UPPSALA
UNIVERSITET

Introduction

Syntactic parsing (5LN713)

2024-01-15

Sara Stymne
Department of Linguistics and Philology

Partly based on slides from Marco Kuhlmann

UPPSALA
UNIVERSITET

Today

- Introduction to syntactic analysis
- Course information
- Exercises

UPPSALA
UNIVERSITET

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’.

UPPSALA UNIVERSITET

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

UPPSALA UNIVERSITET

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
  - ...

UPPSALA UNIVERSITET

Why should you care about syntax?

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

UPPSALA
UNIVERSITET

Parsing

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

UPPSALA UNIVERSITET

Theoretical frameworks

- Generative syntax

  Noam Chomsky (1928–)

- Categorial syntax

  Kazimierz Ajdukiewicz (1890–1963)

- Dependency syntax

  Lucien Tesnière (1893–1954)

UPPSALA
UNIVERSITET

Theoretical frameworks

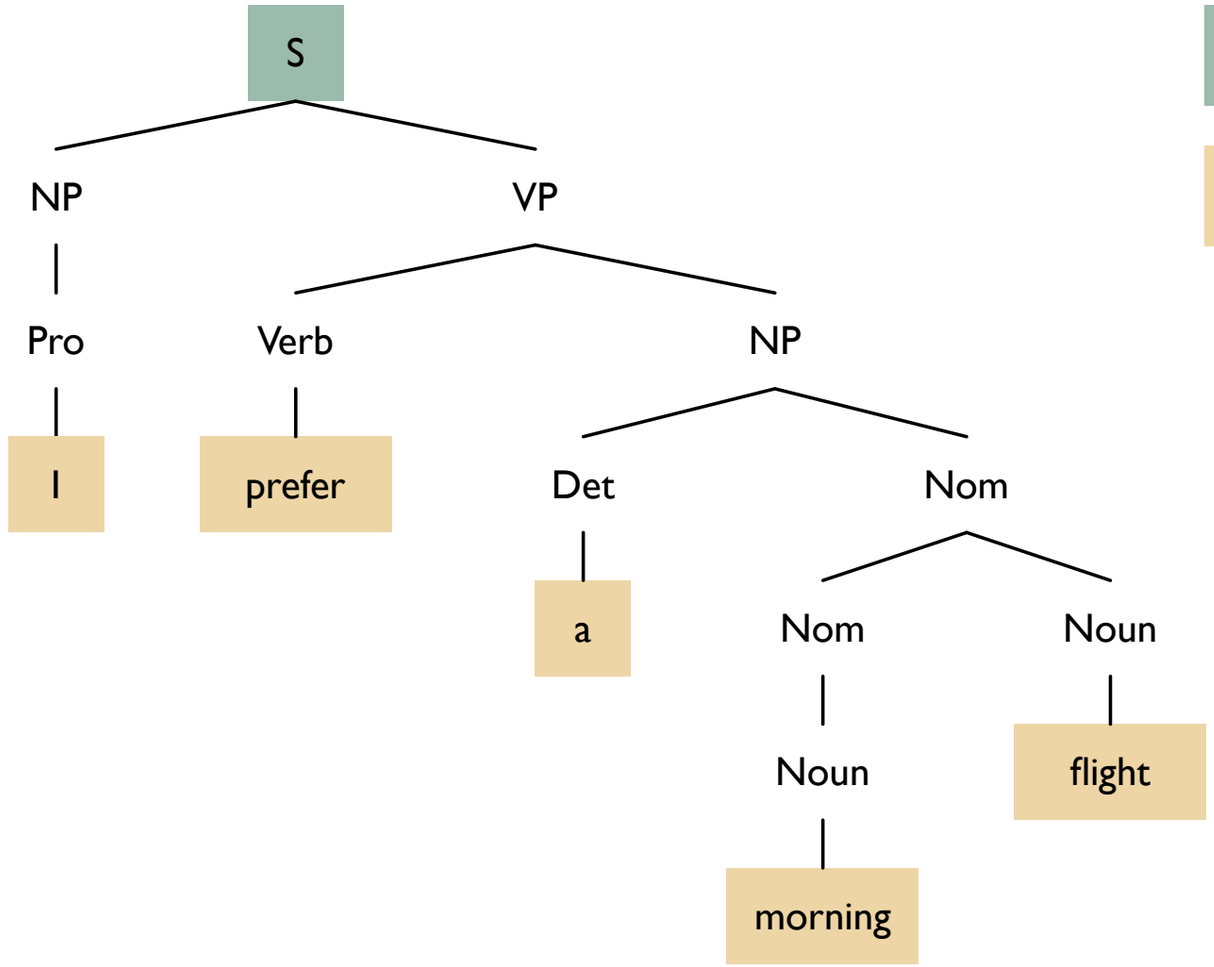
Chomsky

Ajdukiewicz

Tesnière

UPPSALA UNIVERSITET

Phrase structure trees

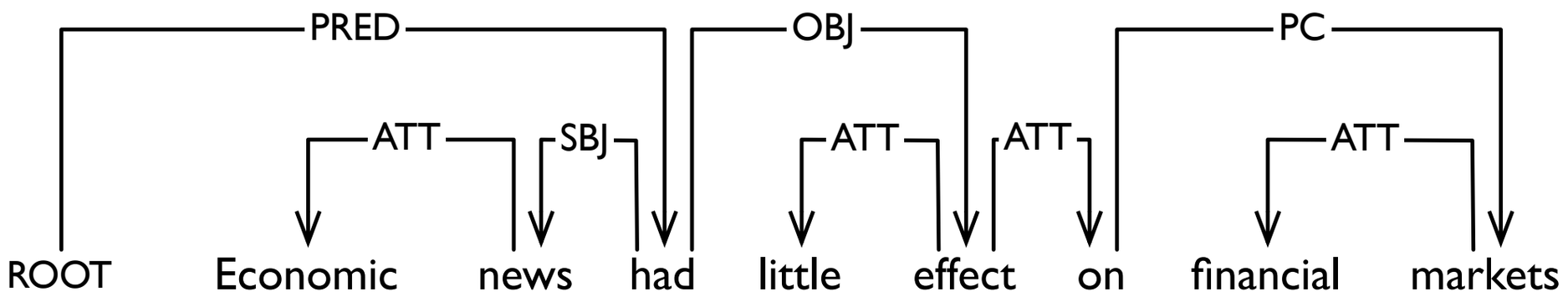


root (top)

leaves (bottom)

