LING/C SC/PSYC 438/538

Lecture 26

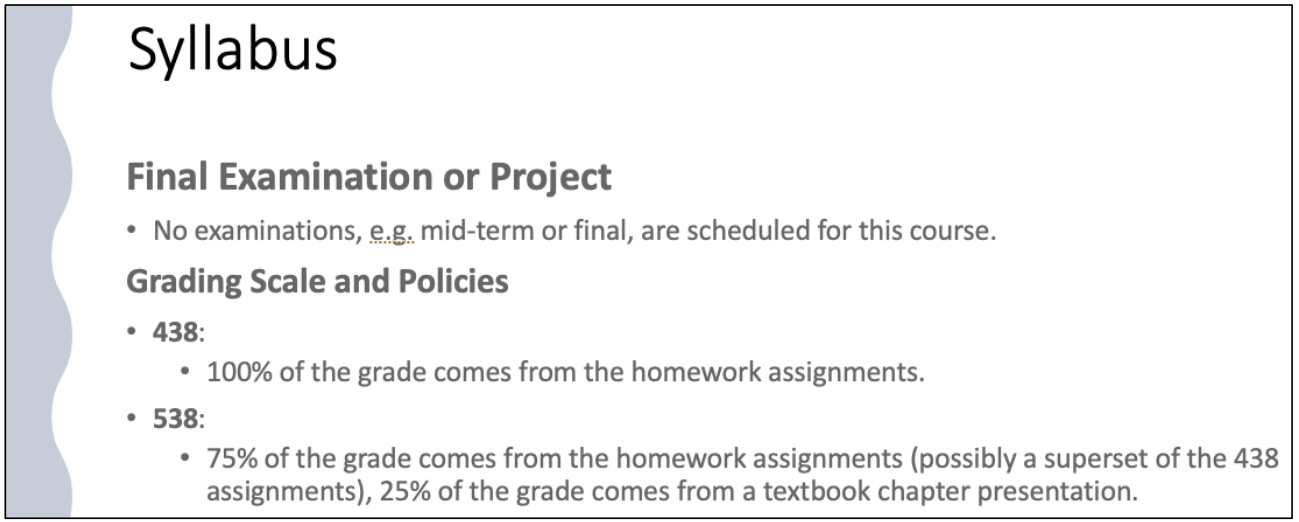
Sandiway Fong

Today's Topics

- Last time, we introduced parse tree using Prolog terms.
- Today:
  - we will go one step further with extra arguments
    - demonstrate an effect on the expressive power of context-free grammars (CFGs) (type-2)
  - we will write our first natural language CFG
- But first, the 538 Homework …

538 Homework

- 438:
  - no homework
  - only do this if you plan to do the HLT Master's Program (538 is required)
- 538:
  - this is mandatory



Syllabus

Final Examination or Project

- No examinations, e.g. mid-term or final, are scheduled for this course.

Grading Scale and Policies

- **438**:
  - 100% of the grade comes from the homework assignments.
- **538**:
  - 75% of the grade comes from the homework assignments (possibly a superset of the 438 assignments), 25% of the grade comes from a textbook chapter presentation.

538 Homework

- Procedure:
  1. first come, first served
  2. pick a section (*if large enough*) or sections from those offered
     - e.g. *a one page section is probably too short*
     - caution: *the PDF is an incomplete draft*
     - *feel free to supplement the material from other sources*
  3. pick something interesting to you
  4. plan for a 7-8 min presentation
  5. plan for a couple of questions
  6. you will be graded on how well you explain the topic in class

538 Homework

- Procedure contd.:
  7. do not simply read the book, use your words: explain, clarify!
  8. make sure you (quickly) explain terms introduced

538 Homework

- Procedure contd.:
  - email me your top three choices (ranked) from those offered
  - SUBJECT: 538 Presentation Your Name
  - cite section(s) title and page numbers in each case
  - choice of presentation dates (ranked)
    - Tuesday 11/28,
    - Thursday 11/30, or
    - Tuesday 12/5

538 Homework

- Procedure contd.:
  - once approved (by me), you can go ahead and make slides
    - PowerPoint is fine
    - PDF is fine
    - we will use my laptop for presentations (*save time*)
    - no "fancy" software
    - you can use the whiteboard too

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 12 Constituency Grammars 260
- 12.1 Constituency..... 261
- 12.2 Context-Free Grammars..... 261
- 12.3 Some Grammar Rules for English..... 266
- 12.4 Treebanks 273
- 12.5 Grammar Equivalence and Normal Form 278
- 12.6 Lexicalized Grammars..... 279
- 12.7 Summary..... 284

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 13 Constituency Parsing 288
- 13.1 Ambiguity.....288
- 13.2 CKY Parsing: A Dynamic Programming Approach 290
- 13.3 Span-Based Neural Constituency Parsing 296
- 13.4 Evaluating Parsers..... 298
- 13.5 Partial Parsing 299
- 13.6 CCG Parsing..... 300
- 13.7 Summary..... 307

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 14 Dependency Parsing 310
- 14.1 Dependency Relations..... 311
- 14.2 Dependency Formalisms..... 313
- 14.3 DependencyTreebanks 314
- 14.4 Transition-BasedDependencyParsing 316
- 14.5 Graph-BasedDependencyParsing 325
- 14.6 Evaluation 331
- 14.7 Summary..... 332

