UPPSALA
UNIVERSITET

Introduction

Syntactic parsing (5LN713)

2023-01-16

Sara Stymne
Department of Linguistics and Philology

Partly based on slides from Marco Kuhlmann

Today

- Introduction to syntactic analysis
- Course information
- Exercises

What is syntax?

- Syntax addresses the question of how sentences are constructed in particular languages.
- The English (and Swedish) word *syntax* comes from the Ancient Greek word *sýntaxis* ‘arrangement’.

What is syntax not?

Syntax does not answer questions about …

… how speech is articulated and perceived (phonetics, phonology)

… how words are formed (morphology)

… how utterances are interpreted in context (semantics, pragmatics)

simplified

Why should you care about syntax?

- Syntax describes the distinction between well-formed and ill-formed sentences.
- Syntactic structure can serve as the basis for semantic interpretation and can be used for
  - Machine translation
  - Information extraction and retrieval
  - Question answering
  - …

Why should you care about syntax?

- Syntactic structure can be useful for analysing large text materials
  - Research in digital humanities
  - Economic analysis

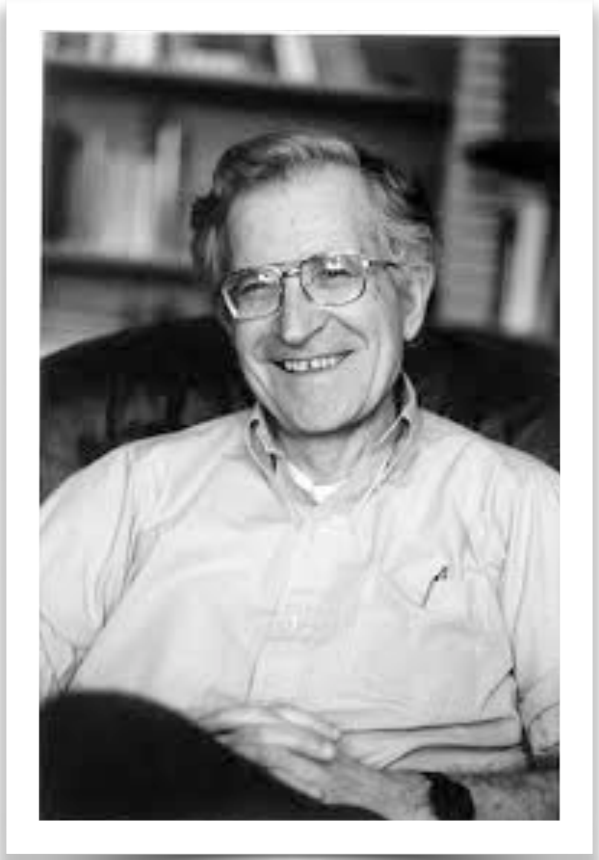
Parsing

The automatic analysis of a sentence with respect to its syntactic structure.

Theoretical frameworks

- Generative syntax
  Noam Chomsky (1928–)
- Categorial syntax
  Kazimierz Ajdukiewicz (1890–1963)
- Dependency syntax
  Lucien Tesnière (1893–1954)

