



UPPSALA
UNIVERSITET

Introduction

Syntactic analysis (5LN455)

Syntactic parsing (5LN713/5LN717)

2017-11-07

Sara Stymne

Department of Linguistics and Philology

Mostly based on slides from Marco Kuhlmann





UPPSALA
UNIVERSITET

Today

- Introduction to syntactic analysis
- Course information
- Exercises



What is syntax?

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



UPPSALA
UNIVERSITET

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)





UPPSALA
UNIVERSITET

Theoretical frameworks



Chomsky



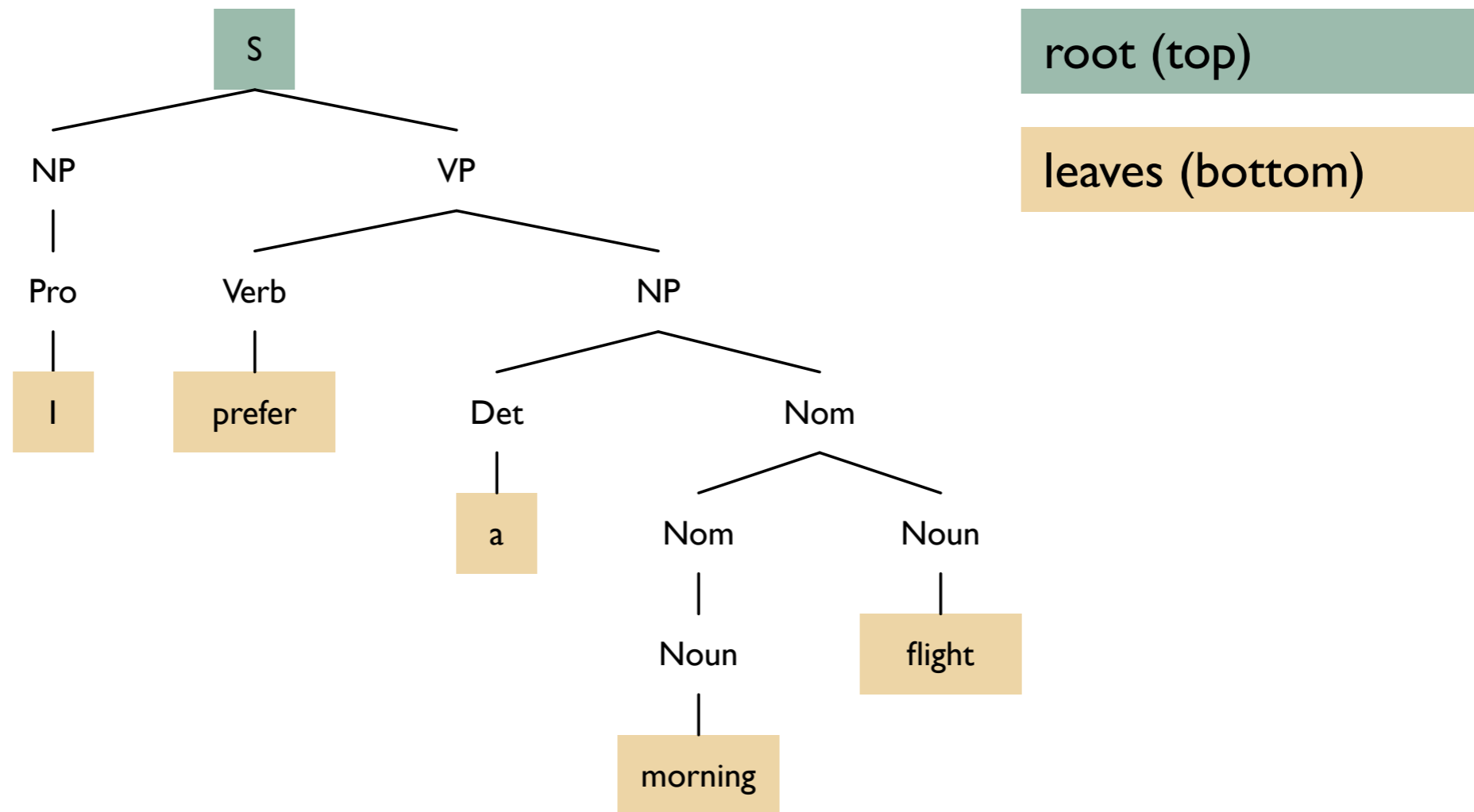
Ajdukiewicz



Tesnière

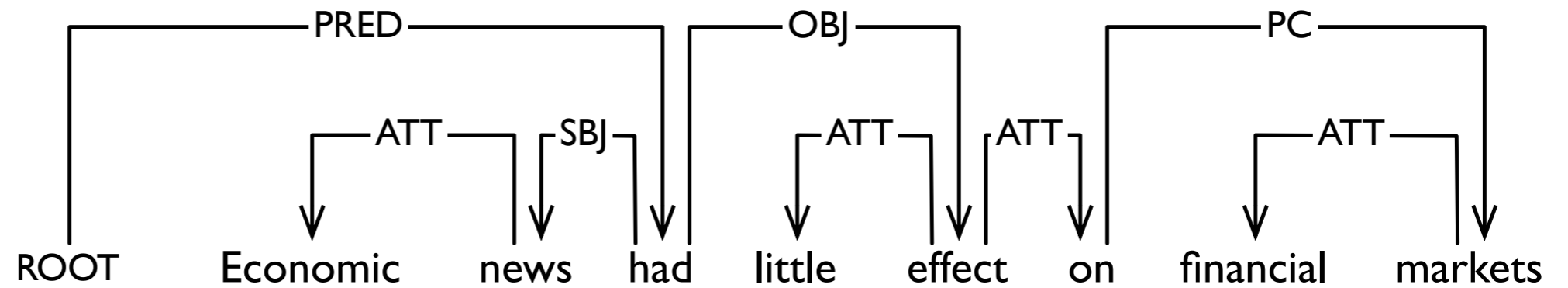


Phrase structure trees



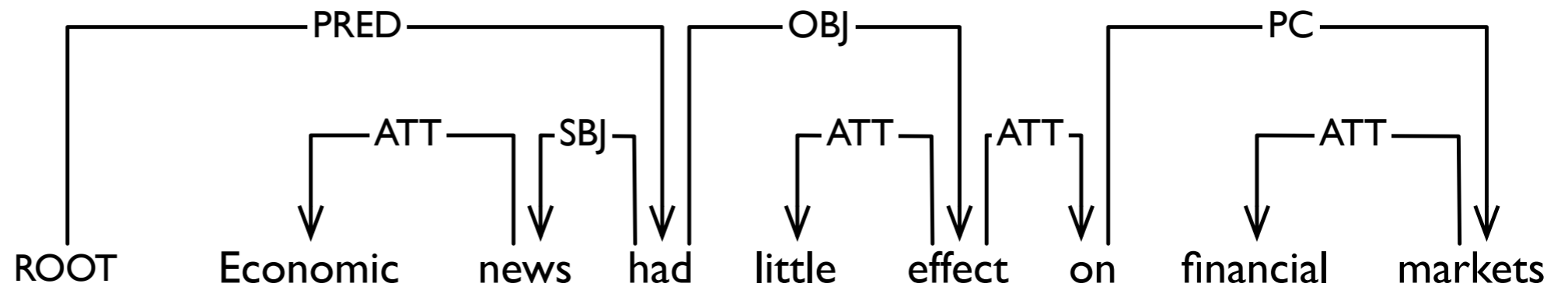
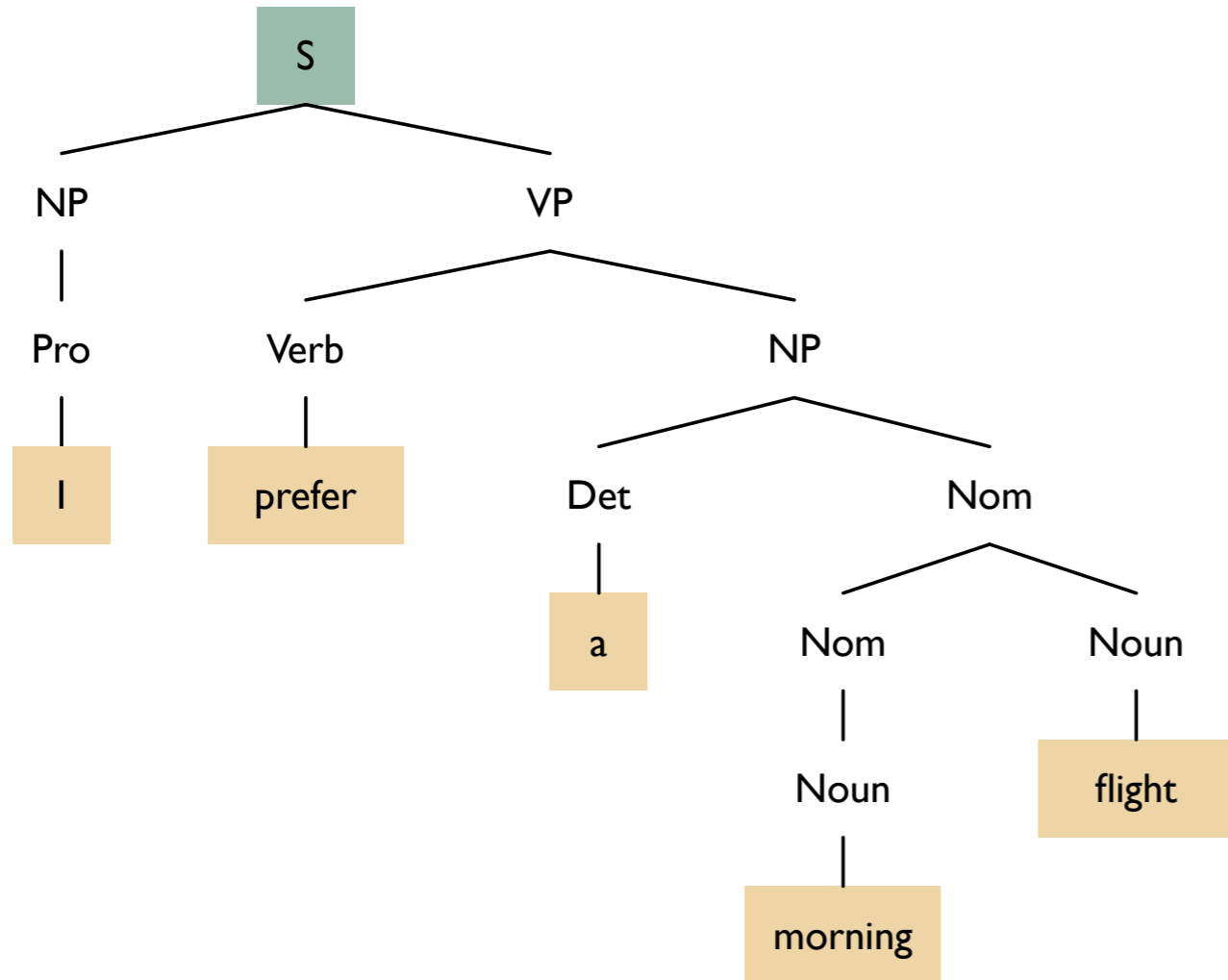


