Context-free grammars and syntactic analysis

Roger Levy

Massachusetts Institute of Technology
Department of Brain & Cognitive Sciences

Describing the hierarchical structure of sentences

- Sentences are not just sequences of words

Describing the hierarchical structure of sentences

- Sentences are not just sequences of words
- Some words are closely associated with other words into **phrases**

Describing the hierarchical structure of sentences

- Sentences are not just sequences of words
- Some words are closely associated with other words into **phrases**
- These phrases are in turn associated with other words or phrases to form larger phrases

Describing the hierarchical structure of sentences

- Sentences are not just sequences of words
- Some words are closely associated with other words into **phrases**
- These phrases are in turn associated with other words or phrases to form larger phrases
- The largest phrase is the **sentence**

Describing the hierarchical structure of sentences

- Sentences are not just sequences of words
- Some words are closely associated with other words into **phrases**
- These phrases are in turn associated with other words or phrases to form larger phrases
- The largest phrase is the **sentence**
- We use **formal grammars** to describe these phrasal arrangements

Describing the hierarchical structure of sentences

- Sentences are not just sequences of words
- Some words are closely associated with other words into **phrases**
- These phrases are in turn associated with other words or phrases to form larger phrases
- The largest phrase is the **sentence**
- We use **formal grammars** to describe these phrasal arrangements
- The formal grammatical description of a sentence gives us considerable inroads into understanding its meaning

Context-free Grammars

- A context-free grammar (CFG) consists of a tuple (N, V, S, R) such that:

Context-free Grammars

- A context-free grammar (CFG) consists of a tuple (N, V, S, R) such that:
  - N is a finite set of **non-terminal** symbols;

Context-free Grammars

- A context-free grammar (CFG) consists of a tuple (N, V, S, R) such that:
  - N is a finite set of **non-terminal** symbols;
  - V is a finite set of **terminal** symbols;

Context-free Grammars

- A context-free grammar (CFG) consists of a tuple (N, V, S, R) such that:
  - N is a finite set of **non-terminal** symbols;
  - V is a finite set of **terminal** symbols;
  - $S \in N$ is the **start** symbol;

Context-free Grammars

- A context-free grammar (CFG) consists of a tuple (N, V, S, R) such that:
  - N is a finite set of **non-terminal** symbols;
  - V is a finite set of **terminal** symbols;
  - $S \in N$ is the **start** symbol;
  - R is a finite set of **rules** of the form $X \rightarrow \alpha$ where $X \in N$ and α is a sequence of symbols drawn from $N \cup V$.

Context-free Grammars

- A context-free grammar (CFG) consists of a tuple (N, V, S, R) such that:
  - N is a finite set of **non-terminal** symbols;
  - V is a finite set of **terminal** symbols;
  - $S \in N$ is the **start** symbol;
  - R is a finite set of **rules** of the form $X \rightarrow \alpha$ where $X \in N$ and α is a sequence of symbols drawn from $N \cup V$.
- A CFG **derivation** is the recursive expansion of non-terminal symbols in a string by rules in R, starting with S;

Context-free Grammars

- A context-free grammar (CFG) consists of a tuple (N, V, S, R) such that:
  - N is a finite set of **non-terminal** symbols;
  - V is a finite set of **terminal** symbols;
  - $S \in N$ is the **start** symbol;
  - R is a finite set of **rules** of the form $X \rightarrow \alpha$ where $X \in N$ and α is a sequence of symbols drawn from $N \cup V$.
- A CFG **derivation** is the recursive expansion of non-terminal symbols in a string by rules in R, starting with S;
- A **derivation tree** T is the history of those rule applications.

Context-free Grammars: an example

Let our grammar (the rule-set R) be

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Context-free Grammars: an example

Let our grammar (the rule-set R) be

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

The nonterminal set N is $\{S, NP, VP, Det, N, P, V\}$, the terminal set V is $\{the, dog, cat, near, growled\}$, and our start symbol S is S.