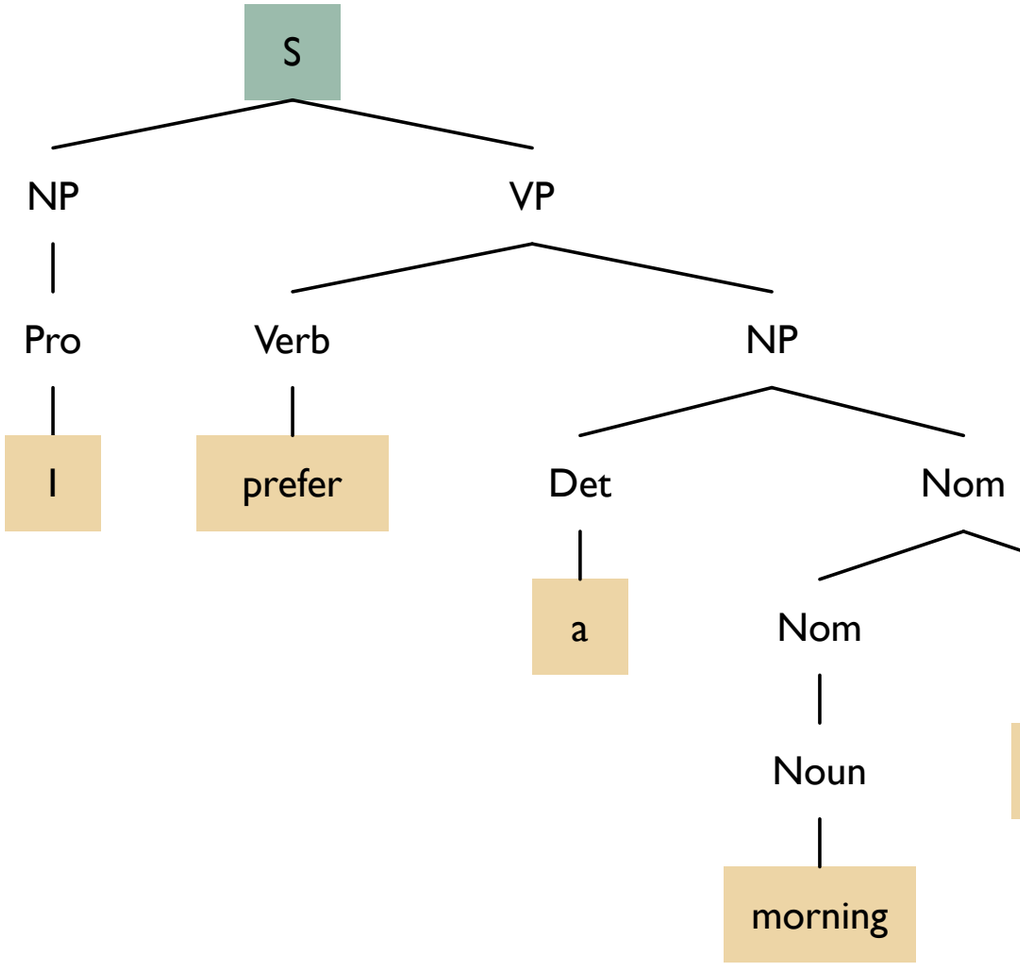
UPPSALA
UNIVERSITET

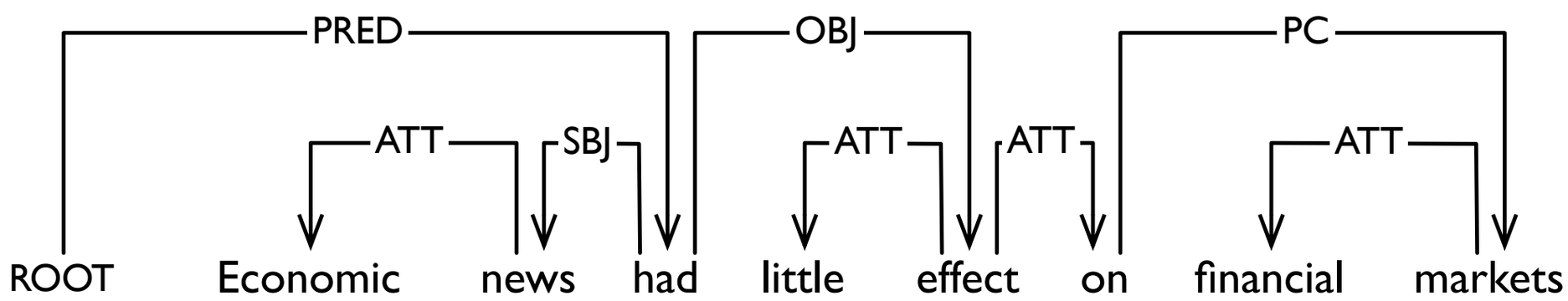
Dependency trees



UPPSALA UNIVERSITET