Dependency trees





Phrase structure vs dependency trees





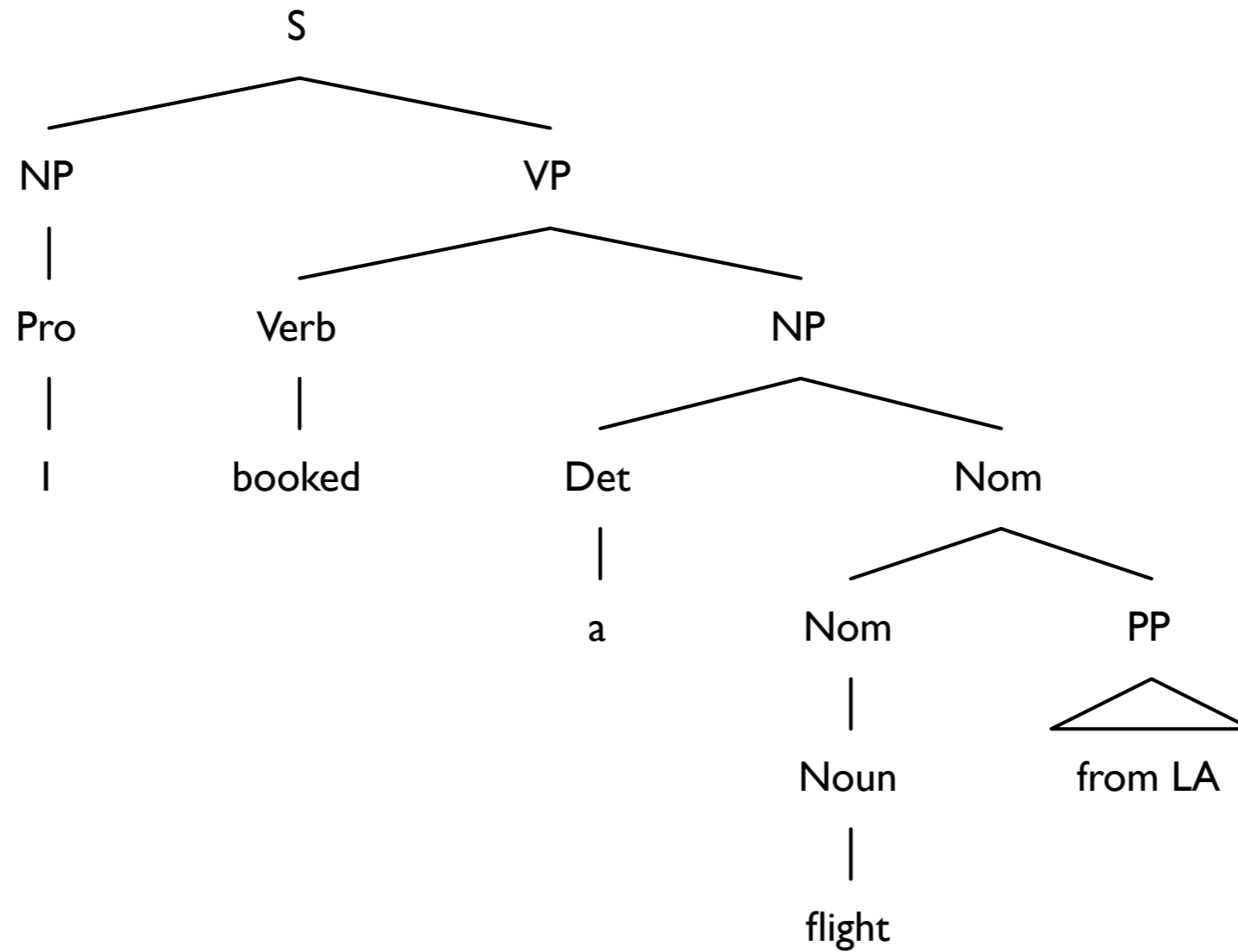
Ambiguity

I booked a flight from LA.

