

Grundlegende Parsingalgorithmen

Chart-Parsing: CYK-Parser

Kurt Eberle

k.eberle@lingenio.de

(Viele Folien, Teile von Folien, Materialien von **Helmut Schmid's**
Parsing-Kurs WS14 Tübingen, u.a.)

26. Februar, 2020

Überblick

Chart-Parser

CYK Parser

CYK-Erkenner

Varianten des CYK-Parsers

Nachteile

Überblick

Chart-Parser

CYK Parser

CYK-Erkenner

Varianten des CYK-Parsers

Nachteile

Chart-Parser

- ▶ Alle Parsing-Zwischenergebnisse (d.h. schon gefundene Konstituenten) werden in eine Tabelle (die Chart) eingetragen
- ▶ mit Feldern $Chart[i, j]$ (mit Zeilen i und Spalten j) wobei
- ▶ $A \in Chart[i, j]$ gdw $A \xRightarrow{*} w_{i+1} \dots w_j$
- ▶ Varianten: CYK, Earley, Left-Corner

Chart

	1	2	3	4	5
0	1	3	6	10	15
1		2	5	9	14
2			4	8	13
3				7	12
4					11

Überblick

Chart-Parser

CYK Parser

CYK-Erkenner

Varianten des CYK-Parsers

Nachteile

CYK Parser

- ▶ unabhängig entwickelt von Cocke, Younger und Kasami (Kasami 1965, Younger 1967, Cocke 1970)
- ▶ verlangt Grammatiken in Chomsky-Normalform:
Regeln der Art $A \rightarrow a$ oder $A \rightarrow B C$
Transformation:
allgemeine CF-Grammatik $\rightarrow$ Grammatik in CNF
- ▶ Allgemeines Prinzip:
Die Chart-Felder werden Bottom-up gefüllt.
Ein Symbol A wird in $Chart[i, j]$ eingeführt, wenn
 1. $i = j - 1$ und $A \rightarrow x_j$
 2. $i < j - 1$ und
$$\exists_{A \rightarrow BC \in P} \exists_k B \in Chart[i, k], C \in Chart[k, j]$$

Chart

	1	2	3	4	5
0	1	3	6	10	15
1		2	5	9	14
2			4	8	13
3				7	12
4					11

Überblick

Chart-Parser

CYK Parser

CYK-Erkenner

Varianten des CYK-Parsers

Nachteile

CYK-Erkennen

Recognize($x_1 \dots x_n$)

```

For  $k := 1$  to  $n$                                 % for all input symbols
   $t_{k-1,k} := \{A \mid A \rightarrow x_k \in P\}$     % insert terminal rules
  For  $i := k - 2$  down to  $0$                       % for all fields in the  $k^{th}$  column
     $t_{ik} := \{\}$ 
    For  $j := i + 1$  to  $k - 1$                     % for all splittings of  $x_{i+1} \dots x_k$ 
      For all  $A \rightarrow B \ C \in P$             % for all grammar rules
        If  $B \in t_{ij}$  and  $C \in t_{jk}$            % if  $B \xRightarrow{*} x_{i+1} \dots x_j$  &  $C \xRightarrow{*} x_{j+1} \dots x_k$ 
          then  $t_{ik} := t_{ik} \cup \{A\}$  %  $A \xRightarrow{*} x_{j+1} \dots x_k$ 
  if  $S \in t_{0n}$  then accept, else report error
  
```

Überblick

Chart-Parser

CYK Parser

CYK-Erkenner

Varianten des CYK-Parsers

Nachteile

CYK Parser1

Parse($x_1 \dots x_n$)

Recognize($x_1 \dots x_n$)

If $S \in t_{0,n}$ then

$p_{ik} := \{\}$ for all $0 \leq i < k \leq n$

return Gen($S, 0, n$)

else

return error

% build chart

% analysis exists

% initialization

% create parse forest

CYK Parser1

Gen(A, i, k)

```

If  $\exists n \in p_{ik}$  such that  $n = \langle A, l \rangle$  then    % parse forest already generated?
    return  $n$ 
else if  $i = k - 1$  then                        % terminal symbol
     $l := \{x_k\}$                                % leaf node
else
     $l := \{\}$ 
    For  $j := i + 1$  to  $k - 1$                     % all splittings
        For all  $A \rightarrow B \ C \in P$ 
            If  $B \in t_{ij}$  and  $C \in t_{jk}$  then    % analysis for A found
                 $n_1 := \text{Gen}(B, i, j)$           % generate parse forest for B
                 $n_2 := \text{Gen}(C, j, k)$           % generate parse forest for C
                 $l := l \cup \{\langle n_1, n_2 \rangle\}$ 
     $n := \langle A, l \rangle$                         % create new node
     $p_{ik} := p_{ik} \cup \{n\}$                   % store node in table
Return  $n$ 

```

CYK Parser2

Parse($x_1 \dots x_n$)

For $k := 1$ to n

$t_{k-1k} := \{\}$

For all $A \rightarrow x_k \in P$

$p_{k-1k} := p_{k-1k} \cup \{\langle A, \{x_k\} \rangle\}$

For $i := k - 2$ down to 0 % for all fields in the k^{th} column

$t_{ik} := \{\}$

For $j := i + 1$ bis $k - 1$ % for all splittings of $x_{i+1} \dots x_k$

For all $A \rightarrow B \ C \in P$

If $n_l \in p_{ij}$ such that $n_l = \langle B, l \rangle$

and $n_r \in p_{jk}$ such that $n_r = \langle C, l' \rangle$

Store(A, i, j, n_l, n_r)

If $\exists n \in p_{0n}$ such that $n = \langle S, l \rangle$ then

return n

else

return error

CYK Parser2

Store(A, i, k, n_l, n_r)

If $\exists \langle A, l \rangle \in p_{ik}$ then

$l := l \cup \{\langle n_l, n_r \rangle\}$

else

$p_{ik} := p_{ik} \cup \{\langle A, \{\langle n_l, n_r \rangle\} \rangle\}$

Überblick

Chart-Parser

CYK Parser

CYK-Erkenner

Varianten des CYK-Parsers

Nachteile

Nachteile des CYK Parsers

- ▶ Nachteil 1: Einschränkung auf CNF-Grammatiken
- ▶ Nachteil 2: Zu viele Konstituenten werden in die Chart eingetragen, weil keine Top-Down-Information genutzt wird.
- ▶ Lösung: Earley Parser