Phrase structure vs dependency trees





UPPSALA UNIVERSITET

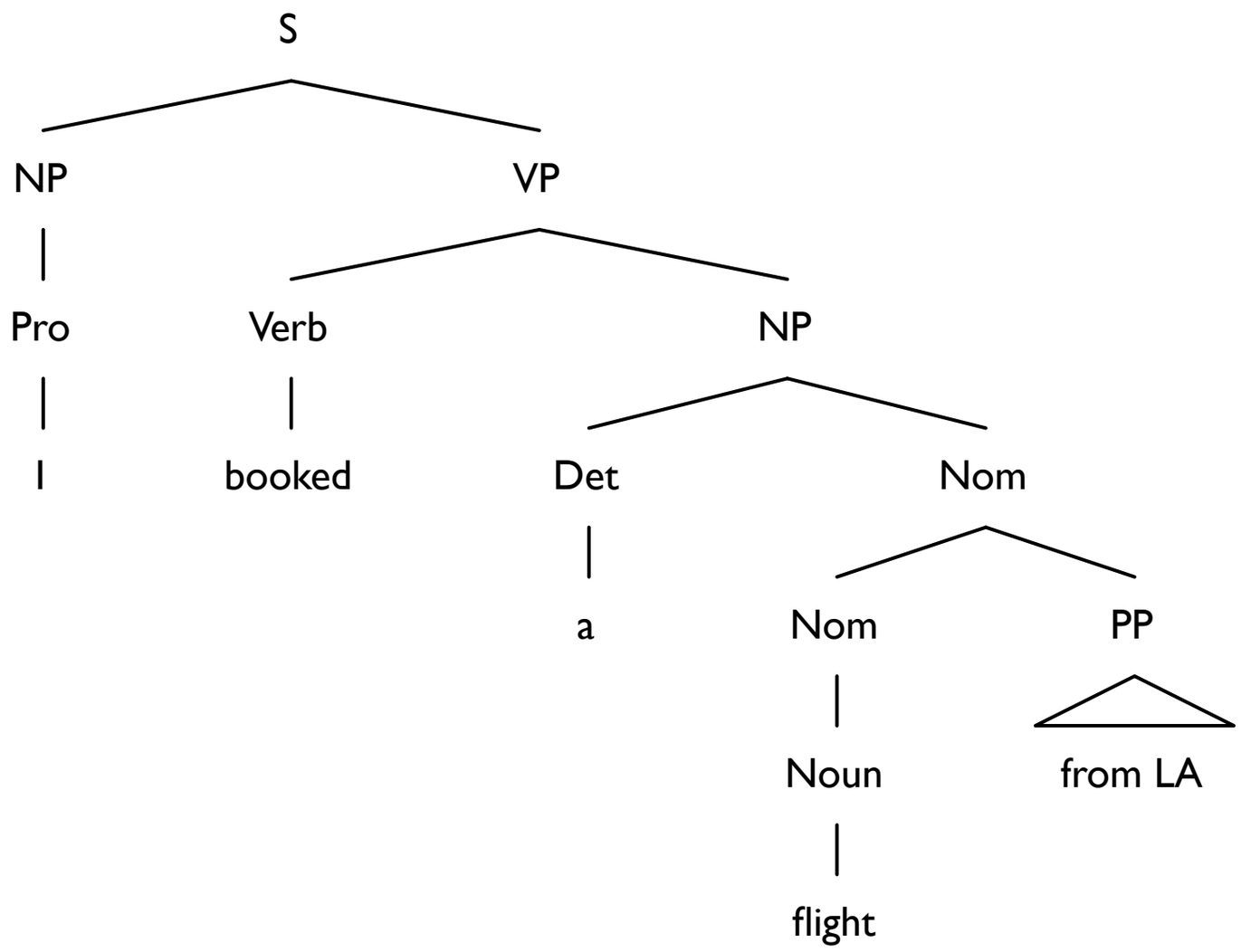
Ambiguity

I booked a flight from LA.