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

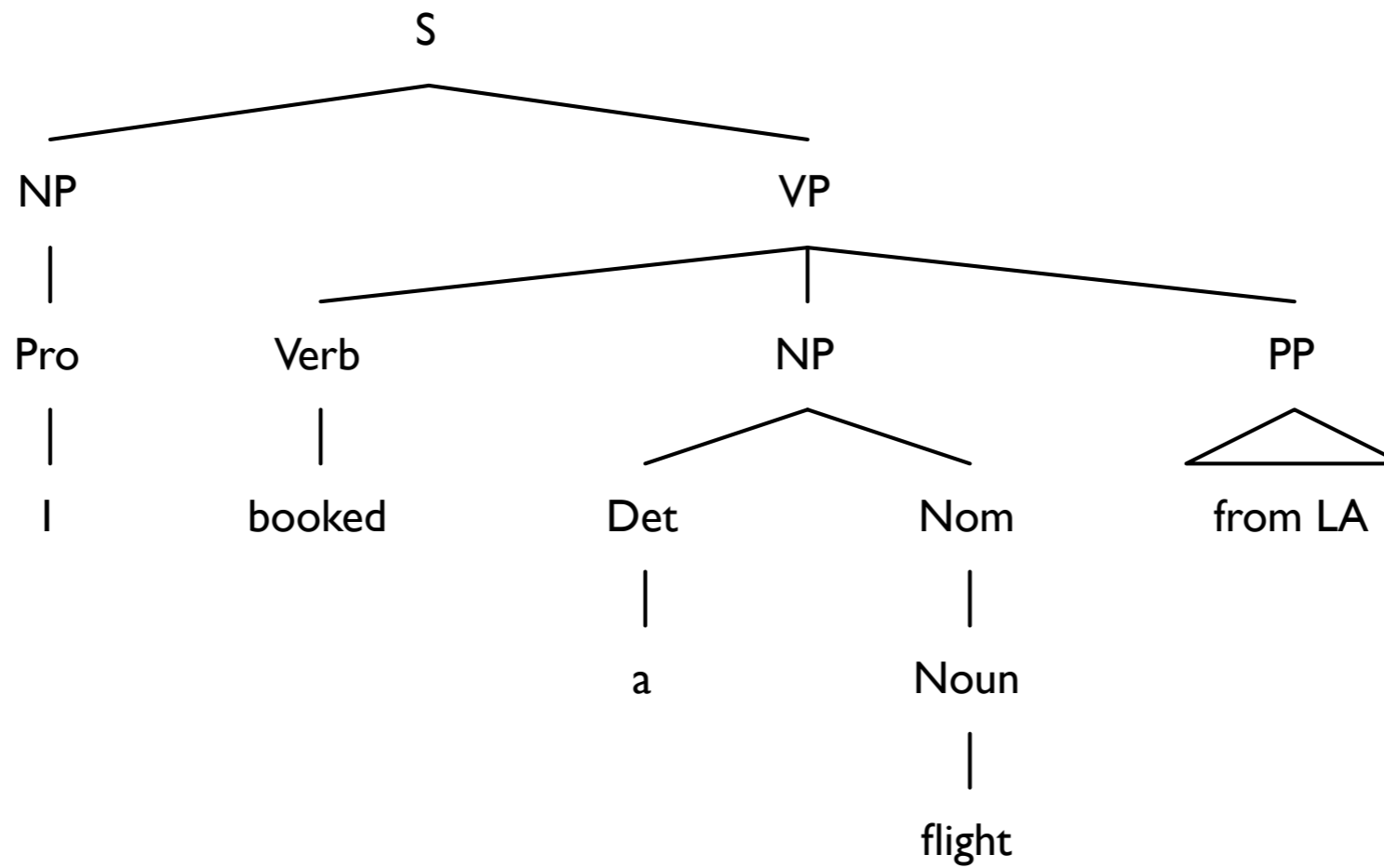


Ambiguity





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

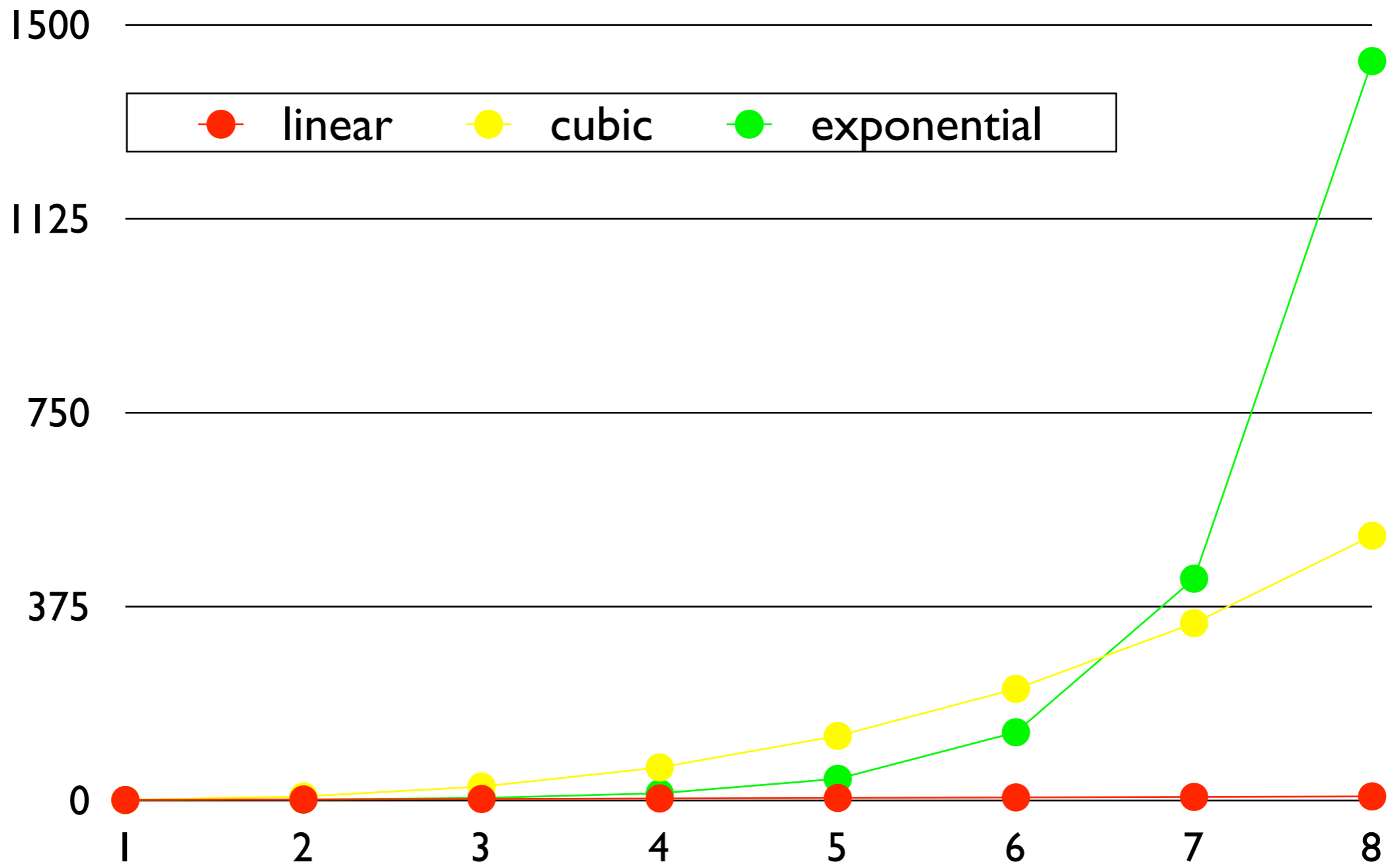
bottom-up

only build trees that match the input

may produce trees that are not rooted at S



How many trees are there?





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.



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.



UPPSALA
UNIVERSITET

Course information





Intended learning outcomes 5LN455

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

- account for the parsing problem of phrase structure grammar and dependency grammar;
- explain at least two different methods for automatic syntactic analysis: one for phrase structure parsing, one for dependency parsing;
- account for statistical methods for syntactic disambiguation;



Intended learning outcomes 5LN455

- apply existing systems that use these methods to realistic data and evaluate them with respect to their accuracy and efficiency;
- implement a central component of at least one approach to syntactic analysis in a suitable programming language.



Examination 5LN455

- Examination is continuous and distributed over four graded assignments and two literature seminars.
- Two assignments are one-page papers.
Time to invest: about 8 hours per assignment.
- The other two assignments are small projects where you need to implement/test parsers.
Time to invest: about 40 hours per assignment.
- In the seminars you will discuss scientific articles about parsing.
Time to invest: about 5 hours per seminar



Assignments 5LN455

1. Written assignment on phrase structure parsing
2. Programming assignment: implement CKY parsing
3. Written assignment on dependency parsing
4. Use and evaluate an existing system for dependency parsing (MaltParser)



Literature seminars (all)

- 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 analysed the article, but it is perfectly fine if you have misunderstood some parts
- The seminars are obligatory
- If you miss a seminar or are unprepared, you will have to hand in a written report.