Context-free Grammars: an example II

S →NP VP
NP→Det N
NP→NP PP
PP→P NP
VP→V

Det→ the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

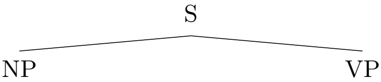
S

Context-free Grammars: an example II

S →NP VP
NP→Det N
NP→NP PP
PP→P NP
VP→V

Det→ the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

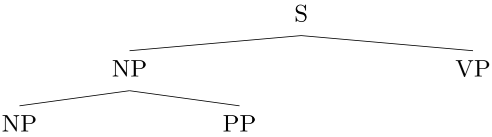


Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

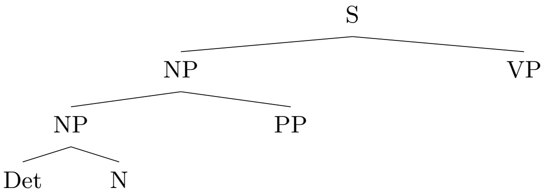


Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

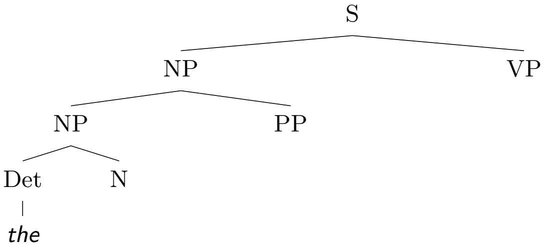


Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

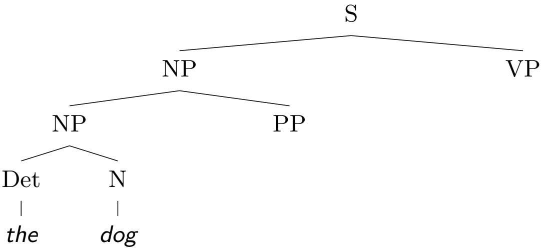


Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

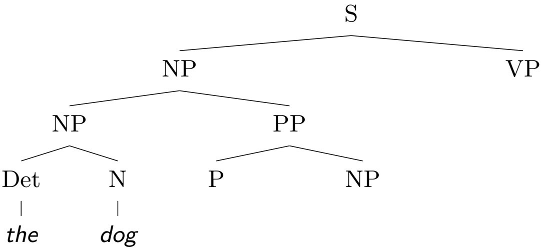


Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

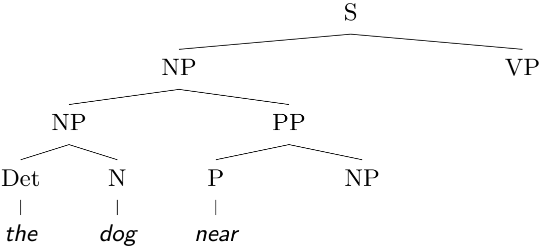


Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

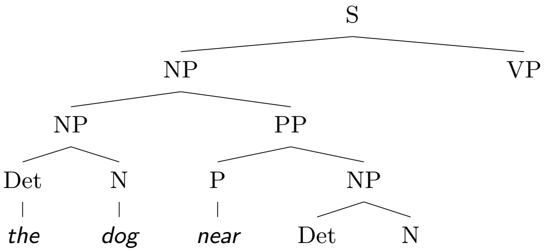


Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

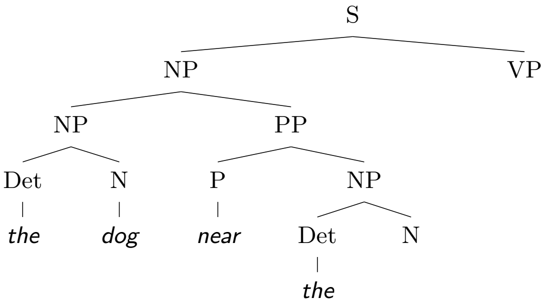


Context-free Grammars: an example II

S →NP VP
NP→Det N
NP→NP PP
PP→P NP
VP→V

Det→ the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

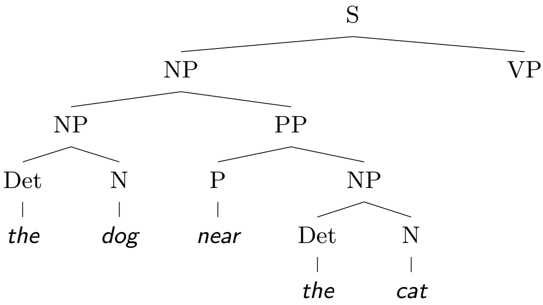


Context-free Grammars: an example II

S →NP VP
NP→Det N
NP→NP PP
PP→P NP
VP→V

Det→ the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

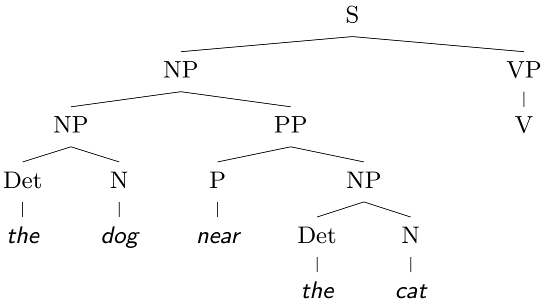


Context-free Grammars: an example II

S →NP VP
NP→Det N
NP→NP PP
PP→P NP
VP→V

Det→ the
N → dog
N → cat
P → near
V → growled

Here is a *derivation* and the resulting *derivation tree*:

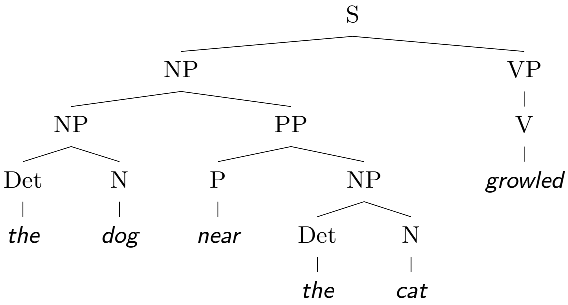


Context-free Grammars: an example II

S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

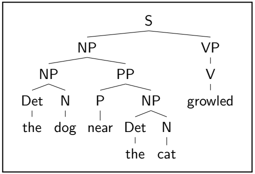
Here is a *derivation* and the resulting *derivation tree*:



Some questions we can ask about a CFG

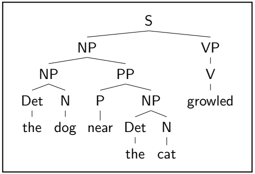
- How many strings does it produce?
- How many trees does it produce?
- Is it ambiguous? (in the general case, this problem is known to be UNDECIDABLE; Hopcroft & Ullman, 1979, Sima'an, 2002)

Context-free rules as generating phrase-structure descriptions



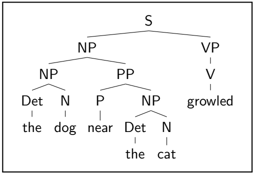
- The nodes in the derivation tree offer a *structural description* of the **syntax** of the generated string

Context-free rules as generating phrase-structure descriptions



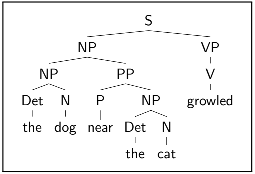
- The nodes in the derivation tree offer a *structural description* of the **syntax** of the generated string
- In generative grammar, this structural description is the **phrase structure** of the strings

Context-free rules as generating phrase-structure descriptions



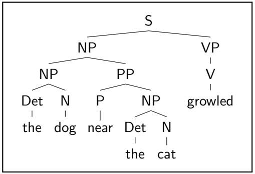
- The nodes in the derivation tree offer a *structural description* of the **syntax** of the generated string
- In generative grammar, this structural description is the **phrase structure** of the strings
- These structural descriptions play a role in the theory of linguistic structure and meaning:

Context-free rules as generating phrase-structure descriptions



- The nodes in the derivation tree offer a *structural description* of the **syntax** of the generated string
- In generative grammar, this structural description is the **phrase structure** of the strings
- These structural descriptions play a role in the theory of linguistic structure and meaning:
  - Syntactic regularities are theoretically stated in terms of these structural descriptions

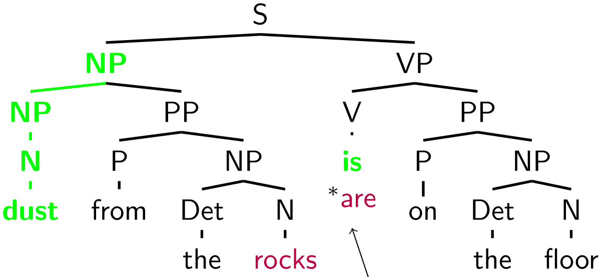
Context-free rules as generating phrase-structure descriptions



- The nodes in the derivation tree offer a *structural description* of the **syntax** of the generated string
- In generative grammar, this structural description is the **phrase structure** of the strings
- These structural descriptions play a role in the theory of linguistic structure and meaning:
  - Syntactic regularities are theoretically stated in terms of these structural descriptions
  - Within the broader theory of linguistic structure, phrase structure also interfaces with other levels of representation, such as prosody and meaning composition

Syntactic regularities: subjects and subject–verb agreement

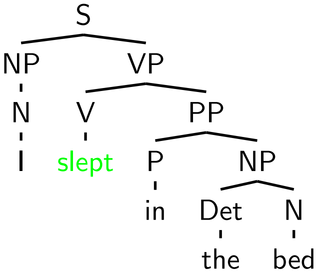
- The **subject** of an English clause is the NP left sister of the VP. The **head word** of a (non-coordinated) subject determines person+number agreement for the verb.



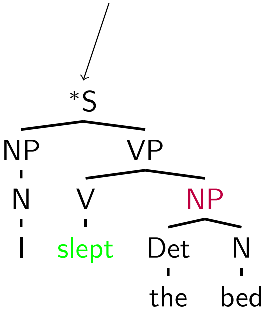
*=would be ungrammatical
with *are* instead of *is*

Syntactic regularities: argument structure

- Verbs differ in their **argument structure**, which constrains the English subject and children of the VP.

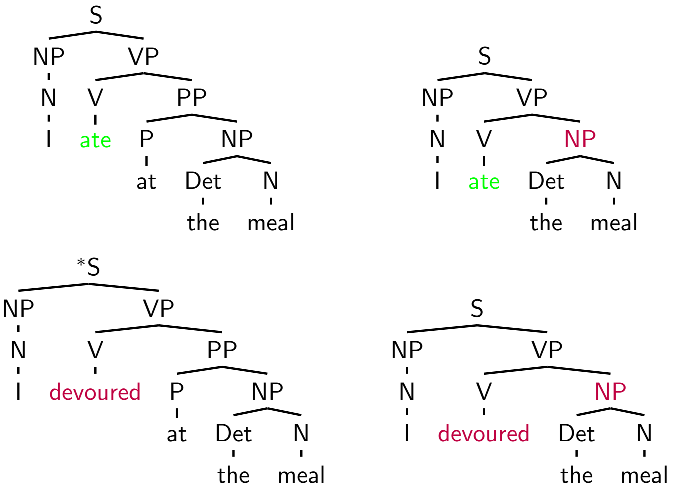


*=an ungrammatical structure



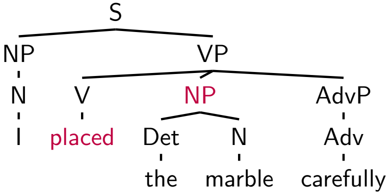
Syntactic regularities: argument structure