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 15 Logical Representations of Sentence Meaning 335
- 15.1 Computational Desiderata for Representations 336
- 15.2 Model-Theoretic Semantics.....338
- 15.3 First-Order Logic 341
- 15.4 Event and State Representations..... 348
- 15.5 Description Logics..... 353
- 15.6 Summary..... 359

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 17 Information Extraction 363
- 17.1 Relation Extraction 364
- 17.2 Relation Extraction Algorithms.....367
- 17.3 Extracting Times..... 375
- 17.4 Extracting Events and their Times..... 379
- 17.5 Template Filling.....381
- 17.6 Summary..... 383

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 18 Word Senses and WordNet 386
- 18.1 Word Senses..... 387
- 18.2 Relations Between Senses..... 389
- 18.3 WordNet: A database of Lexical Relations 391
- 18.4 Word Sense Disambiguation..... 394
- 18.5 Alternate WSD algorithms and Tasks..... 397
- 18.6 Using Thesauruses to Improve Embeddings 400
- 18.7 Word Sense Induction.....400
- 18.8 Summary..... 401

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 19 Semantic Role Labeling 405
- 19.1 Semantic Roles 406
- 19.2 Diathesis Alternations..... 407
- 19.3 Semantic Roles: Problems with Thematic Roles 408
- 19.4 The Proposition Bank 409
- 19.5 FrameNet 410
- 19.6 Semantic Role Labeling..... 412
- 19.7 Selectional Restrictions 416
- 19.8 Primitive Decomposition of Predicates 420
- 19.9 Summary..... 422

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 20 Lexicons for Sentiment, Affect, and Connotation 425
- 20.1 Defining Emotion 426
- 20.2 Available Sentiment and Affect Lexicons 428
- 20.3 Creating Affect Lexicons by Human Labeling 429
- 20.4 Semi-supervised Induction of Affect Lexicons 431
- 20.5 Supervised Learning of Word Sentiment 434
- 20.6 Using Lexicons for Sentiment Recognition 439
- 20.7 Using Lexicons for Affect Recognition 440
- 20.8 Lexicon-based methods for Entity-Centric Affect 441
- 20.9 Connotation Frames..... 441
- 20.10 Summary..... 443

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 21 Coreference Resolution 445
- 21.1 Coreference Phenomena: Linguistic Background 448
- 21.2 Coreference Tasks and Datasets..... 453
- 21.3 Mention Detection..... 454
- 21.4 Architectures for Coreference Algorithms 457
- 21.5 Classifiers using hand-built features. 459
- 21.6 A neural mention-ranking algorithm 461
- 21.7 Evaluation of Coreference Resolution. 464
- 21.8 Winograd Schema problems..... 465
- 21.9 Gender Bias in Coreference.....466
- 21.10 Summary..... 468

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 22 Discourse Coherence 472
- 22.1 Coherence Relations..... 474
- 22.2 Discourse Structure Parsing..... 477
- 22.3 Centering and Entity-Based Coherence 481
- 22.4 Representation learning models for local coherence 485
- 22.5 Global Coherence 487
- 22.6 Summary..... 490

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 23 Question Answering 494
- 23.1 Information Retrieval 495
- 23.2 IR-based Factoid Question Answering 503
- 23.3 Entity Linking.....507
- 23.4 Knowledge-based Question Answering 511
- 23.5 Using Language Models to do QA 514
- 23.6 Classic QA Models 515
- 23.7 Evaluation of Factoid Answers 518

538 Homework

Use ed3book.pdf (*I will email a link to you all*)

- 24 Chatbots & Dialogue Systems 521
- 24.1 Properties of Human Conversation 522
- 24.2 Chatbots 525
- 24.3 GUS: Simple Frame-based Dialogue Systems 533
- 24.4 The Dialogue-State Architecture 537
- 24.5 Evaluating Dialogue Systems 546
- 24.6 Dialogue System Design..... 549
- 24.7 Summary..... 551

Last Time

- Extra arguments are powerful
  - they allow us to impose (grammatical) constraints and change the expressive power of the system
    - if used as read-able memory (cf. *Turing Machine discussion*)

- **Example**:
  - $a^n b^n c^n$ n>0 is not a context-free language (type-2)
  - *i.e. you cannot write rules of the form* n –––> RHS *to generate this language*
  - in fact, it's context-sensitive (type-1)

Extra arguments

- Let's start by writing a context-free grammar (CFG) + extra argument for the parse for $\{ a^p b^q c^r \mid p, q, r > 0 \}$: `abc.prolog`

```
1. s(s(A,B,C)) --> a(A), b(B), c(C).
2. a(a(a)) --> [a].
3. a(a(a,A)) --> [a], a(A).
4. b(b(b)) --> [b].
5. b(b(b,B)) --> [b], b(B).
6. c(c(c)) --> [c].
7. c(c(c,C)) --> [c], c(C).
```

