Context-free grammars and syntactic analysis

Roger Levy

Massachusetts Institute of Technology
Department of Brain & Cognitive Sciences

Nested object-extracted relative clauses

- The following pattern was beyond even the *weak* generative capacity of finite-state models:

```
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.
```

Nested object-extracted relative clauses

- The following pattern was beyond even the *weak* generative capacity of finite-state models:
  `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.`
- This pattern involves $N^i V^i$ "matched-pair" nouns and verbs

Nested object-extracted relative clauses

- The following pattern was beyond even the *weak* generative capacity of finite-state models:

  `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.`

- This pattern involves $N^i V^i$ "matched-pair" nouns and verbs
- **Insight:** the extraction property of relative clauses implies that certain phrasal categories inside relative clauses (RCs) behave in exactly the ordinary way, *except that they are "missing an element"*.

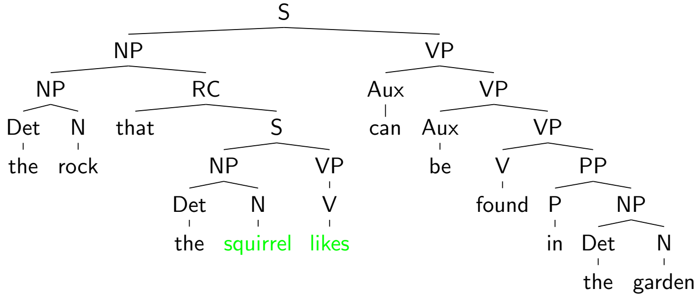
Nested object-extracted relative clauses

- The following pattern was beyond even the *weak* generative capacity of finite-state models:
  `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.`
- This pattern involves $N^i V^i$ "matched-pair" nouns and verbs
- **Insight:** the extraction property of relative clauses implies that certain phrasal categories inside relative clauses (RCs) behave in exactly the ordinary way, *except that they are "missing an element"*.
- This "missing an element" property must be formally represented in the structure of the grammar in order for the grammar not to OVERGENERATE.

The nature of extraction

- For example, the CFG below (some terminal rewrites for Det, N, and V omitted for brevity) would generate the required object-extracted relative clauses sentences:

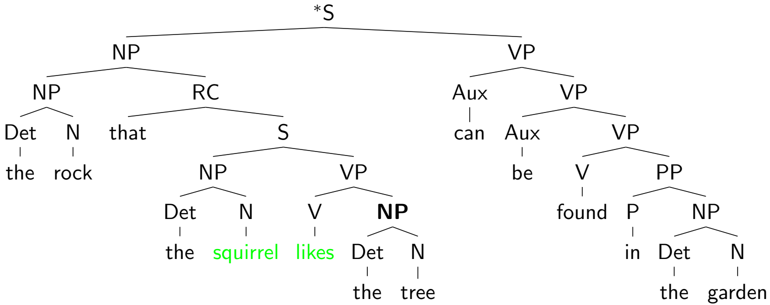
S → NP VP	RC → that S	VP → VP PP
NP → Det N	VP → V NP	VP → V
NP → NP RC	VP → Aux VP	PP → P NP



The nature of extraction

S → NP VP	RC → that S	VP → VP PP
NP → Det N	VP → V NP	VP → V
NP → NP RC	VP → Aux VP	PP → P NP

- But it would also **overgenerate**, allowing cases like the below where the "missing an element" property of the RC is broken by the appearance of *the tree* as the direct object of *likes*:



Wh-questions also involve extraction

- *Wh*-questions also are "missing an element" ("___"=an empty string that would have to have something there if not for the special environment)

Wh-questions also involve extraction

- *Wh*-questions also are "missing an element" ("___"=an empty string that would have to have something there if not for the special environment)

Wh-questions also involve extraction

- *Wh*-questions also are "missing an element" ("___"=an empty string that would have to have something there if not for the special environment)

(1) Who did Kim invite ___ to the party?

Wh-questions also involve extraction

- *Wh*-questions also are "missing an element" ("___"=an empty string that would have to have something there if not for the special environment)

(1) Who did Kim invite ___ to the party?

(2) *Who did Kim invite the neighbors to the party?