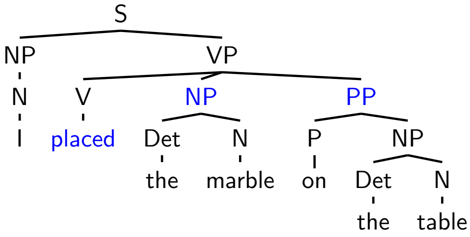
- Argument structure requirements are defined on **syntactic categories**, and are correlated with but not fully predictable from verb semantics:

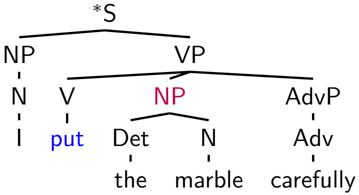


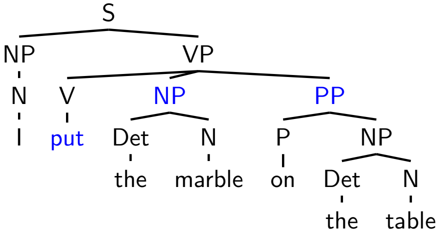
Syntactic regularities: argument structure

Argument structure requirements can be very verb-specific. For example, in English, *put* is the only verb that *requires* an explicit location argument:









Agreement & argument structure in CFGs

- Agreement, argument structure, and other constraints can be built into context-free grammars by adding **features** to phrasal categories, e.g.:

$$\begin{array}{lll}
\mathrm{VP}\rightarrow\mathrm{V}_{\langle\rangle} & \mathrm{V}_{\langle\rangle}\rightarrow\text{slept} & \mathrm{V}_{\langle\mathrm{NP}\rangle}\rightarrow\text{devoured} \\
\mathrm{VP}\rightarrow\mathrm{V}_{\langle\mathrm{NP}\rangle}\ \mathrm{NP} & \mathrm{V}_{\langle\rangle}\rightarrow\text{ate} & \mathrm{V}_{\langle\mathrm{NP}\rangle}\rightarrow\text{placed} \\
\mathrm{VP}\rightarrow\mathrm{V}_{\langle\mathrm{NP},\mathrm{PP}\rangle}\ \mathrm{NP}\ \mathrm{PP} & \mathrm{V}_{\langle\mathrm{NP}\rangle}\rightarrow\text{ate} & \mathrm{V}_{\langle\mathrm{NP},\mathrm{PP}\rangle}\rightarrow\text{put}
\end{array}$$

Agreement & argument structure in CFGs

- Agreement, argument structure, and other constraints can be built into context-free grammars by adding **features** to phrasal categories, e.g.:

$$\begin{array}{lll}
\mathrm{VP}\rightarrow\mathrm{V}_{\langle\rangle} & \mathrm{V}_{\langle\rangle}\rightarrow\mathrm{slept} & \mathrm{V}_{\langle\mathrm{NP}\rangle}\rightarrow\mathrm{devoured} \\
\mathrm{VP}\rightarrow\mathrm{V}_{\langle\mathrm{NP}\rangle}\ \mathrm{NP} & \mathrm{V}_{\langle\rangle}\rightarrow\mathrm{ate} & \mathrm{V}_{\langle\mathrm{NP}\rangle}\rightarrow\mathrm{placed} \\
\mathrm{VP}\rightarrow\mathrm{V}_{\langle\mathrm{NP,PP}\rangle}\ \mathrm{NP}\ \mathrm{PP} & \mathrm{V}_{\langle\mathrm{NP}\rangle}\rightarrow\mathrm{ate} & \mathrm{V}_{\langle\mathrm{NP,PP}\rangle}\rightarrow\mathrm{put}
\end{array}$$

- Unlike arguments, **adjuncts** like *recently*, *with abandon*, and *at school* can combine very freely with verbs, so we also need rules like:

$$\begin{array}{l}
\mathrm{VP}\rightarrow\mathrm{V}_{\langle\rangle}\ \mathrm{PP} \\
\mathrm{VP}\rightarrow\mathrm{V}_{\langle\mathrm{NP}\rangle}\ \mathrm{NP}\ \mathrm{AdvP} \\
\mathrm{VP}\rightarrow\mathrm{V}_{\langle\mathrm{NP,PP}\rangle}\ \mathrm{NP}\ \mathrm{PP}\ \mathrm{AdvP}
\end{array}$$