Theoretical frameworks

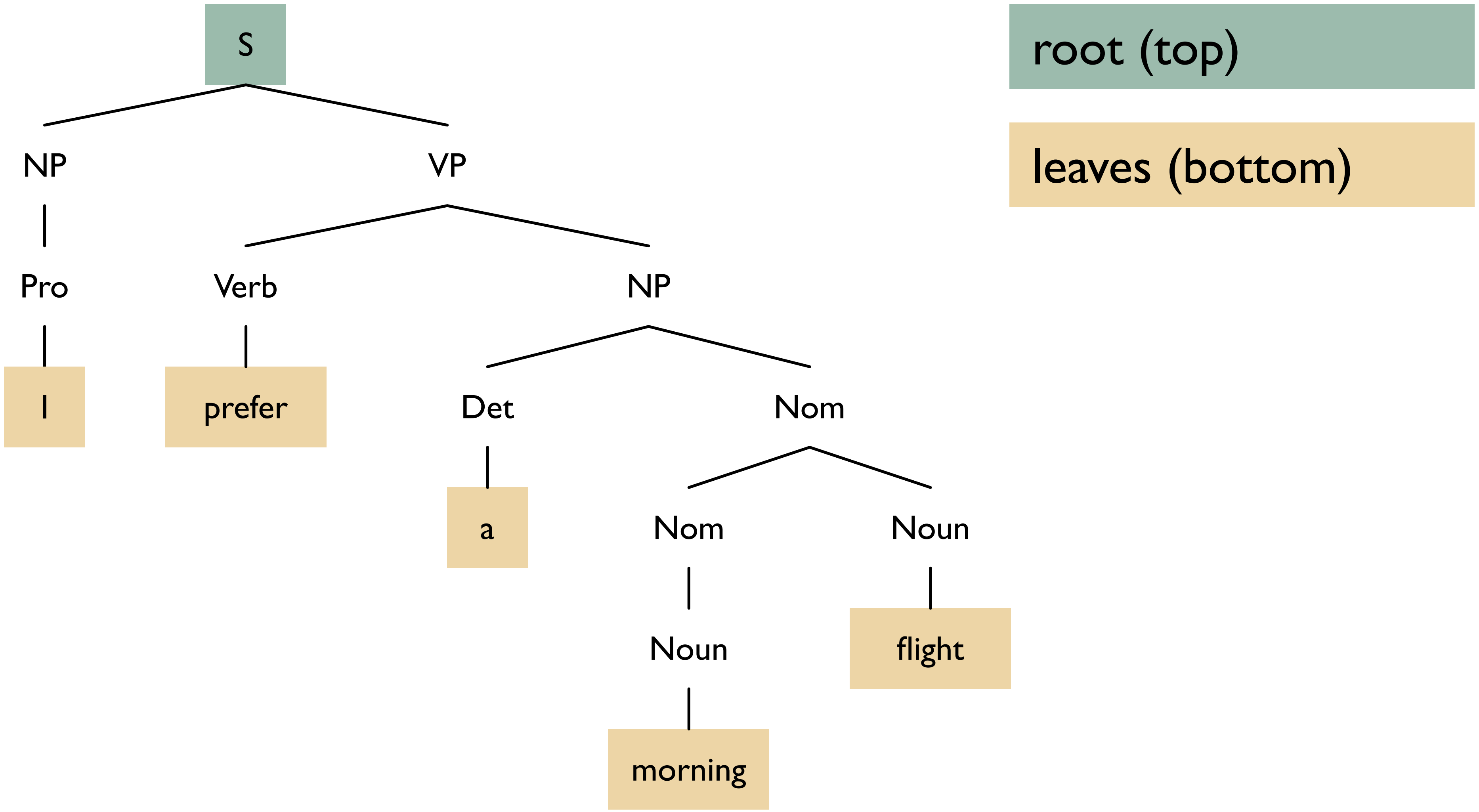


Chomsky

Ajdukiewicz

Tesnière

Phrase structure trees



S

NP

VP

Pro

Verb

NP

I

prefer

Det

Nom

a

Nom

Noun

Noun

flight

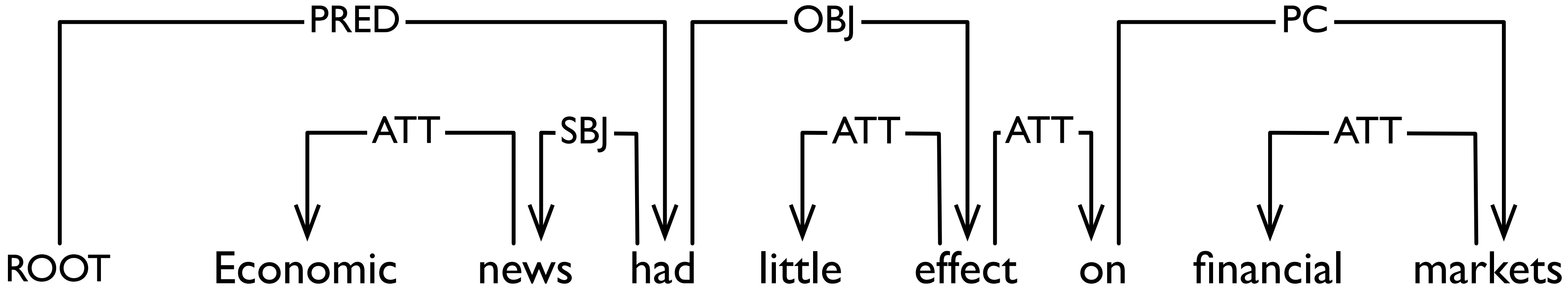
morning

root (top)

leaves (bottom)