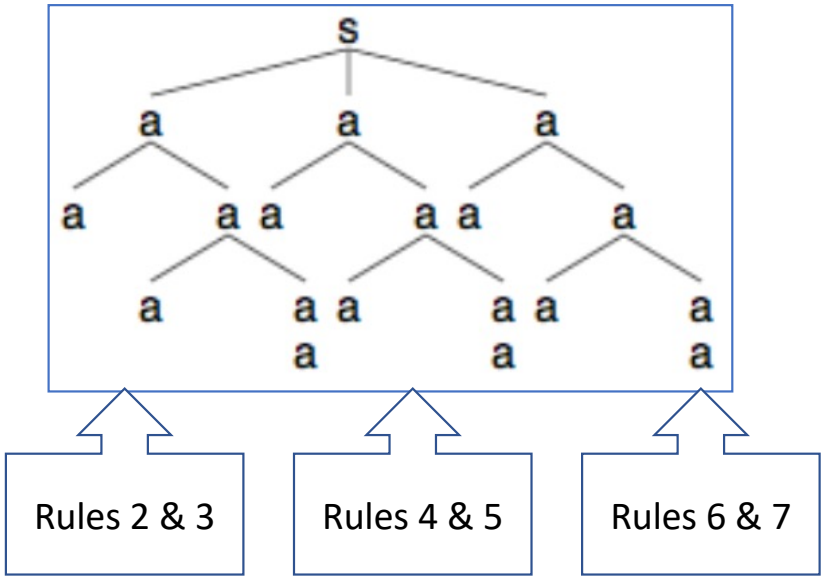
Extra arguments

- $\{ a^p b^q c^r \mid p, q, r > 0\}$: `abc.prolog`
- `?- [abc].`
- `?- s(Parse,[],[]).`
- **`false.`**
- `?- s(Parse,[a,b,c],[]).`
- `Parse = s(a(a), b(b), c(c)) ;`
- **`false.`**
- `?- s(Parse,[a,b,b,c,c,c],[]).`
- `Parse = s(a(a), b(b, b(b)), c(c, c(c, c(c)))) ;`
- **`false.`**
- `?- s(Parse,[a,b,b,c,a],[]).`
- **`false.`**

Extra arguments

- A context-free grammar (CFG) + extra argument (EA) for the context-sensitive language { $a^n b^n c^n$ | n>0}, `abc2.prolog`

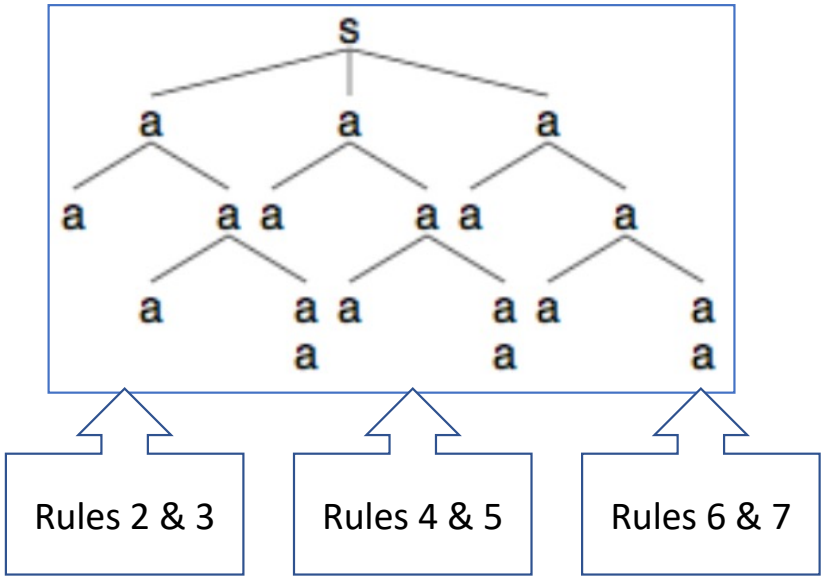
```
1. s(s(A,B,C)) --> a(A), b(B), c(C).
2. a(a(a)) --> [a].
3. a(a(a,X)) --> [a], a(X).
4. b(a(a)) --> [b]. % cf. b(b)
5. b(a(a,X)) --> [b], b(X).
6. c(a(a)) --> [c]. % cf. c(c)
7. c(a(a,X)) --> [c], c(X).
```



Extra arguments

- A context-free grammar (CFG) + extra argument (EA) for the context-sensitive language { $a^n b^n c^n$ | n>0}, `abc2.prolog`

```
1. s(s(A,A,A)) --> a(A), b(A), c(A).
2. a(a(a)) --> [a].
3. a(a(a,X)) --> [a], a(X).
4. b(a(a)) --> [b]. % cf. b(b)
5. b(a(a,X)) --> [b], b(X).
6. c(a(a)) --> [c]. % cf. c(c)
7. c(a(a,X)) --> [c], c(X).
```

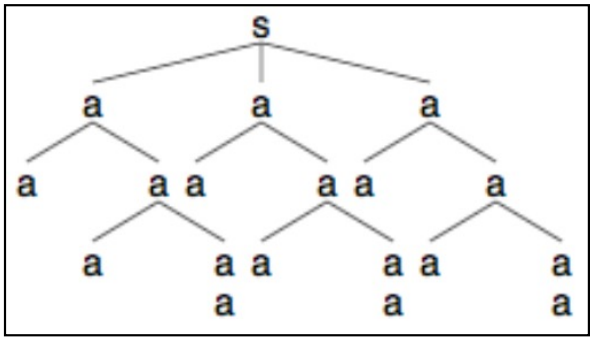


Extra arguments

- A CFG+EA for $a^n b^n c^n$ n>0:

Set membership question

```
?- [abc2].
?- s(Parse,[a,a,a,b,b,b,c,c,c],[]).
Parse = s(a(a, a(a, a(a))), a(a, a(a, a(a))), a(a, a(a, a(a)))) ;
false.
?- s(Parse,[a,a,a,b,b,b,c,c],[]).
false.
?- s(Parse,[a,a,a,b,b,c,c],[]).
false.
?- s(Parse,[a,a,b,b,c,c],[]).
Parse = s(a(a, a(a)), a(a, a(a)), a(a, a(a))) ;
false.
?- s(Parse,[],[]).
false.
```



Extra arguments

- A CFG+EA for $a^n b^n c^n$ n>0:

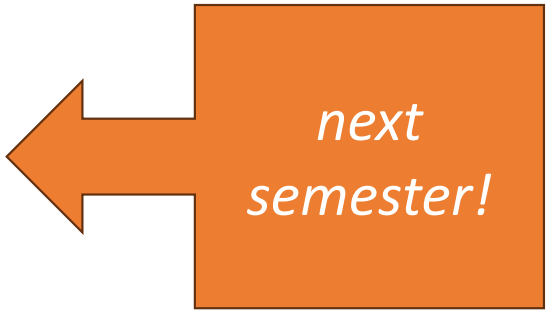
```
?- s(_,[a,a,b,b,c,c,c],[]).
false.

?- s(_,[a,a,b,b,c,c],[]).
true .

?- s(_,[a,a,b,b,c],[]).
false.

?- s(_,[a,a,b,b,c,c,c],[]).
false.

?- s(_,[a,a,a,b,b,b,c,c,c],[]).
true .
```

Set membership question

- underscore (_): a variable
  - *don't care what its value is,*
  - *don't tell me.*

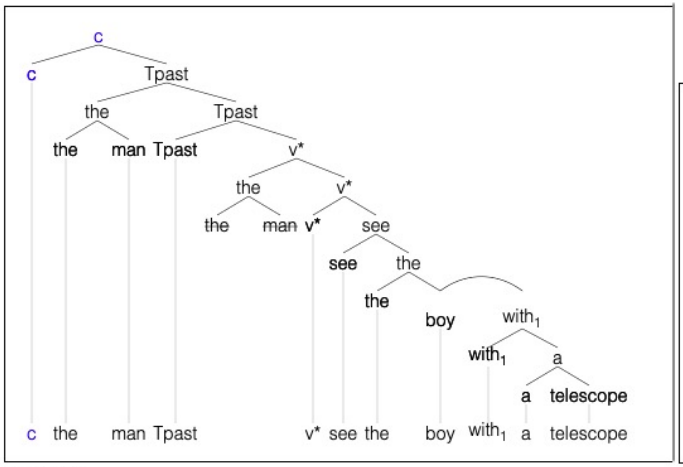
Context-sensitive Grammar (CSG)

- Type-1:
  - **note**: more than one symbol on the LHS of the rule!
- `s --> [a,b,c].`
- `s --> [a],a,[b,c].`
- `a --> [a,b], c.`
- `a --> [a],a,[b],c.`
- `c,[b] --> [b], c.`
- `c,[c] --> [c,c].`

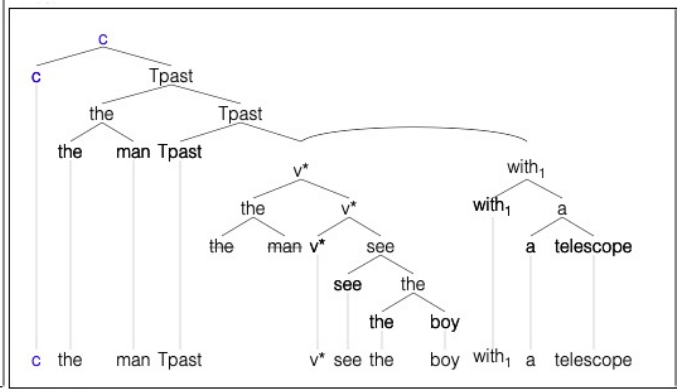


Natural Language Grammar

- Let's write a Prolog natural language grammar!
  - Example: *a boy saw the man with a telescope*
  - if you're a linguist, you can write the rules directly
  - if you're not, you may have trouble diagramming the sentence