Wh-questions also involve extraction

- *Wh*-questions also are "missing an element" ("___"=an empty string that would have to have something there if not for the special environment)

(1) Who did Kim invite ___ to the party?

(2) *Who did Kim invite the neighbors to the party?

(3) Who did you say ___ invited you to the party?

(4) *Who did you say the neighbors invited you to the party?

Wh-questions also involve extraction

- *Wh*-questions also are "missing an element" ("___"=an empty string that would have to have something there if not for the special environment)

(1) Who did Kim invite ___ to the party?

(2) *Who did Kim invite the neighbors to the party?

(3) Who did you say ___ invited you to the party?

(4) *Who did you say the neighbors invited you to the party?

(5) What did you say you ate ___ at the party?

(6) *What did you say you ate food at the party?

(7) What did you say Kim told you that Pat ate ___ at the party?

(8) *What did you say Kim told you that Pat ate food at the party?

- This "missing element" property in RCs and *wh*-questions is called **extraction**.

Extraction is an unbounded-depth dependency

- Extraction can span an unbounded number of levels of clausal embedding:
  `Who did Kim invite ___ to the party?`
  `Who did you say [Kim invited ___ to the party]?`
  `Who did you say [Pat suspected [ Kim invited ___ to the party]]?`
  `Who did you say [Terry texted that [Pat suspected [Kim invited ___ to the party]]]?`
  ⋮

Extraction is an unbounded-depth dependency

- Extraction can span an unbounded number of levels of clausal embedding:

  `Who did Kim invite ___ to the party?`
  `Who did you say [Kim invited ___ to the party]?`
  `Who did you say [Pat suspected [ Kim invited ___ to the party]]?`
  `Who did you say [Terry texted that [Pat suspected [Kim invited ___ to the party]]]?`

  ⋮

Extraction is an unbounded-depth dependency

- Extraction can span an unbounded number of levels of clausal embedding:

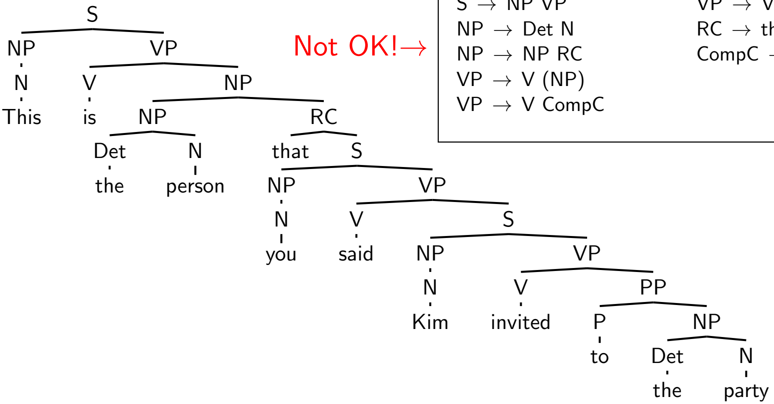
  `Who did Kim invite ___ to the party?`
  `Who did you say [Kim invited ___ to the party]?`
  `Who did you say [Pat suspected [ Kim invited ___ to the party]]?`
  `Who did you say [Terry texted that [Pat suspected [Kim invited ___ to the party]]]?`

  ⋮

  `This is the person that [Kim invited ___ to the party].`
  `This is the person that [you said [Kim invited ___ to the party]].`
  `This is the person that [you said [Pat suspected [Kim invited ___ to the party]]].`
  `This is the person that [you said [Terry texted that [Pat suspected [Kim invited ___ to the party]]]].`