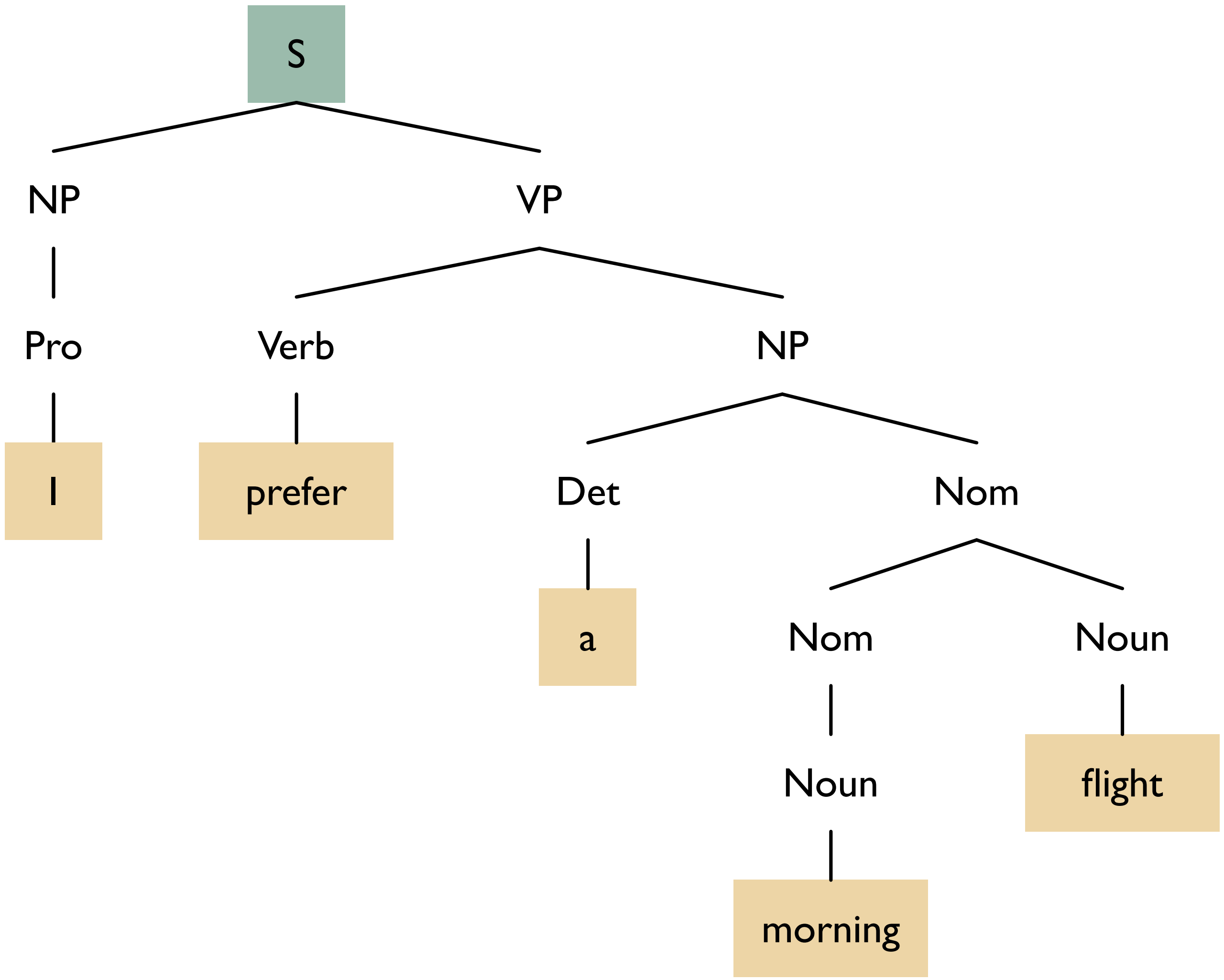
UPPSALA
UNIVERSITET

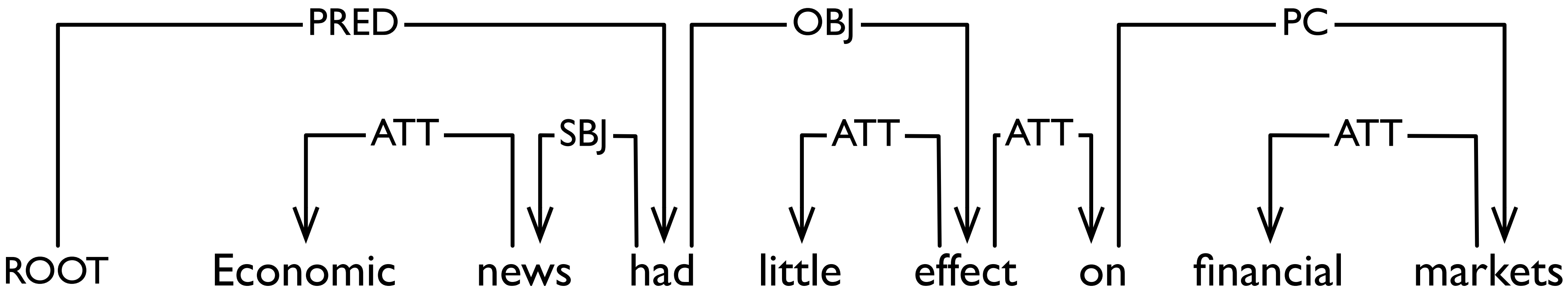