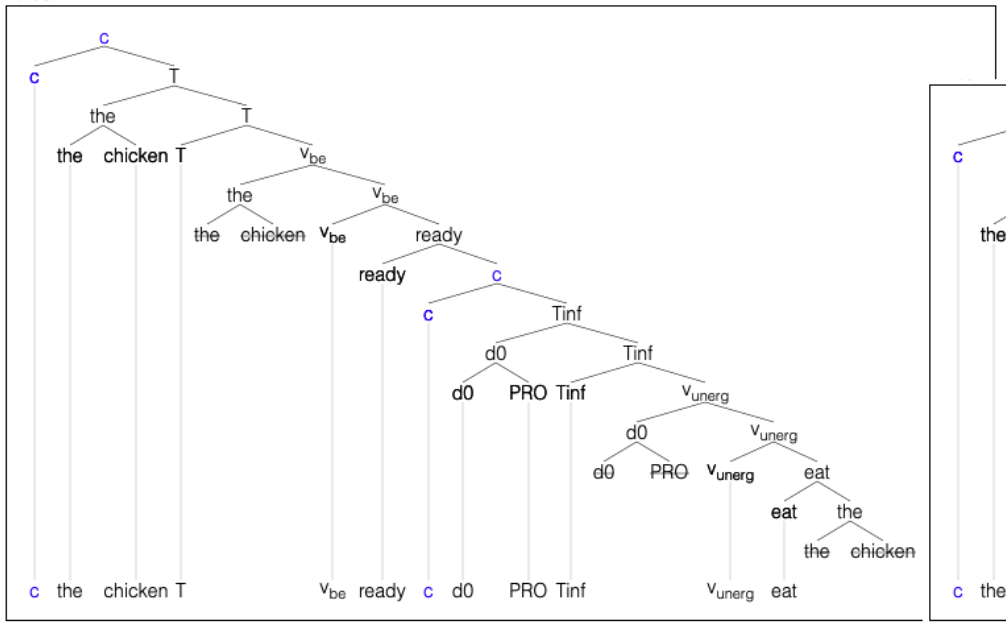
Spell-out:
the man -ed(sg) see the boy with a telescope (after morpheme realization)
the man see -ed(sg) the boy with a telescope (after affix-hop)
the man see -ed(sg) the boy with a telescope (after morpheme realization, stage 2)
the man saw the boy with a telescope



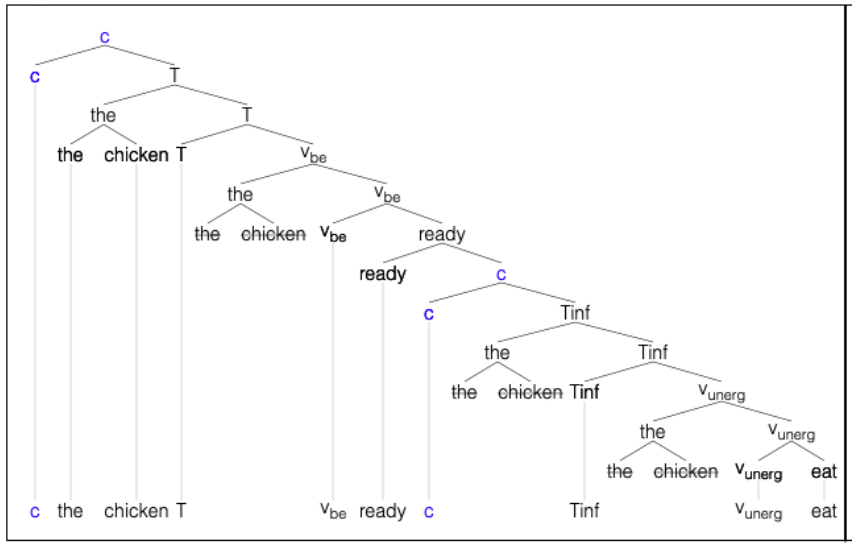
Spell-out:
the man -ed(sg) see the boy with a telescope (after morpheme realization)
the man see -ed(sg) the boy with a telescope (after affix-hop)
the man see -ed(sg) the boy with a telescope (after morpheme realization, stage 2)
the man saw the boy with a telescope

Natural Language Grammar

- Next semester, topics could include implementations of syntactic theory:



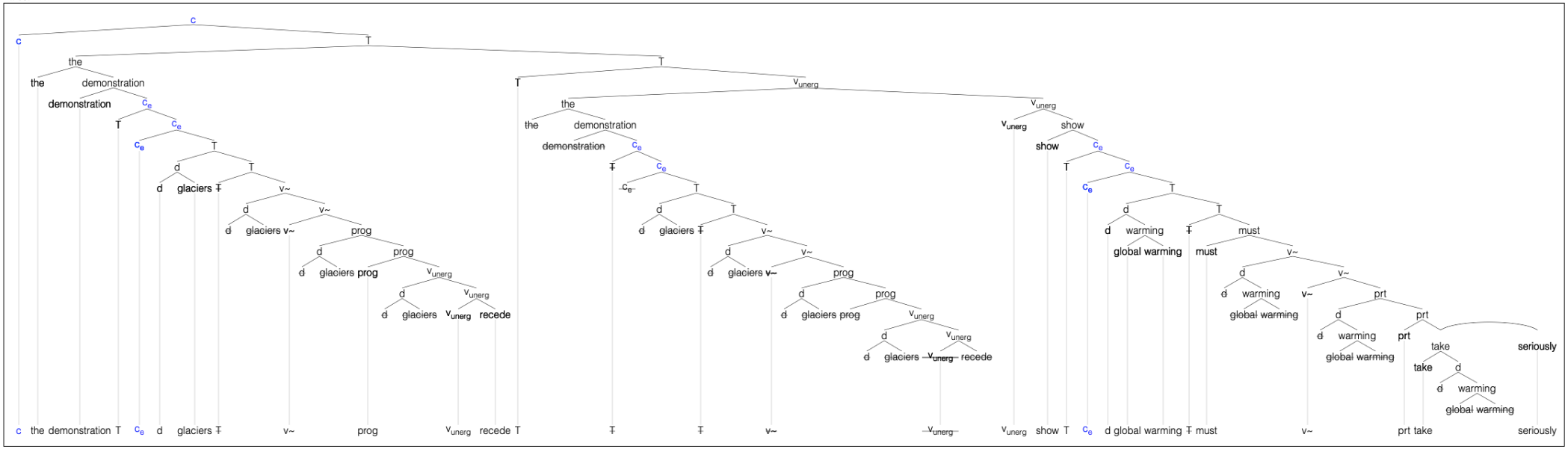
Spell-out:
the chicken -s be ready to eat (after morpheme realization)
the chicken be -s ready to eat (after affix-hop)
the chicken be -s ready to eat (after morpheme realization, stage 2)
the chicken is ready to eat