  ⋮

Unbounded dependency constructions

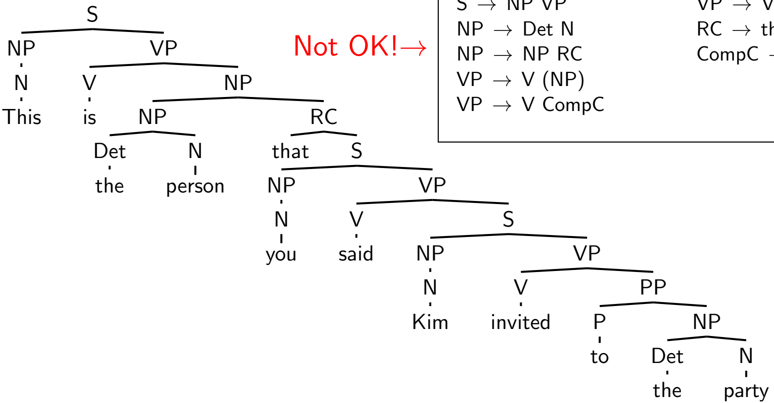


Not OK!→

S → NP VP
NP → Det N
NP → NP RC
VP → V (NP)
VP → V CompC
VP → V (NP) PP
RC → that S
CompC → (that) S

- The simple categories in the grammar above fail to "remember" that there is an extraction once we are inside the RC

Unbounded dependency constructions



Not OK!→

S → NP VP
NP → Det N
NP → NP RC
VP → V (NP)
VP → V CompC
VP → V (NP) PP
RC → that S
CompC → (that) S

- The simple categories in the grammar above fail to “remember” that there is an extraction once we are inside the RC
- Thus, the grammar wouldn’t rule out adding an object NP after *invited*, like it should

Unbounded dependency constructions

- These UNBOUNDED DEPENDENCIES were part (though not all) of the motivation originally given by Chomsky (1956) for a TRANSFORMATIONAL GRAMMAR that went beyond the expressive capabilities of context-free grammars

Unbounded dependency constructions

- These UNBOUNDED DEPENDENCIES were part (though not all) of the motivation originally given by Chomsky (1956) for a TRANSFORMATIONAL GRAMMAR that went beyond the expressive capabilities of context-free grammars
- However, it turns out that we can incorporate long-distance dependency constraints within the context-free formalism to avoid this type of overgeneration, as shown by Gazdar (1981) and others

Unbounded dependency constructions

- These UNBOUNDED DEPENDENCIES were part (though not all) of the motivation originally given by Chomsky (1956) for a TRANSFORMATIONAL GRAMMAR that went beyond the expressive capabilities of context-free grammars
- However, it turns out that we can incorporate long-distance dependency constraints within the context-free formalism to avoid this type of overgeneration, as shown by Gazdar (1981) and others
- A theoretical innovation of GENERALIZED PHRASE STRUCTURE GRAMMAR (Gazdar et al., 1985) was to introduce METARULES stating implicational relationships between the presence of certain types of categories & rules and certain other types of categories & rules.

Unbounded dependency constructions

- For unbounded *wh*- and relative-clause dependencies, we use the following metarules:

Unbounded dependency constructions

- For unbounded *wh*- and relative-clause dependencies, we use the following metarules:
  1. For a CFG with non-terminal inventory N and rule set R, distinguish a **basic** set of non-terminal symbol $N_{\text{basic}} \subset N$ and a **basic** set of rules $R_{\text{basic}} \subset R$.

Unbounded dependency constructions

- For unbounded *wh*- and relative-clause dependencies, we use the following metarules:
  1. For a CFG with non-terminal inventory N and rule set R, distinguish a **basic** set of non-terminal symbol $N_{\text{basic}} \subset N$ and a **basic** set of rules $R_{\text{basic}} \subset R$.
  2. For every category pair $X, Y \in N_{\text{basic}}$, a **derived** non-terminal symbol X/Y must be in N. X/Y can be interpreted as "an X that is missing a Y inside".

Unbounded dependency constructions

- For unbounded *wh-* and relative-clause dependencies, we use the following metarules:
  1. For a CFG with non-terminal inventory N and rule set R, distinguish a **basic** set of non-terminal symbol $N_{\text{basic}} \subset N$ and a **basic** set of rules $R_{\text{basic}} \subset R$.
  2. For every category pair $X, Y \in N_{\text{basic}}$, a **derived** non-terminal symbol X/Y must be in N. X/Y can be interpreted as "an X that is missing a Y inside".
  3. For every basic rule $X \rightarrow \alpha_1 \ldots \alpha_n \in R_{\text{basic}}$ and basic category Y in N, then for all $1 \leq i \leq n$ a **derived** rule $X/Y \rightarrow \alpha_1 \ldots \alpha_i/Y \ldots \alpha_n$ must be in R.