Dependency trees



UPPSALA
UNIVERSITET

Phrase structure vs dependency trees



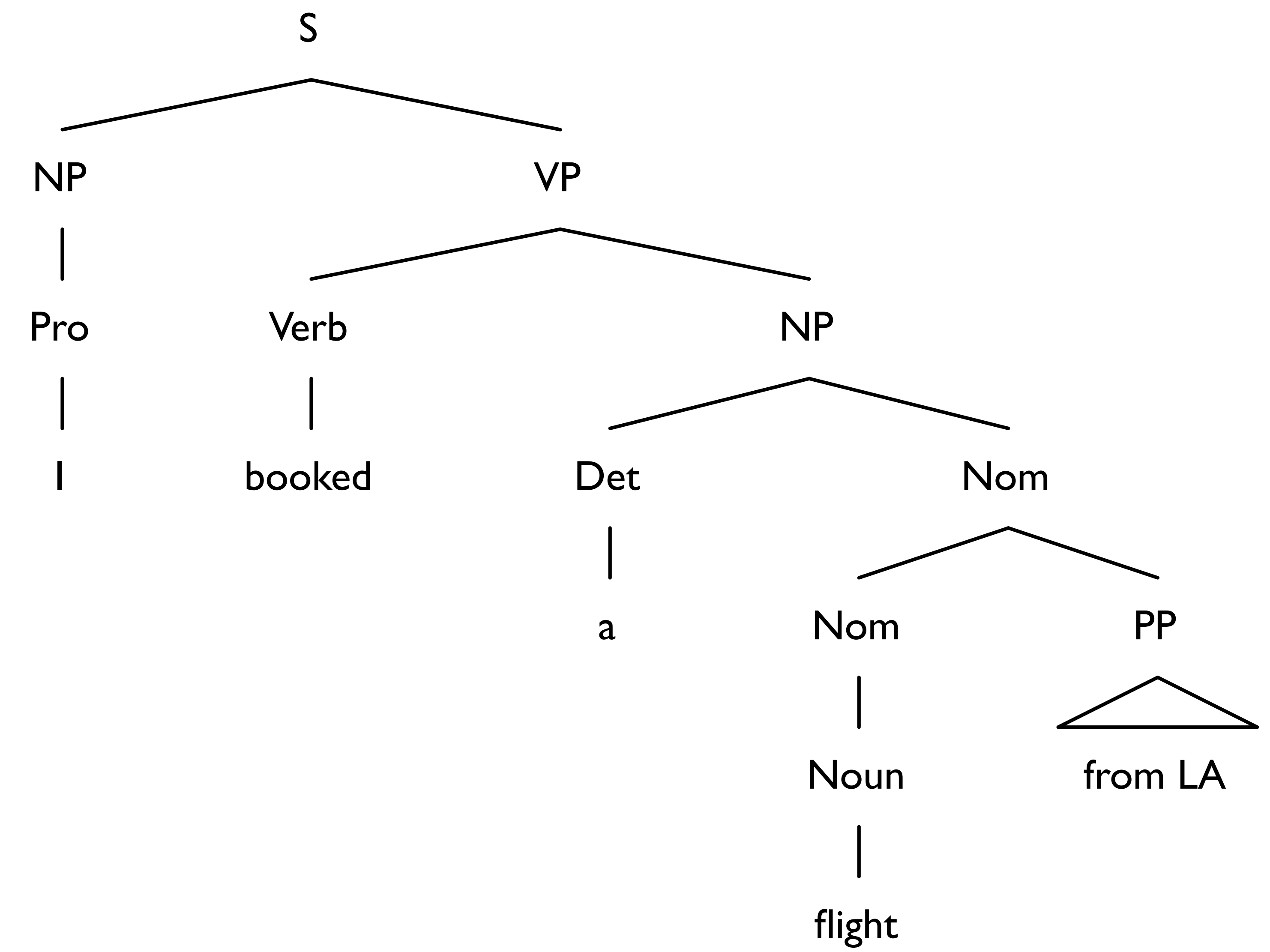