Agreement & argument structure in CFGs

- Agreement, argument structure, and other constraints can be built into context-free grammars by adding **features** to phrasal categories, e.g.:

$$
\begin{array}{lll}
\text{VP} \rightarrow \text{V}_{\langle\rangle} & \text{V}_{\langle\rangle} \rightarrow \text{slept} & \text{V}_{\langle \text{NP} \rangle} \rightarrow \text{devoured} \\
\text{VP} \rightarrow \text{V}_{\langle \text{NP} \rangle}\ \text{NP} & \text{V}_{\langle\rangle} \rightarrow \text{ate} & \text{V}_{\langle \text{NP} \rangle} \rightarrow \text{placed} \\
\text{VP} \rightarrow \text{V}_{\langle \text{NP,PP} \rangle}\ \text{NP PP} & \text{V}_{\langle \text{NP} \rangle} \rightarrow \text{ate} & \text{V}_{\langle \text{NP,PP} \rangle} \rightarrow \text{put}
\end{array}
$$

- Unlike arguments, **adjuncts** like *recently*, *with abandon*, and *at school* can combine very freely with verbs, so we also need rules like:

$$
\begin{array}{l}
\text{VP} \rightarrow \text{V}_{\langle\rangle}\ \text{PP} \\
\text{VP} \rightarrow \text{V}_{\langle \text{NP} \rangle}\ \text{NP AdvP} \\
\text{VP} \rightarrow \text{V}_{\langle \text{NP,PP} \rangle}\ \text{NP PP AdvP}
\end{array}
$$

- (**Note:** it's widely agreed that there is *some* distinction between arguments and adjuncts, and many cases are clear, but some other cases are harder to classify!)

Agreement & argument structure in CFGs

- Agreement, argument structure, and other constraints can be built into context-free grammars by adding **features** to phrasal categories, e.g.:

$$\begin{array}{lll}
\mathrm{VP} \rightarrow \mathrm{V}_{\langle\rangle} & \mathrm{V}_{\langle\rangle} \rightarrow \text{slept} & \mathrm{V}_{\langle \mathrm{NP} \rangle} \rightarrow \text{devoured} \\
\mathrm{VP} \rightarrow \mathrm{V}_{\langle \mathrm{NP} \rangle}\ \mathrm{NP} & \mathrm{V}_{\langle\rangle} \rightarrow \text{ate} & \mathrm{V}_{\langle \mathrm{NP} \rangle} \rightarrow \text{placed} \\
\mathrm{VP} \rightarrow \mathrm{V}_{\langle \mathrm{NP},\mathrm{PP} \rangle}\ \mathrm{NP}\ \mathrm{PP} & \mathrm{V}_{\langle \mathrm{NP} \rangle} \rightarrow \text{ate} & \mathrm{V}_{\langle \mathrm{NP},\mathrm{PP} \rangle} \rightarrow \text{put}
\end{array}$$

- Unlike arguments, **adjuncts** like *recently*, *with abandon*, and *at school* can combine very freely with verbs, so we also need rules like:

$$\begin{array}{l}
\mathrm{VP} \rightarrow \mathrm{V}_{\langle\rangle}\ \mathrm{PP} \\
\mathrm{VP} \rightarrow \mathrm{V}_{\langle \mathrm{NP} \rangle}\ \mathrm{NP}\ \mathrm{AdvP} \\
\mathrm{VP} \rightarrow \mathrm{V}_{\langle \mathrm{NP},\mathrm{PP} \rangle}\ \mathrm{NP}\ \mathrm{PP}\ \mathrm{AdvP}
\end{array}$$

- (**Note:** it's widely agreed that there is *some* distinction between arguments and adjuncts, and many cases are clear, but some other cases are harder to classify!)
- **Recommended exercise:** work out how to use features to start build agreement constraints into a context-free grammar for English

Syntax and meaning composition

S →NP VP	NP→NP PP	VP→V	VP→V PP
NP→Det N	PP→P NP	VP→V NP	VP→V NP PP

- **Principle of compositionality:** the meaning of a complex expression derives from the meanings of the parts and the rules for combination

Syntax and meaning composition

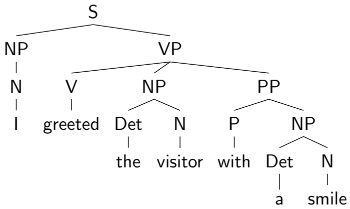
S →NP VP
NP→Det N
NP→NP PP
PP→P NP
VP→V
VP→V NP
VP→V PP
VP→V NP PP

- **Principle of compositionality:** the meaning of a complex expression derives from the meanings of the parts and the rules for combination
- **Relation with syntax:** for the most part (and possibly entirely!), meaning composition occurs via the hierarchy of the syntactic tree.

