

# Chart Parsing

## Parsing ISCL-BA-06

Çağrı Çöltekin  
ccoltekin@fs.uni-tuebingen.de

University of Tübingen  
Seminar für Sprachwissenschaft

Winter Semester 2020/21

version: 04/07/2020 02:34

# Parsing so far

- We can formulate parsing as
  - Top-down: begin with the start symbol, try to produce the input string to be parsed
  - Bottom up: begin with the input, and try to reduce it to the start symbol
- For both options, we have seen examples of chart parser
- Parsing can also be *directional* or *non-directional*
- In this lecture, we introduce a general mechanism for chart parsing that has all these forms of parsing methods as special cases

Ç. Çöltekin, NR / University of Tübingen

Winter Semester 2020/21 2 / 13

# The overall idea

- We adopt Early-like chart entries of the form:  $X \rightarrow \alpha \cdot \beta [i, j]$  where,
  - $i$  and  $j$  are indexes starting from 0 (0 indicating beginning of the input string)
  - The chart entry indicates  $\alpha$  is found between  $i$  and  $j$ , we are looking for a  $\beta$  starting from  $j$
- At any time, we have two sets of items:
  - active items are those we expect to complete
  - inactive items are those with a dot at the end
- The goal is to complete  $S \rightarrow \dots [0, n]$

Ç. Çöltekin, NR / University of Tübingen

Winter Semester 2020/21 2 / 11

# Components of a typical chart parsing algorithm

- Besides the chart, we keep an agenda of 'unexplored items'
- A set of inference rules determine how to modify the chart when processing items from the agenda
- Typically inference rules are similar to completion process of Earley parser
- The following inference rule is part of every chart parser (so-called 'fundamental rule' of chart parsing)
  - If there is an inactive item of the form  $A \rightarrow \alpha \cdot$  and an active item of the form  $B \rightarrow \beta \cdot \gamma$  add item  $B \rightarrow \beta \cdot \gamma$
- We also need a strategy for selecting the items from the agenda and applying the inference rules
- Depending on the data structure used for the agenda, and order of processing of inference rules, we may get different types of parsers

Ç. Çöltekin, NR / University of Tübingen

Winter Semester 2020/21 3 / 13

# The sketch of a chart parsing algorithm

```

1: Initialize A (agenda) and C (chart)
2: repeat
3:   i ← next(A)
4:   if i ∈ C then
5:     discard i
6:   else
7:     apply all inference rules to i
8:     place new items in A
9:   place the item in C
10: until A is empty
            
```

- Very simple, but unspecified parts:
  - Initialization
  - Inference rules
  - The order of items received from the agenda
- An item is put into chart only after all inferences from it are in the chart or in the agenda
- Chart is a set, items do not repeat

Ç. Çöltekin, NR / University of Tübingen

Winter Semester 2020/21 4 / 11

# Bottom-up chart parsing

- Single additional inference rule:
  - If a new item has the form  $A \rightarrow \alpha \cdot$ , add  $B \rightarrow A \cdot \beta$  for each rule  $B \rightarrow A \beta$  in the grammar.
- Initialization:
  - Empty chart
  - Place  $P \rightarrow w_i [i-1, i]$  in the agenda for all word  $w_i$  ( $P$  is the pre-terminal symbol, typically the PCGS tag in CL)
  - If there are  $\epsilon$  rules, add  $P \rightarrow \epsilon [i, i]$  for all  $P \rightarrow \epsilon$  in the grammar, for  $i$  in  $[0, n]$
- Choice of agenda does not matter. A stack is typical, but a queue or a priority queue is also an option

Ç. Çöltekin, NR / University of Tübingen

Winter Semester 2020/21 5 / 13

# Example: bottom-up chart parsing

```

grammar
S → NP VP
NP → Pron N
NP → Pron
VP → V NP
VP → V
VP → V S
Pron → I
Pron → her
V → saw
N → duck
V → duck
            
```

Ç. Çöltekin, NR / University of Tübingen

Winter Semester 2020/21 6 / 11

# Example: bottom-up chart parsing

```

grammar
S → NP VP
NP → Pron N
NP → Pron
VP → V NP
VP → V
VP → V S
Pron → I
Pron → her
V → saw
N → duck
V → duck
            
```

Ç. Çöltekin, NR / University of Tübingen

Winter Semester 2020/21 6 / 13

# Example: bottom-up chart parsing

```

grammar
S → NP VP
NP → Pron N
NP → Pron
VP → V NP
VP → V
VP → V S
Pron → I
Pron → her
V → saw
N → duck
V → duck
            
```

Ç. Çöltekin, NR / University of Tübingen

Winter Semester 2020/21 6 / 11

# Example: bottom-up chart parsing

```

grammar
S → NP VP
NP → Pron N
NP → Pron
VP → V NP
VP → V
VP → V S
Pron → I
Pron → her
V → saw
N → duck
V → duck
            
```

Ç. Çöltekin, NR / University of Tübingen

Winter Semester 2020/21 6 / 13

# Example: bottom-up chart parsing

```

grammar
S → NP VP
NP → Pron N
NP → Pron
VP → V NP
VP → V
VP → V S
Pron → I
Pron → her
V → saw
N → duck
V → duck
            
```