Ambiguity

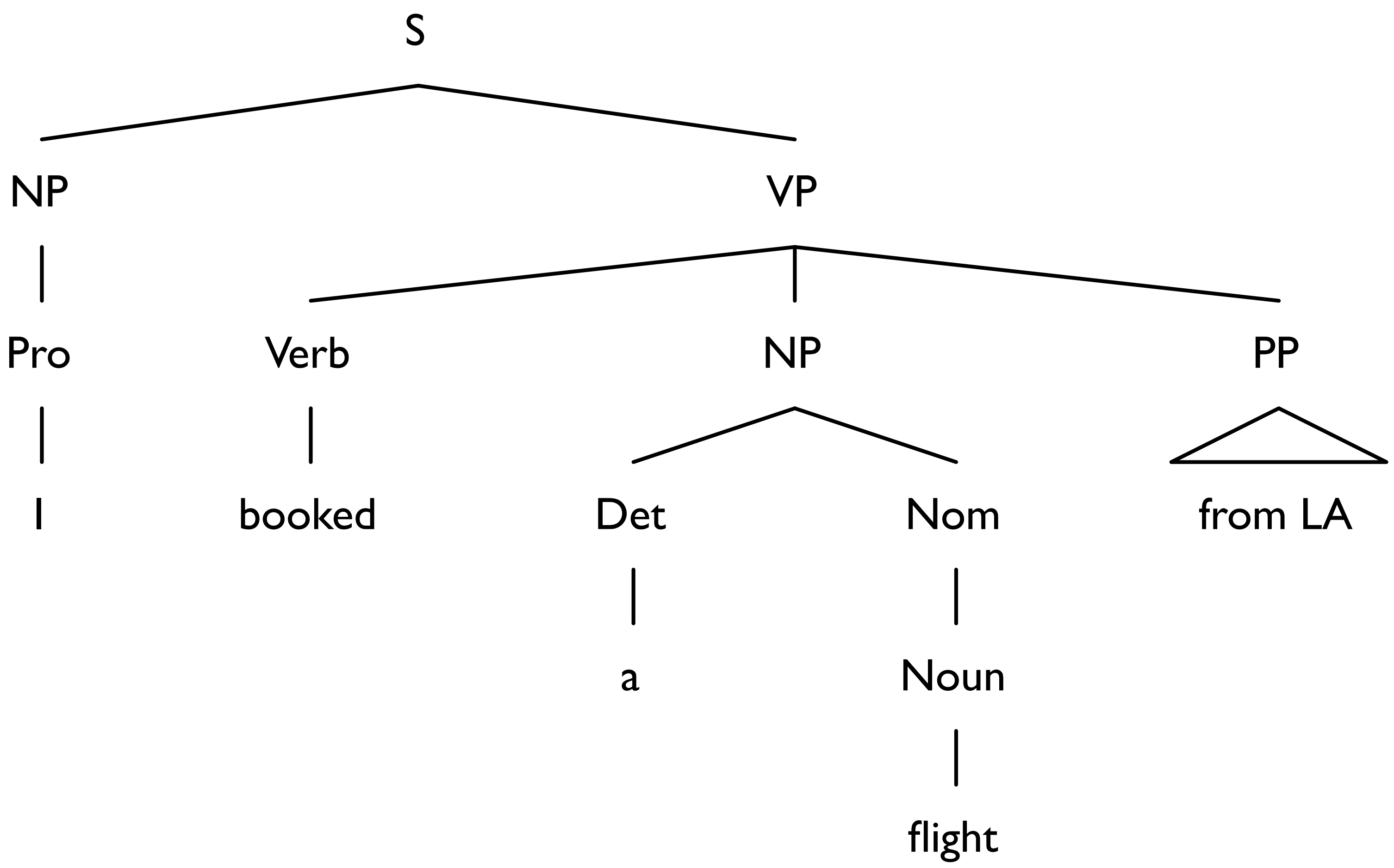
I booked a flight from LA.