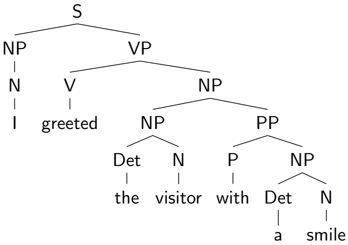
Syntax and meaning composition

S →NP VP NP→NP PP VP→V VP→V PP
NP→Det N PP→P NP VP→V NP VP→V NP PP

- **Principle of compositionality:** the meaning of a complex expression derives from the meanings of the parts and the rules for combination
- **Relation with syntax:** for the most part (and possibly entirely!), meaning composition occurs via the hierarchy of the syntactic tree.
- This means that syntax immediately gives us an account of many meaning ambiguities. For example:



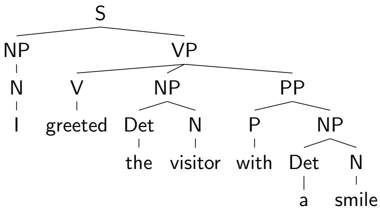
“I smiled as I greeted the visitor”

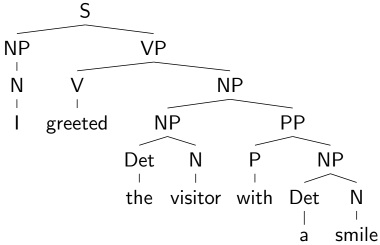


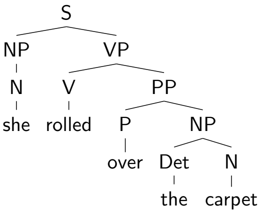
“I greeted the visitor that was smiling”

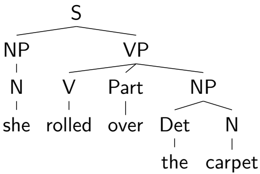
The phrase structure–prosody interface

- Phrasal boundaries (N**P**, V**P**, P**P**) often correspond with prosodic breaks (Lehiste et al., 1976):

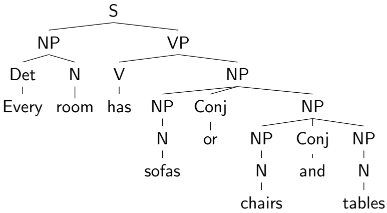


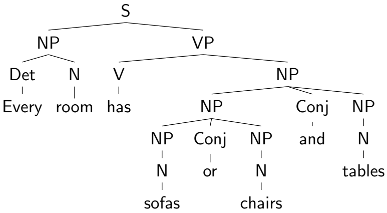


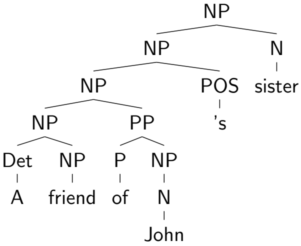


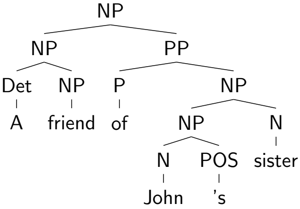


The phrase structure–prosody interface









Revisiting problem case studies for finite-state models

- We previously identified several problem cases in English syntax for finite-state models, including:

Revisiting problem case studies for finite-state models

- We previously identified several problem cases in English syntax for finite-state models, including:
  - Multiple PP attachment

```
a joke
a joke about the woman
a joke about the woman with an umbrella
a joke about the woman with an umbrella on the street
```

Revisiting problem case studies for finite-state models

- We previously identified several problem cases in English syntax for finite-state models, including:
  - Multiple PP attachment

```
a joke
a joke about the woman
a joke about the woman with an umbrella
a joke about the woman with an umbrella on the street
```

  - Nested *if/then* sentences

```
if students work hard, then they generally do well in class.
if if students work hard, then they generally do well in class, then the teacher is rewarded.
if if if students work hard, then they generally do well in class, then the teacher is rewarded,
then the university is well-run.
```

Revisiting problem case studies for finite-state models

- We previously identified several problem cases in English syntax for finite-state models, including:
  - Multiple PP attachment

```
a joke
a joke about the woman
a joke about the woman with an umbrella
a joke about the woman with an umbrella on the street
```

  - Nested *if/then* sentences

```
if students work hard, then they generally do well in class.
if if students work hard, then they generally do well in class, then the teacher is rewarded.
if if if students work hard, then they generally do well in class, then the teacher is rewarded,
then the university is well-run.
```

  - Multiply center-embedded object-extracted relative clauses

```
the rock can be found in the garden.
the rock that the squirrel likes can be found in the garden.
the rock that the squirrel that the dog chases likes can be found in the garden.
the rock that the squirrel that the dog that the woman owns chases likes can be found in the
garden.
```

Revisiting problem case studies for finite-state models

- We previously identified several problem cases in English syntax for finite-state models, including:
  - Multiple PP attachment

    `a joke`
    `a joke about the woman`
    `a joke about the woman with an umbrella`
    `a joke about the woman with an umbrella on the street`

  - Nested *if/then* sentences

    `if students work hard, then they generally do well in class.`
    `if if students work hard, then they generally do well in class, then the teacher is rewarded.`
    `if if if students work hard, then they generally do well in class, then the teacher is rewarded, then the university is well-run.`

  - Multiply center-embedded object-extracted relative clauses

    `the rock can be found in the garden.`
    `the rock that the squirrel likes can be found in the garden.`
    `the rock that the squirrel that the dog chases likes can be found in the garden.`
    `the rock that the squirrel that the dog that the woman owns chases likes can be found in the garden.`

- We will now revisit each case with context-free grammars

Multiple PP attachment

S → NP VP	Det → the
NP → Det N	N → dog
NP → NP PP	N → cat
PP → P NP	P → near
VP → V	V → growled

We just add the required lexical rewrite rules to the grammar we already specified, and we're all set!