Learning outcomes and examination

5LN455

- account for the parsing problem of phrase structure grammar and dependency grammar;
paper assignments + seminars
- explain at least two different methods for automatic syntactic analysis: one for phrase structure parsing, one for dependency parsing;
paper assignments + seminars
- account for statistical methods for syntactic disambiguation; **paper assignments**



Learning outcomes and examination

5LN455

- apply existing systems that use these methods to realistic data and evaluate them with respect to their accuracy and efficiency; **project assignment 2**
- implement a central component of at least one approach to syntactic analysis in a suitable programming language. **project assignment 1**



Grading 5LN455

- The assignments are graded with G and VG
- G on the seminars if present, prepared and active. The seminars are obligatory!
- To achieve G on the course:
 - G on all assignments and seminars
- To achieve VG on the course:
 - Same as for G **and** VG on at least two assignments, of which at least one is practical



Intended learning outcomes

5LN713/5LN717

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 (5LN713)



Examination 5LN713/5LN717

- Examination is continuous and distributed over three graded assignments, two literature seminars, and a project (for 7,5 credits)
- Two assignments are small projects where you implement (parts of) parsers.
- Literature review assignment
- Two literature seminars



Grading 5LN713/5LN717

- The assignments are graded with G and VG
- G on the seminars if present, prepared and active. The seminars are obligatory!
- To achieve G on the course:
 - G on all assignments and seminars
- To achieve VG on the course:
 - Same as for G **and** VG on at least two assignments/project



Teachers

- Sara Stymne
 - Examiner, course coordinator, lectures, assignments
- Miryam de Lhoneux
 - Seminars, lecture



Teaching

- 10 lectures
- 2 seminars
- No scheduled supervision / lab hours
- Supervision available on demand (with Sara):
 - Email
 - Knock on office door
 - Book a meeting

Lectures

- Lectures and course books cover basic parsing algorithms, enough material for the bachelor course
- They touch on more advanced material, but master students will need to read up on that independently
- Lectures will usually include small practical tasks
- Do not expect the slides to be self contained! You will not be able to pass the course only by looking at the slides.



Course workload 5LN455

- **7.5 hp means about 200 hours work:**
- 20 h lectures
- 2 h seminars
- 178 h work on your own
 - ~ 96 h assignment work
 - ~ 10 h seminar preparation
 - ~ 72 h additional reading



Course workload 5LN713/5LN717

- **7.5 hp means about 200 hours work:**
- **5 hp means about 133 hours work:**
- 20 h lectures
- 2 h seminars
- 178/111 h work on your own
 - ~ 101 h assignment work (including reading)
 - ~ 10 h seminar preparation
 - ~ 67 h project work (5LN713)



Deadlines

Assignment	Bachelor	Master
1	Dec 4	Dec 4
2	Dec 4	Dec 18
3	Jan 12	Jan 12
4	Jan 12	--
Project	--	Jan 12
<i>Backup</i>	<i>Feb 9</i>	<i>Feb 9</i>

Seminar	Everyone
1	Nov 28
2	Jan 11



Reading: course books

- Daniel Jurafsky and James H. Martin.
Speech and Language Processing. 2nd edition.
Pearson Education, 2009.
Chapters 12-14.
- Sandra Kübler, Ryan McDonald,
and Joakim Nivre. Dependency Parsing.
Morgan and Claypool, 2009.
Chapter 1-4, 6.



Reading: articles

- Seminar 1
 - Mark Johnson. PCFG Models of Linguistic Tree Representations. *Computational Linguistics* 24(4). Pages 613-632.
- Seminar 2
 - Joakim Nivre and Jens Nilsson. Pseudo-Projective Dependency Parsing. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Pages 99-106. Ann Arbor, USA.



Evaluation from last year

- Overall score: 3.9 (3.7 in 2015; 4.75 in 2014)
- Good:
 - Good with practical exercises during the lectures
 - The seminar articles were a good fit with the practical assignments
 - Good mix of tasks for the examination
 - The practical assignments not easy, but led to insights, and felt valuable with respect to future jobs
- Bad:
 - The first practical assignment was difficult, and could maybe have been better explained
 - Will be updated for master students!
 - It would be good if a mathematician could explain algorithms
 - Maybe it would be better with separate deadlines for theoretical and practical assignments
 - Deadlines changed for master students
 - The instructions for the assignments could have been more thorough.
 - Master student assignments are updated!
- Not much change, since the course has been working well for some years. New advanced lectures.



Work to do this week

- Read chapter 12.1-12.7
- Read chapter 13.1-13.3 in preparation for Thursday
- Read descriptions of assignment 1 and 2
- If you need, repeat:
 - parts of grammar course: phrase structure grammars and dependency grammars
 - programming course: practice some python