- This sentence is ambiguous. In what way?
- What should happen if we parse the sentence?

Ambiguity



Ambiguity



Interesting questions

- Is there any parse tree at all?
  - *Recognition*
- What is the best parse tree?
  - *Parsing*

Parsing as search

- Parsing as search:
  Search through all possible parse trees for a given sentence.
- In order to search through all parse trees we have to 'build' them.

Top–down and bottom–up

- top–down
  - only build trees that are rooted at *S*
  - may produce trees that do not match the input
- bottom–up
  - only build trees that match the input
  - may produce trees that are not rooted at *S*

Dynamic programming (DP)

- Divide and conquer:
  In order to solve a problem, split it into subproblems, solve each subproblem, and combine the solutions.
- Dynamic programming (DP) (bottom up):
  Solve each subproblem only once and save the solution in order to use it as a partial solution in a larger subproblem.
- Memoisation (top down):
  Solve only the necessary subproblems and store their solutions for reuse in solving other subproblems.

Example: fibonacci numbers

Naive implementation

```
def fib(n):
    if n <= 1:
        return n
    else:
        return fib(n-1) + fib(n-2)
```

Time complexity: $O(2^n)$

Example: fibonacci numbers

Memoization (top down)

```
fibC = {0:0, 1:1}
def fibMem(n):
    if n <= 1:
        return n
    if not n in fibC:
        fibC[n] = fibMem(n-1) + fibMem(n-2)
    return fibC[n]
```

Time complexity: O(n)