Multiple PP attachment

S →NP VP
NP→Det N
NP→NP PP
PP→P NP
VP→V

Det→ the
N → dog
N → cat
P → near
V → growled

We just add the required lexical rewrite rules to the grammar we already specified, and we're all set!

Multiple PP attachment

S →NP VP
NP→Det N
NP→NP PP
PP→P NP
VP→V

Det→ the
N → dog
N → cat
P → near
V → growled

Det→a
Det→an
N →joke
N →woman
N →umbrella

N→ street
P→ about
P→ with
P→ on

We just add the required lexical rewrite rules to the grammar we already specified, and we're all set!

Multiple PP attachment

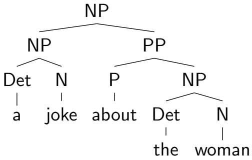
S → NP VP
NP → Det N
NP → NP PP
PP → P NP
VP → V

Det → the
N → dog
N → cat
P → near
V → growled

Det → a
Det → an
N → joke
N → woman
N → umbrella

N → street
P → about
P → with
P → on

We just add the required lexical rewrite rules to the grammar we already specified, and we're all set!



Multiple PP attachment

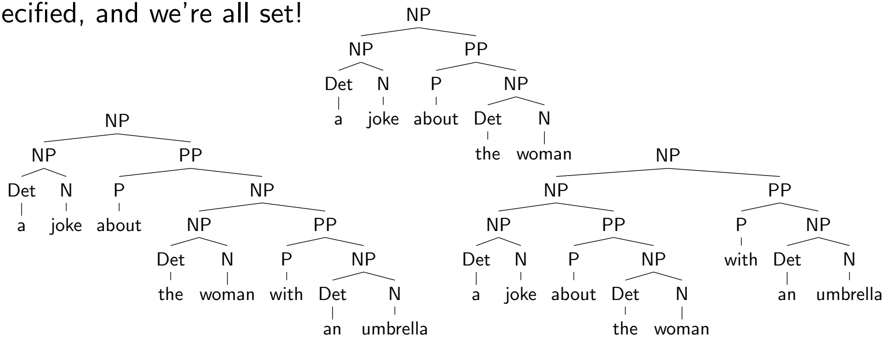
S →NP VP
NP→Det N
NP→NP PP
PP→P NP
VP→V

Det→ the
N → dog
N → cat
P → near
V → growled

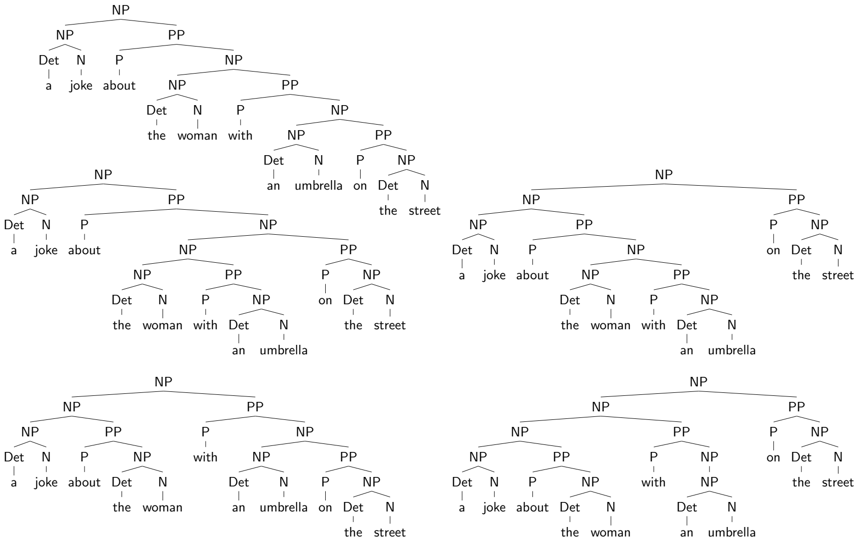
Det→a
Det→an
N →joke
N →woman
N →umbrella

N→ street
P→ about
P→ with
P→ on

We just add the required lexical rewrite rules to the grammar we already specified, and we're all set!