Unbounded dependency constructions

- For unbounded *wh*- and relative-clause dependencies, we use the following metarules:
  1. For a CFG with non-terminal inventory N and rule set R, distinguish a **basic** set of non-terminal symbol $N_{\text{basic}} \subset N$ and a **basic** set of rules $R_{\text{basic}} \subset R$.
  2. For every category pair $X, Y \in N_{\text{basic}}$, a **derived** non-terminal symbol X/Y must be in N. X/Y can be interpreted as "an X that is missing a Y inside".
  3. For every basic rule $X \to \alpha_1 \dots \alpha_n \in R_{\text{basic}}$ and basic category Y in N, then for all $1 \leq i \leq n$ a **derived** rule $X/Y \to \alpha_1 \dots \alpha_i/Y \dots \alpha_n$ must be in R.
  4. For every basic category X, a rule $X/X \to \epsilon$ must be in R.

Unbounded dependencies with metarules

1. For a CFG with non-terminal inventory N and rule set R, distinguish a **basic** set of non-terminal symbol $N_{\text{basic}} \subset N$ and a **basic** set of rules $R_{\text{basic}} \subset R$.
2. For every category pair $X, Y \in N_{\text{basic}}$, a **derived** non-terminal symbol X/Y must be in N. X/Y can be interpreted as "an X that is missing a Y inside".
3. For every basic rule $X \to \alpha_1 \ldots \alpha_n \in R_{\text{basic}}$ and basic category Y in N, then for all $1 \leq i \leq n$ a **derived** rule $X/Y \to \alpha_1 \ldots \alpha_i/Y \ldots \alpha_n$ must be in R.
4. For every basic category X, a rule $X/X \to \epsilon$ must be in R.

S	$\to$ NP VP
NP	$\to$ Det N
NP	$\to$ NP RC
VP	$\to$ V (NP)
VP	$\to$ V CompC
VP	$\to$ VP (NP) PP
CompC	$\to$ (that) S

Unbounded dependencies with metarules

1. For a CFG with non-terminal inventory N and rule set R, distinguish a **basic** set of non-terminal symbol $N_{\text{basic}} \subset N$ and a **basic** set of rules $R_{\text{basic}} \subset R$.
2. For every category pair $X, Y \in N_{\text{basic}}$, a **derived** non-terminal symbol X/Y must be in N. X/Y can be interpreted as "an X that is missing a Y inside".
3. For every basic rule $X \to \alpha_1 \dots \alpha_n \in R_{\text{basic}}$ and basic category Y in N, then for all $1 \leq i \leq n$ a **derived** rule $\boxed{X/Y \to \alpha_1 \dots \alpha_i/Y \dots \alpha_n}$ must be in R.
4. For every basic category X, a rule $X/X \to \epsilon$ must be in R.

S	→ NP VP
NP	→ Det N
NP	→ NP RC
VP	→ V (NP)
VP	→ V CompC
VP	→ VP (NP) PP
CompC	→ (that) S

- Basic categories: S, NP, VP, CompC

Unbounded dependencies with metarules

1. For a CFG with non-terminal inventory N and rule set R, distinguish a **basic** set of non-terminal symbol $N_{\text{basic}} \subset N$ and a **basic** set of rules $R_{\text{basic}} \subset R$.
2. For every category pair $X, Y \in N_{\text{basic}}$, a **derived** non-terminal symbol X/Y must be in N. X/Y can be interpreted as "an X that is missing a Y inside".
3. For every basic rule $X \to \alpha_1 \ldots \alpha_n \in R_{\text{basic}}$ and basic category Y in N, then for all $1 \leq i \leq n$ a **derived** rule $\boxed{X/Y \to \alpha_1 \ldots \alpha_i/Y \ldots \alpha_n}$ must be in R.
4. For every basic category X, a rule $X/X \to \epsilon$ must be in R.