Example: fibonacci numbers

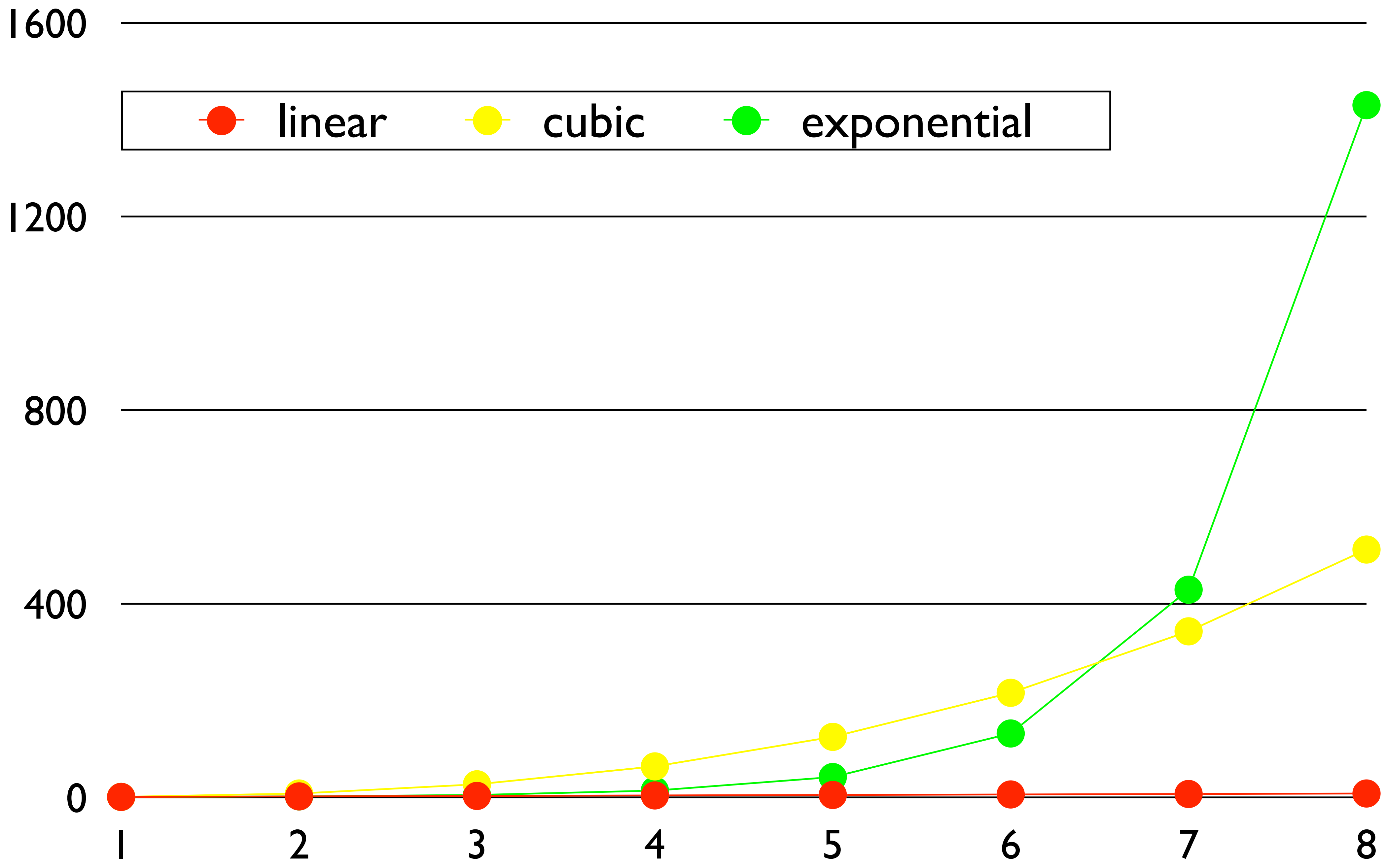
Dynamic programming (bottom up)

```
def fib_dp(n):
    fibV = [0,1]
    for i in range(2, n+1):
        fibV.append(fib[i-1] + fibV[i-2])
    return fibV[n]
```

Time complexity: O(n)

UPPSALA
UNIVERSITET

How many trees are there?



Complexity

- Using DP we can (sometimes) search through all parsetrees in polynomial time.
- That is much better than to spend exponential time!
- But it may still be too expensive!
  In these cases one can use an approximative method such as greedy search or beam search.
  - Often possible in linear time

Course information

Intended learning outcomes 5LN713

At the end of the course, you should be able to

- explain the standard models and algorithms used in phrase structure and dependency parsing;
- implement and evaluate some of these techniques;
- critically evaluate scientific publications in the field of syntactic parsing,
- design, evaluate, or theoretically analyse the syntactic component of an NLP system

Examination 5LN713

- Examination is continuous and distributed over three graded assignments, two literature seminars, and a graded project
- Two assignments are programming tasks where you implement (parts of) parsers.
- Literature review assignment
- Two literature seminars

Practical assignments

- Assignment 1: PCFG
  - Implement conversion of treebank to CNF
  - Implement CKY algorithm
- Assignment 3: Dependency parsing
  - Implement an oracle for transition-based dependency parsing

Literature review

- Pick two research articles about parsing
- Can be from journals, conferences or workshops
- The main topic of the articles should be parsing, and they should be concerned with algorithms (i.e. not focusing on applying parsing to other tasks, evaluation, et.c.)
- Write a 2-page report: summarize, analyse and critically discuss

Literature seminars

- Read one given article for each seminar
- Prepare according to the instructions on the homepage
- Everyone is expected to be able to discuss the article and the questions about it
  - It should be clear that you have read and analyzed the article, but it is perfectly fine if there are parts that you find difficult and do not fully understand
- The seminars are obligatory
  - If you miss a seminar or are unprepared, you will have to hand in a written report.