Ç. Çöltekin, NR / University of Tübingen

Winter Semester 2020/21 6 / 11

# Example: bottom-up chart parsing

```

grammar
S → NP VP
NP → Pron N
NP → Pron
VP → V NP
VP → V
VP → V S
Pron → I
Pron → her
V → saw
N → duck
V → duck
            
```

Ç. Çöltekin, NR / University of Tübingen

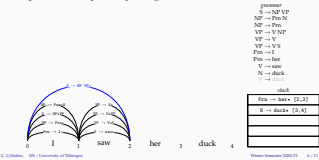
Winter Semester 2020/21 6 / 13

Example: bottom-up chart parsing



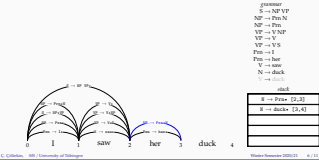
### Chart parsing

Example: bottom-up chart parsing



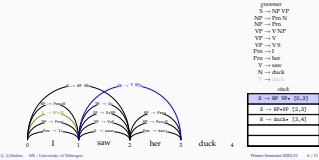
## Chart parsing

Example: bottom-up chart parsing



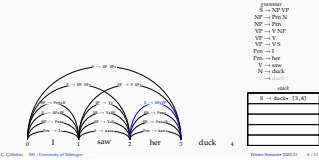
## Chart parsing

Example: bottom-up chart parsing



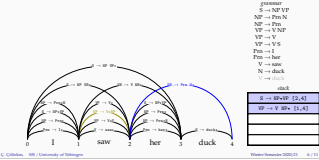
## Chart parsing

Example: bottom-up chart parsing

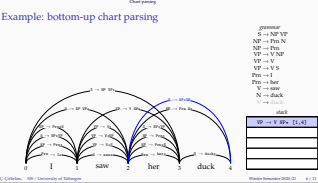


### Chart parsing

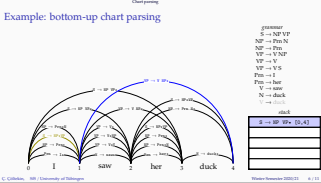
Example: bottom-up chart parsing



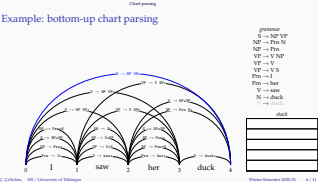
## Example: bottom-up chart parsing



## Example: bottom-up chart parsing



## Example: bottom-up chart parsing



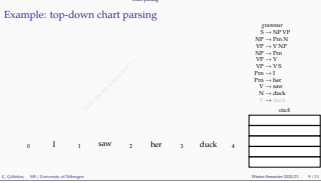
## Bottom-up chart parsing

- additional remarks
- The parser (as described) proceeds bottom up (left-corner)
  - It can process arbitrary CF grammars
  - Stack-based agenda is common, queue-based agenda is rarely used
  - An interesting alternative is so-called *head-corner* parsing: using a priority queue (e.g., processing 'heads' first) one can build the 'important' parts of the tree first
  - The time complexity is  $O(n^3)$
  - There are many variants, optimizations (based on, different inference rules, processing strategies)
- Winter Semester 2020/21 7 / 11

## Top-down chart parsing

- The basic algorithm is the same, but we specify
    - Inference rule (besides the 'fundamental rule'):
      - If the new edge has the form  $A \rightarrow \alpha \bullet \beta$  [ $i, j$ ], add  $B \rightarrow \gamma$  [ $j, j$ ] for each rule  $B \rightarrow \gamma$  in the grammar.
    - Initialization
      - Empty chart
      - Push  $\rightarrow st$  [ $0, n$ ] into the stack
      - Push all productions for the terminal symbols to the stack (or to the chart, as there is nothing to predict for these productions)
    - Typically we use a stack as an agenda
- Winter Semester 2020/21 8 / 11

## Example: top-down chart parsing



## Top-down chart parsing

- additional remarks
- The parser (as described) is purely top-down
  - In practice, it is common to use 'lookup'
  - Stack-based agenda is common, queue-based agenda is rarely used
  - The time complexity is  $O(n^3)$
  - There are many variants, optimizations (based on, different inference rules, processing strategies)
- Winter Semester 2020/21 10 / 11

## Summary

- Chart parsing is a general framework for constructing a variety of parsers
  - It shares many similarities with the CKY and Earley
- Next:
- Deterministic parsing (maybe after the break)
- Winter Semester 2020/21 11 / 11

## Acknowledgments, references, additional reading material

Miriam Dick and Cord J.E. Smeets (2007). *Parsing Techniques: A Practical Guide*. second. *Monographs in Computer Science*. The best address is available at [http://algorithms.wtf/Books/PTTC2007\\_text\\_data/acknowledging.pdf](http://algorithms.wtf/Books/PTTC2007_text_data/acknowledging.pdf). Springer New York, isbn: 9780307076555.

Winter Semester 2020/21 A.1

Winter Semester 2020/21 A.2

Example: top-down chart parsing

grammar  
S → NP VP  
NP → Pm N  
VP → V NP  
VP → V  
VP → V S  
Pm → I  
Pm → her  
V → saw  
N → duck  
V → is/are

0        I        1        saw        2        her        3        duck        4