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

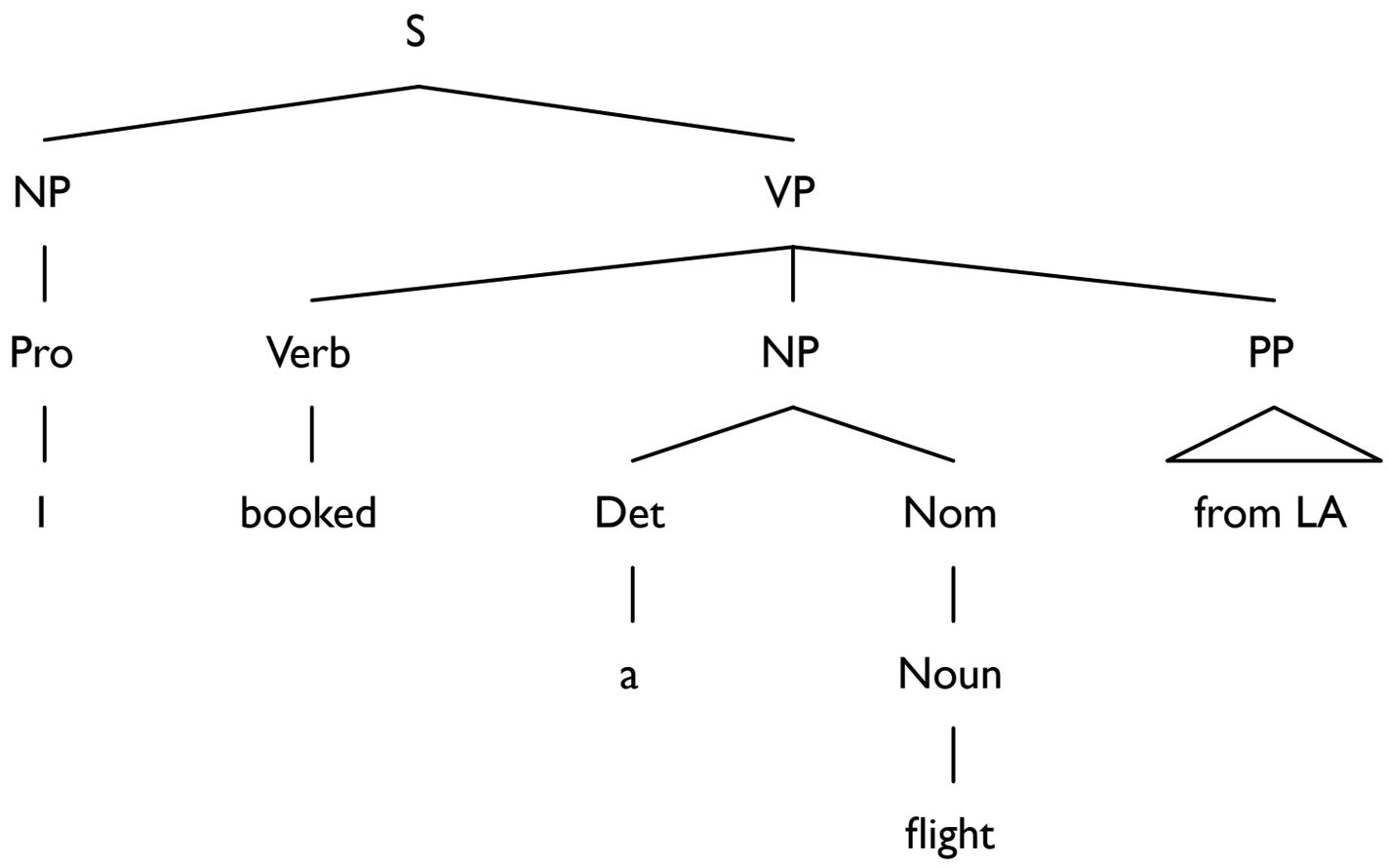
UPPSALA UNIVERSITET

Ambiguity



UPPSALA UNIVERSITET

Ambiguity



UPPSALA UNIVERSITET

Interesting questions

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

UPPSALA UNIVERSITET

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.