Spell-out:
the chicken -s be ready to eat (after morpheme realization)
the chicken be -s ready to eat (after affix-hop)
the chicken be -s ready to eat (after morpheme realization, stage 2)
the chicken is ready to eat

Natural Language Grammar

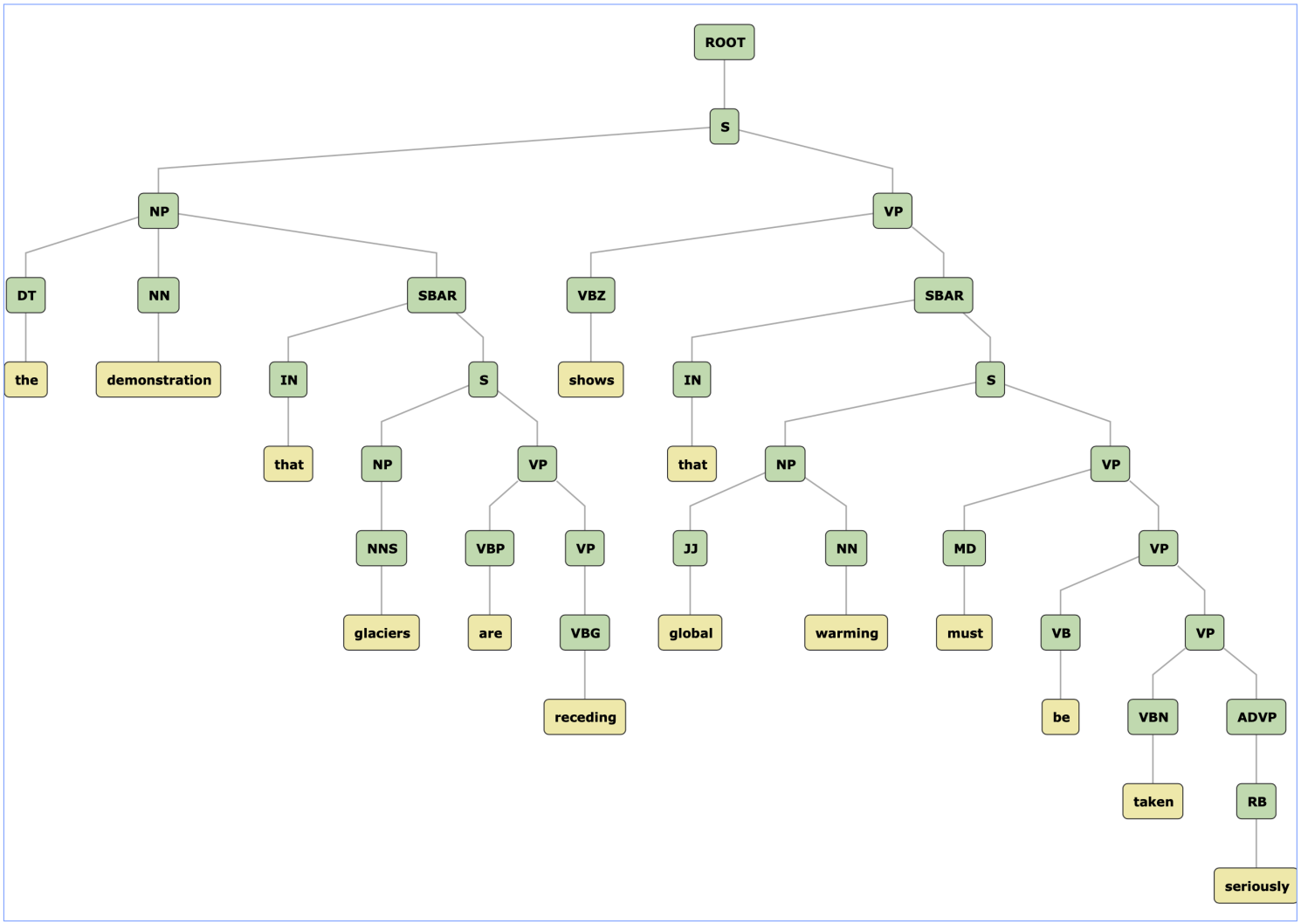
- Next semester, topics could include implementations of syntactic theory:



Spell-out:
the demonstration that glaciers -re be -ing recede -s show that global warming -s must be -en take seriously (after morpheme realization)
the demonstration that glaciers be -re recede -ing show -s that global warming must -s be take -en seriously (after affix-hop)
the demonstration that glaciers be -re recede -ing show -s that global warming must -s be take -en seriously (after morpheme realization, stage 2)
the demonstration that glaciers are receding shows that global warming must be taken seriously

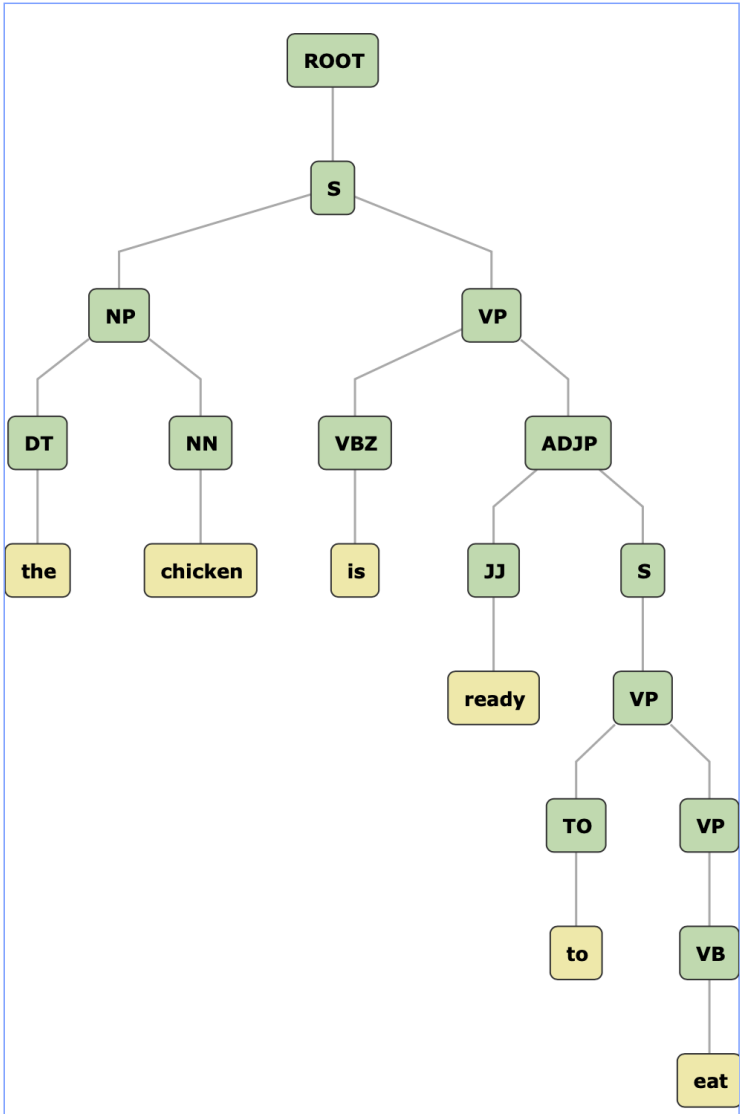
Parser

Constituency Parse:



Parser

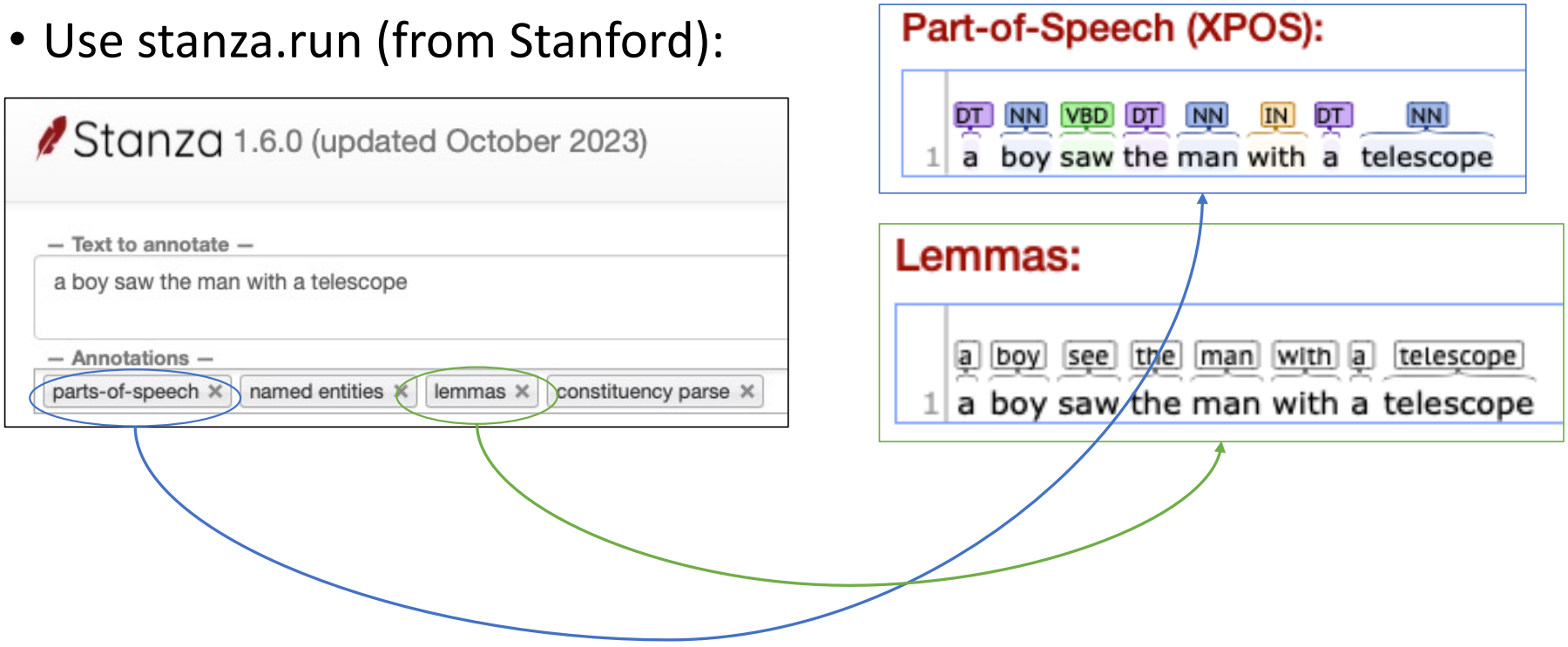
Constituency Parse:



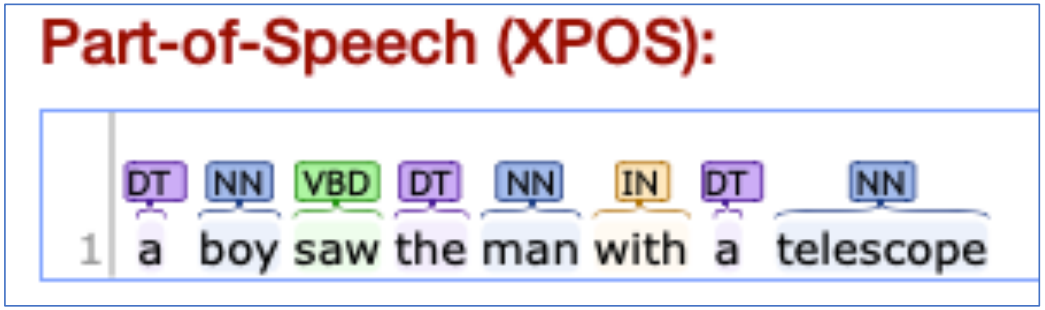
Which parse is this?

Natural Language Grammar

- Use stanza.run (from Stanford):



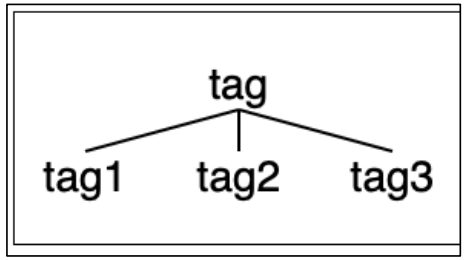
Natural Language Grammar

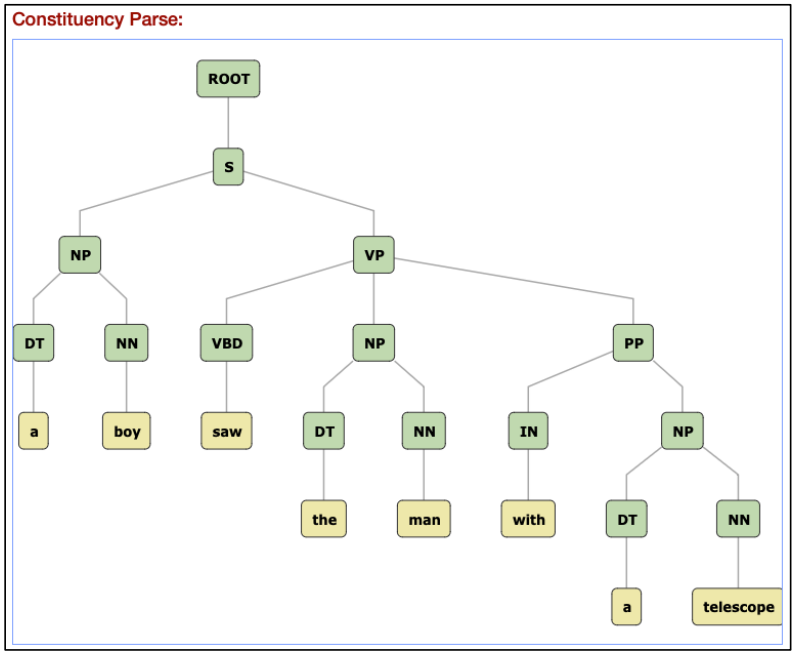


- POS tag to grammar rule:
  - **note**: use lowercase for tags and words
    - (*variables begin with an uppercase letter*)
  - `tag(tag(word)) --> [word].`

Natural Language Grammar

- Phrasal tag to grammar rule:
  - **note**: use lowercase for tags and words (*variables begin with an uppercase letter*)
  - `tag(tag(`*`Subphrase,..,Subphrase`*`')) -->` tag_1`(`*`Subphrase`*$_1$`),..,` tag_n`(`*`Subphrase`*$_n$`).`





Natural Language Grammar