Multiple PP attachment



⋮

Nested *if/then* sentences

S → NP VP
VP → V NP
S → If S then S

NP → dogs
NP → cats
NP → mice

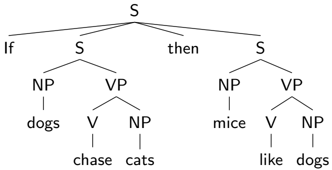
V → chase
V → watch
V → seek

Nested *if/then* sentences

S → NP VP
VP → V NP
S → If S then S

NP → dogs
NP → cats
NP → mice

V → chase
V → watch
V → seek



Nested *if/then* sentences

S $\rightarrow$ NP VP

VP $\rightarrow$ V NP

S $\rightarrow$ If S then S

NP $\rightarrow$ dogs

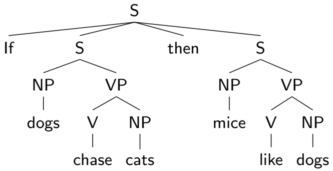
NP $\rightarrow$ cats

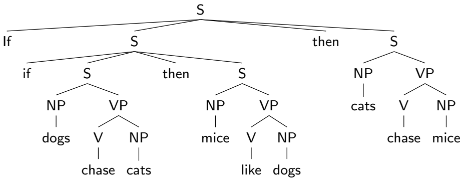
NP $\rightarrow$ mice

V $\rightarrow$ chase

V $\rightarrow$ watch

V $\rightarrow$ seek



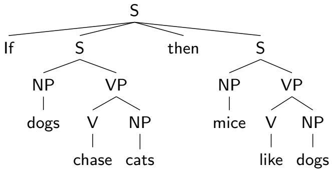


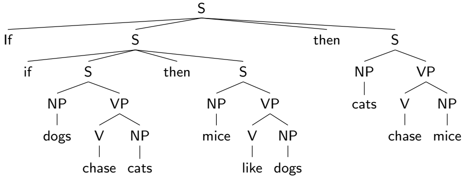
Nested *if/then* sentences

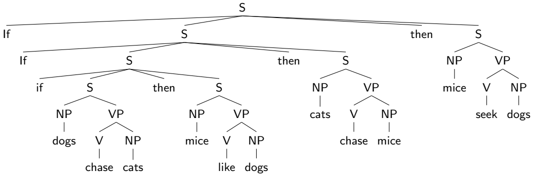
S → NP VP
VP → V NP
S → If S then S

NP → dogs
NP → cats
NP → mice

V → chase
V → watch
V → seek







⋮

References: general formal language theory I

Aho, A. V. (1990). Algorithms for finding patterns in strings. In *Handbook of theoretical computer science* (pp. 255–300).

Harrison, M. A. (1978). *Introduction to formal language theory*. Addison-Wesley Longman Publishing Co., Inc.

Jäger, G., & Rogers, J. (2012). Formal language theory: Refining the chomsky hierarchy. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *367*(1598), 1956–1970.

Kornai, A. (2007). *Mathematical linguistics*. Springer Science & Business Media.

Sipser, M. (2012). *Introduction to the theory of computation*. Cengage learning.

References: finite-state automata and transducers I

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., & Mohri, M. (2007). OpenFst: A general and efficient weighted finite-state transducer library [http://www.openfst.org]. In *Proceedings of the ninth international conference on implementation and application of automata, (ciaa 2007)*, Springer. http://www.openfst.org.

Beesley, K. R., & Karttunen, L. (2003). *Finite-state morphology: Xerox tools and techniques*. CSLI, Stanford.

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational Linguistics*, *23*(2), 269–311.

Mohri, M., Pereira, F., & Riley, M. (2002). Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, *16*(1), 69–88.

Roche, E., & Schabes, Y. (1997). *Finite-state language processing*. MIT Press.

References: phonotactics I

Futrell, R., Albright, A., Graff, P., & O'Donnell, T. J. (2017). A generative model of phonotactics. *Transactions of the Association for Computational Linguistics*, *5*, 73–86.

Hayes, B. (2011). *Introductory phonology* (Vol. 32). John Wiley & Sons.

Hayes, B., & Wilson, C. (2007). A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry*, *39*, 379–440.

Heinz, J. (2010). Learning long-distance phonotactics. *Linguistic Inquiry*, *41*(4), 623–661.

McQueen, J. M. (1998). Segmentation of continuous speech using phonotactics. *Journal of Memory and Language*, *39*(1), 21–46.

References: weak and strong generative capacity I

Bresnan, J., Kaplan, R. M., Peters, S., & Zaenen, A. (1982). Cross-serial dependencies in dutch. In *The formal complexity of natural language* (pp. 286–319). Springer.

Chomsky, N., & Miller, G. A. (1963). Introduction to the formal analysis of natural languages. In R. D. Luce, R. R. Bush, & E. Galanter (Eds.), *Handbook of mathematical psychology* (pp. 269–321). New York: John Wiley & Sons, Inc.

Kornai, A., & Pullum, G. K. (1990). The x-bar theory of phrase structure. *Language*, 24–50.

Miller, P. (2000). *Strong generative capacity: The semantics of linguistic formalism*. Cambridge.

Vijay-Shanker, K., & Weir, D. J. (1994). The equivalence of four extensions of context-free grammar. *Mathematical Systems Theory*, *27*(6), 511–546.

References: limits of finite-state models for natural language I

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, *2*(3), 113–124.

Karlsson, F. (2007). Constraints on multiple center-embedding of clauses. *Journal of Linguistics*, 365–392.

Kornai, A. (1985). Natural languages and the Chomsky hierarchy. In *Second conference of the European chapter of the association for computational linguistics*, Geneva, Switzerland, Association for Computational Linguistics.

Miller, G. A., & Chomsky, N. (1963). Finitary models of language users. In R. D. Luce, R. R. Bush, & E. Galanter (Eds.), *Handbook of mathematical psychology* (pp. 419–491). New York: John Wiley & Sons, Inc.