UPPSALA
UNIVERSITET

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*

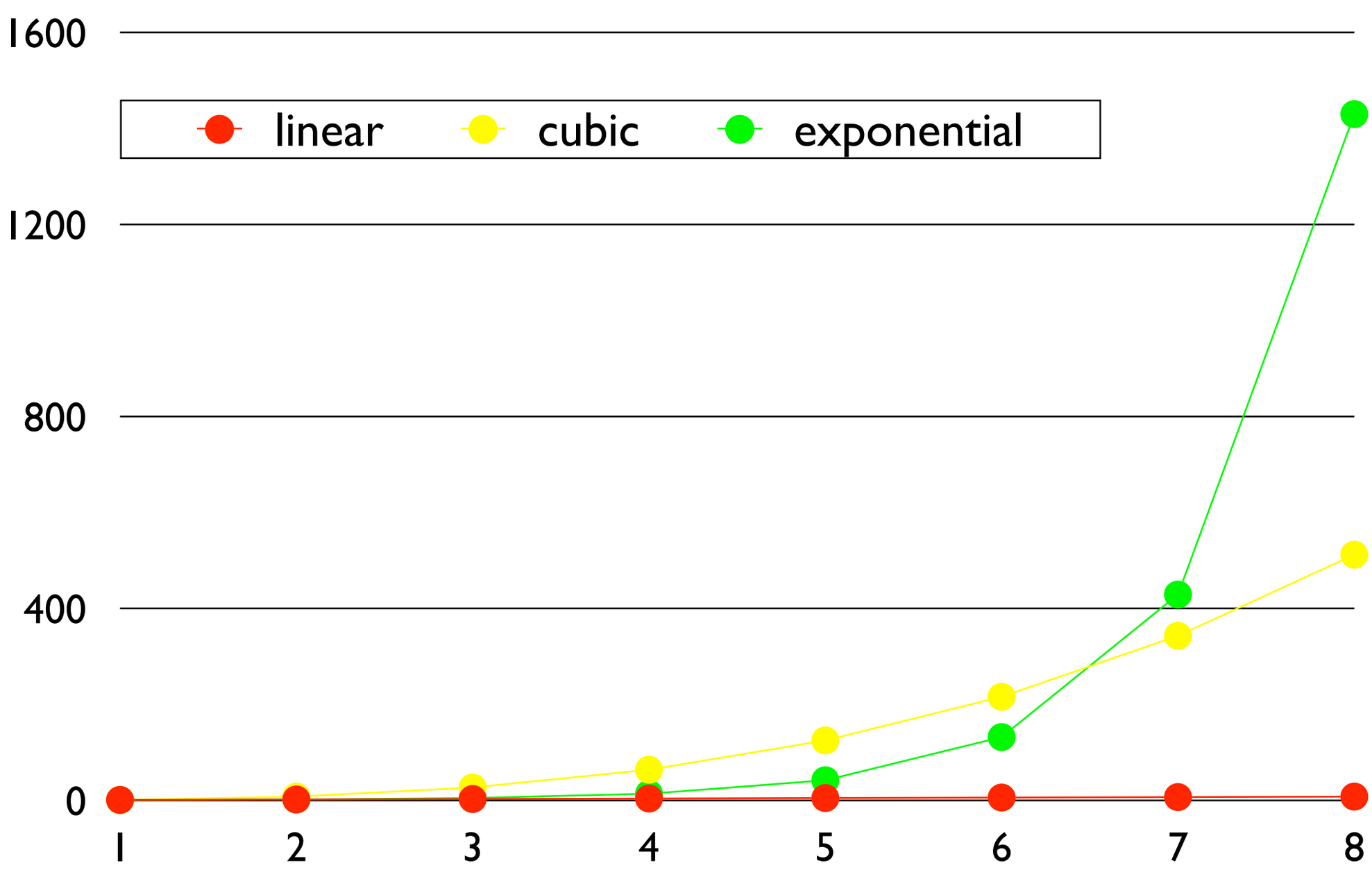
UPPSALA UNIVERSITET

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.

UPPSALA UNIVERSITET

How many trees are there?



UPPSALA
UNIVERSITET

Example: fibonacci numbers

Number series:

0 1 1 2 3 5 8 13 21 …

Formula:

F(n) = F(n-1) + F(n-2)

Base cases:

F(0) = 0, F(1) = 1

For n>1

UPPSALA UNIVERSITET

Example: fibonacci numbers

Naive implementation

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

Time complexity: $O(2^n)$

UPPSALA UNIVERSITET

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)

UPPSALA UNIVERSITET

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

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

UPPSALA
UNIVERSITET

Course information

UPPSALA UNIVERSITET

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

UPPSALA UNIVERSITET

Examination 5LN713

- Examination is continuous and distributed over three graded assignments, two literature seminars, and a graded project
  - 2 assignments are programming tasks where you implement (parts of) parsers.
  - 1 assignment where you run and evaluate a parser in a cross-lingual setting
  - Two literature seminars

