Chart Parsing

Parsing
ISCL-BA-06

Çağrı Çöltekin
ccoltekin@sfs.uni-tuebingen.de

University of Tübingen
Seminar für Sprachwissenschaft

Winter Semester 2020/21

version: e00e075 @2020-12-22

Parsing so far

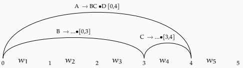
- We can formulate parsing as
  - Top-down: begin with the start symbol, try to *produce* the input string to be parsed
  - Bottom up: begin with the input, and try to *reduce* it to the start symbol
- For both options, we have seen examples of chart parser
- Parsing can also be *directional* or *non-directional*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

The overall idea

- We adopt Early-like chart entries of the form: $X \rightarrow \alpha \bullet \beta$ [i,j] where,
  - i and j are indexes starting from 0 (0 indicating beginning of the input string)
  - The chart entry indicates α is found between i and j, we are looking for a β 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 $S \rightarrow \ldots\bullet$ [0,n]



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 $A \rightarrow \alpha\bullet$ and an active item of the form $B \rightarrow \beta \bullet A\gamma$ add item $B \rightarrow \beta A \bullet \gamma$
- 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

The sketch of a chart parsing algorithm

```
Initialize A (agenda) and C (chart)
repeat
    i ← next(A)
    if i ∈ C then
        discard i
    else
        apply all inference rules to i
        place new items in A
        place the item in C
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

Bottom-up chart parsing

- Single additional inference rule:
  - If a new item has the form $A \rightarrow \alpha\bullet$, add $B \rightarrow A\bullet\beta$ 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 POS tag in CL)
  - if there are ϵ rules, add $P \rightarrow \bullet$[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

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

stack

Prn → I• [0,1]
V → saw• [1,2]
Prn → her• [2,3]
N → duck• [3,4]

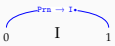
0 I 1 saw 2 her 3 duck 4

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

stack

NP → Prn• [0,1]
NP → Prn•N [0,1]
V → saw• [1,2]
Prn → her• [2,3]
N → duck• [3,4]



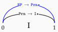
0 I 1 saw 2 her 3 duck 4

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

stack

S → NP•VP [0,1]
NP → Prn•N [0,1]
V → saw• [1,2]
Prn → her• [2,3]
N → duck• [3,4]



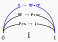
0 I 1 saw 2 her 3 duck 4

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

stack

NP → Prn•N [0,1]
V → saw• [1,2]
Prn → her• [2,3]
N → duck• [3,4]



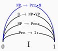
0 I 1 saw 2 her 3 duck 4

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

stack

V → saw• [1,2]
Prn → her• [2,3]
N → duck• [3,4]



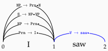
0 I 1 saw 2 her 3 duck 4

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

stack

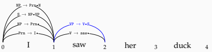
VP → V•S [1,2]
VP → V•NP [1,2]
VP → V• [1,2]
Prn → her• [2,3]
N → duck• [3,4]



0 I 1 saw 2 her 3 duck 4

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

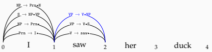


stack

VP → V•NP [1,2]
VP → V• [1,2]
Prn → her• [2,3]
N → duck• [3,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

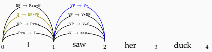


stack

VP → V• [1,2]
Prn → her• [2,3]
N → duck• [3,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

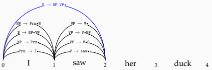


stack

S → NP VP• [0,2]
Prn → her• [2,3]
N → duck• [3,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

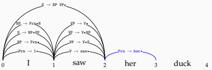


stack

Prn → her• [2,3]
N → duck• [3,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

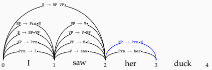


stack

NP → Prn•N [2,3]
N → Prn• [2,3]
N → duck• [3,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

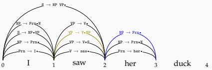


stack

N → Prn• [2,3]
N → duck• [3,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

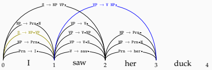


stack

VP → V NP• [1,3]
S → NP•VP [2,3]
N → duck• [3,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

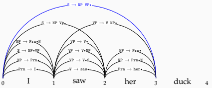


stack

S → NP VP• [0,3]
S → NP•VP [2,3]
N → duck• [3,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

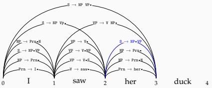


stack

S → NP•VP [2,3]
N → duck• [3,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

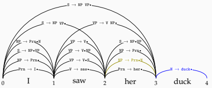


stack

N → duck• [3,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

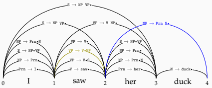


stack

NP → Prn N• [2,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

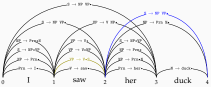


stack

S → NP VP• [2,4]
VP → V NP• [1,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

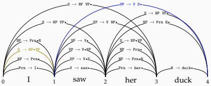


stack

VP → V S• [1,4]
VP → V NP• [1,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

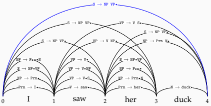


stack

S → NP VP• [0,4]
VP → V NP• [1,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

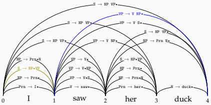


stack

VP → V NP• [1,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

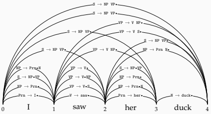


stack

S → NP VP• [0,4]

Example: bottom-up chart parsing

grammar
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck



stack

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)

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 → α•Bβ [i,j]`, add `B → •γ [j,j]` for each rule `B → γ` in the grammar.
  - Initialization
    - Empty chart
    - Push `→ •S [0,0]` 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

Example: top-down chart parsing

grammar
S → NP VP
NP → Prn N
VP → V NP
NP → Prn
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

Left as an exercise :-)

0 I 1 saw 2 her 3 duck 4

stack

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, que-based agenda is rarely used
- The time complexity is $O(n^3)$
- There are many variants, optimizations (based on, different inference rules, processing strategies)

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)

Acknowledgments, references, additional reading material

Grune, Dick and Ceriel J.H. Jacobs (2007). *Parsing Techniques: A Practical Guide.* second. Monographs in Computer Science. The first edition is available at `http://dickgrune.com/Books/PTAPG_1st_Edition/BookBody.pdf`. Springer New York. ISBN: 9780387689548.

blank

blank

blank

blank

blank

blank

Example: top-down chart parsing

grammar

S → NP VP
NP → Prn N
VP → V NP
NP → Prn
VP → V
VP → V S
Prn → I
Prn → her
V → saw
N → duck
V → duck

$_0$ I $_1$ saw $_2$ her $_3$ duck $_4$