):
For every \(B\) in \([i,k]\) and \(C\) in \([k,j]\),
If exists rule \(A \rightarrow B\ C\),
\textit{add} \(A\) to cell \([i,j]\)

How do we fill in \(C(1,n)\)?

Computational Complexity?
For cell $[i,j]$
   For possible splitpoint $k=(i+1) \ldots (j-1)$:
      For every $B$ in $[i,k]$ and $C$ in $[k,j]$,
         If exists rule $A \rightarrow B \ C$,
            add $A$ to cell $[i,j]$

How do we fill in $C(1,n)$?

$n - 1$ ways!

Computational Complexity?
For cell $[i,j]$
   For possible splitpoint $k=(i+1)\ldots(j-1)$:
     For every $B$ in $[i,k]$ and $C$ in $[k,j]$,
       If exists rule $A \rightarrow B\;C$,
         add $A$ to cell $[i,j]$