UPPSALA UNIVERSITET

Assignments

- Assignment 1: PCFG
  - Implement conversion of treebank to CNF
  - Implement CKY algorithm
- Assignment 2: Dependency parsing
  - Implement an oracle for transition-based dependency parsing
  - Discuss issues with the implementation

UPPSALA UNIVERSITET

Assignments

- Assignment 3: Cross-lingual parsing
  - Use an existing dependency parser (UUparser)
  - Compare different transfer languages for a (simulated) low-resource language
  - Try different evaluation methods

UPPSALA UNIVERSITET

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 and discuss it briefly with the teacher

UPPSALA UNIVERSITET

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 26
  - Discussion seminar: March 20 (may be changed)
  - Report: March 22

UPPSALA UNIVERSITET

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, 2, (project)
- critically evaluate scientific publications in the field of syntactic parsing, seminars, project
- design, evaluate, or theoretically analyze the syntactic component of an NLP system project, assignment 3

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 the project and at least one assignment

UPPSALA UNIVERSITET

Teaching

- Lectures
  - Mainly:
    - Distributed as recordings
      - Followed by summary+exercise on Campus (+Zoom)
  - In a few cases standard lectures
- 2 seminars
  - On Campus in smaller groups
- Assignment and project supervision on Campus, plus on request by email

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 exercises 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

Lecture recordings

- From 2020
  - Some 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
- Note that the order of lectures has also changed from previous years
  - Some references in the recordings may not be accurate, due to this
  - Please ask about anything that you find unclear!
  - Most obvious for the first lecture on dependency parsing

Course information

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

UPPSALA UNIVERSITET

Course workload 5LN713

- **7.5 hp means about 200 hours work:**
- ~ 40 h lectures (including preparation)
- 2 h seminars
- 13 h scheduled supervision
- 145 h work on your own
  - ~ 70 h assignment work (including reading)
  - ~ 10 h seminar preparation
  - ~ 65 h project work

UPPSALA UNIVERSITET

Deadlines

Assignment	Deadline	Backup
1: PCFG	Feb 9	April 1
2: Dependency	Feb 22	April 1
3: Cross-lingual	March 4	April 1
Project proposal	Feb 26	March 3
Project report	March 22	April 15
Missing seminar report	Feb 14/March 11	April 15

Seminar	Date
1	February 7
2	March 4
Project seminar	March 20

UPPSALA UNIVERSITET

Course status

- This course is given for the last time in 2024
- There will be three opportunities for reexamination after the end of the course, one per term, autumn 24, spring 25 and autumn 25.
  - Dates will be distributed on the course web page.
- There may have to be some changes to the schedule, due to traveling, possibly affecting the final seminar

UPPSALA
UNIVERSITET

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

- 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

UPPSALA UNIVERSITET

Reading: additional material

- Lecture notes by Joakim Nivre – in Studium
- Additional research articles
  - Especially for project and assignment 3

UPPSALA UNIVERSITET

Evaluation from previous years

- 2023: Overall score: 4.8/5
- 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, and to work in pairs
  - Good structure, assignments, and teacher
- Weaknesses:
  - Assignment 1 had too much time
    - Updated
  - Add more new material
    - Handled mainly by the literature seminar, and also by the renewed focus on cross-lingual parsing

UPPSALA
UNIVERSITET

Recorded lectures

- Available in Studium (with automatic subtitles)
- Until you get access to Studium, you can find the first block of recorded lectures here:
  - https://www.youtube.com/playlist?list=PLH4LBlvRWr95-h6-g8R4P3hUF1wZK3sdh

UPPSALA UNIVERSITET

Work until Friday 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)

UPPSALA UNIVERSITET

Schedule changes

- Lecture January 31:
  - Will have to be moved, info coming shortly
- Final seminar
  - May have to be moved
  - Depending on me possibly presenting at a conference
  - Info will come next week!

UPPSALA UNIVERSITET

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”