Project

- Can be done individually or in pairs:
    - To be self-organized by you!
- Suggestions for topics/themes will be on the web page
- Project activities:
    - Proposal: February 27
    - Discussion seminar: March 22
    - Report: March 24

Learning outcomes and examination

- explain the standard models and algorithms used in phrase structure and dependency parsing; all assignments and seminars
- implement and evaluate some of these techniques; assignments 1, 3
- critically evaluate scientific publications in the field of syntactic parsing, assignment 2, seminars
- design, evaluate, or theoretically analyze the syntactic component of an NLP system project

UPPSALA
UNIVERSITET

Grading 5LN713

- The assignments and project are graded with G and VG
- G on the seminars if present, prepared and active. The seminars are obligatory, and not graded!
- To achieve G on the course:
  - G on all assignments, seminars and project
- To achieve VG on the course:
  - VG on the three assignments or
  - VG on project and at least one assignment

Teaching

- Lectures
  - Mainly:
    - Distributed as recordings
    - Followed by summary+exercise on Campus (+Zoom)
  - In a few cases live
- 2 seminars
  - Tentatively on Campus
- Assignment supervision on Campus 4 times, plus on request

Lectures

- Lectures and course books cover basic parsing algorithms in detail
- They touch on more advanced material, but you will need to read up on that independently

Lecture organization

- Watch recorded lectures (slides+voice) on your own
- Read relevant course literature
- Work on given small exercise on your own
- This is followed by a summary session
  - Repetition of the most important concepts
  - Discussion of exercise + questions from recordings
  - Questions by students

Course information

- Web page:
  - Course information
  - Assignments and other instructions
  - Annotated schedule
- Studium:
  - Zoom links
  - Recorded lectures and lecture materials
  - Hand in assignments

Course workload 5LN713

- **7.5 hp means about 200 hours work:**
- ~ 40 h lectures (including preparation)
- 2 h seminars
- 158 h work on your own
  - ~ 80 h assignment work (including reading)
  - ~ 10 h seminar preparation
  - ~ 68 h project work

UPPSALA
UNIVERSITET

Deadlines

Assignment	Deadline	Backup
1: PCFG	Feb 13	April 3
2: Lit review	March 6	April 3
3: Dependency	March 13	April 3
Project proposal	Feb 27	March 3
Project report	March 24	April 17
Missing seminar report	March 24	April 17

Seminar	Date
1	February 8
2	March 2
Project seminar	March 22

Reading: course books

- Daniel Jurafsky and James H. Martin.
  Speech and Language Processing. 3rd edition.
  2023. Available online as pdf.
  Chapters 17-18.

- Sandra Kübler, Ryan McDonald,
  and Joakim Nivre. Dependency Parsing.
  Morgan and Claypool, 2009. Available online
  through UU.
  Chapters 1-4, 6.

UPPSALA
UNIVERSITET

Reading: articles (tentatively)

- Seminar 1
  - Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, Noah A. Smith. Recurrent Neural Network Grammars. NAACL 2016.
- Seminar 2
  - Eliyahu Kiperwasser and Yoav Goldberg. Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. TACL. Volume 4, 2016

Reading: additional material

- Lecture notes by Joakim Nivre – in Studium
- Additional research articles
    - Especially for project and assignment 2

Evaluation from previous years

- 2022: Overall score: 4/5
- Strengths (from recent years):
  - Implementation assignments were useful (but hard)
  - Good to combine the implementation of basic algorithms with discussions of more advanced topics
  - Freedom to choose a project
  - Literature review
- Weaknesses:
  - Assignment I is a bit difficult
    - We have added more supervision in recent years
  - The course might be more rewarding if taught after ML.
    - This will change in the future. For now, I will take that into account, and adjust both explanations and expectations.

Recorded lectures

- Available in Studium (with automatic subtitles)
- Until you get access to Studium, you can find the first block of recorded lectures (without subititles) here:
    - https://www.youtube.com/playlist?list=PLH4LBIvRWr95-h6-g8R4P3hUF1wZK3sdh
- From 2020, so a few comments may not be relevant (e.g. referring to the advanced programming course as finished). Time complexity, for instance, will be discussed in more detail during the classroom lectures.

Work until Monday lecture

- Read J&M 17.1–17.5 (introduction)
- Read J&M 17.6 (CKY)
- Watch recorded lectures about CKY
- Read description of assignment 1: CKY
- Work on exercises (in Studium)

Exercise

- Try to come up with parse trees for all possible interpretations of the below example sentence:
  - Phrase-structure trees
  - Dependency trees

- ”Time flies like an arrow”