S	→ NP VP
NP	→ Det N
NP	→ NP RC
VP	→ V (NP)
VP	→ V CompC
VP	→ VP (NP) PP
CompC	→ (that) S

- Basic categories: S, NP, VP, CompC
- Derived categories (showing only the relevant ones): S/NP, VP/NP, NP/NP, CompC/NP; and the corresponding derived rules

Unbounded dependencies with metarules

1. For a CFG with non-terminal inventory N and rule set R, distinguish a **basic** set of non-terminal symbol $N_{\text{basic}} \subset N$ and a **basic** set of rules $R_{\text{basic}} \subset R$.
2. For every category pair $X, Y \in N_{\text{basic}}$, a **derived** non-terminal symbol X/Y must be in N. X/Y can be interpreted as "an X that is missing a Y inside".
3. For every basic rule $X \to \alpha_1 \ldots \alpha_n \in R_{\text{basic}}$ and basic category Y in N, then for all $1 \leq i \leq n$ a **derived** rule $X/Y \to \alpha_1 \ldots \alpha_i/Y \ldots \alpha_n$ must be in R.
4. For every basic category X, a rule $X/X \to \epsilon$ must be in R.

S	→ NP VP
NP	→ Det N
NP	→ NP RC
VP	→ V (NP)
VP	→ V CompC
VP	→ VP (NP) PP
CompC	→ (that) S
S/NP	→ NP/NP VP
S/NP	→ NP VP/NP
NP/NP	→ ϵ
VP/NP	→ V/NP NP
VP/NP	→ V NP/NP
VP/NP	→ V/NP CompC
VP/NP	→ V CompC/NP
CompC/NP	→ (that) S/NP

- Basic categories: S, NP, VP, CompC
- Derived categories (showing only the relevant ones): S/NP, VP/NP, NP/NP, CompC/NP; and the corresponding derived rules

Unbounded dependencies with metarules

1. For a CFG with non-terminal inventory N and rule set R, distinguish a **basic** set of non-terminal symbol $N_{\text{basic}} \subset N$ and a **basic** set of rules $R_{\text{basic}} \subset R$.
2. For every category pair $X, Y \in N_{\text{basic}}$, a **derived** non-terminal symbol X/Y must be in N. X/Y can be interpreted as "an X that is missing a Y inside".
3. For every basic rule $X \to \alpha_1 \ldots \alpha_n \in R_{\text{basic}}$ and basic category Y in N, then for all $1 \leq i \leq n$ a **derived** rule $X/Y \to \alpha_1 \ldots \alpha_i/Y \ldots \alpha_n$ must be in R.
4. For every basic category X, a rule $X/X \to \epsilon$ must be in R.

S	$\to$	NP VP
NP	$\to$	Det N
NP	$\to$	NP RC
VP	$\to$	V (NP)
VP	$\to$	V CompC
VP	$\to$	VP (NP) PP
CompC	$\to$	(that) S
S/NP	$\to$	NP/NP VP
S/NP	$\to$	NP VP/NP
NP/NP	$\to$	ϵ
VP/NP	$\to$	V/NP NP
VP/NP	$\to$	V NP/NP
VP/NP	$\to$	V/NP CompC
VP/NP	$\to$	V CompC/NP
CompC/NP	$\to$	(that) S/NP
RC	$\to$	**that S/NP**

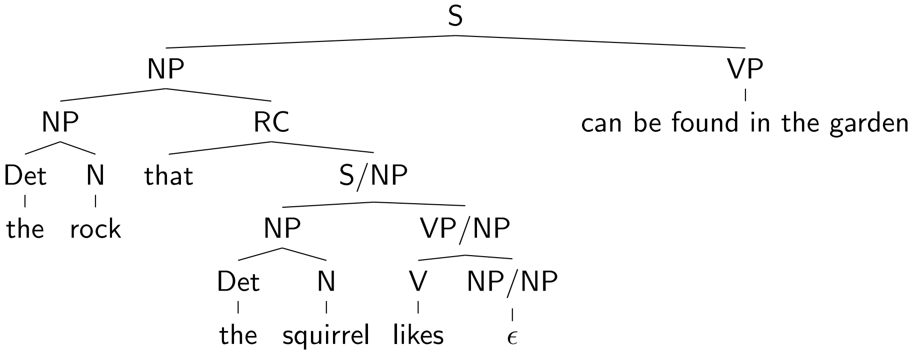
- Basic categories: S, NP, VP, CompC
- Derived categories (showing only the relevant ones): S/NP, VP/NP, NP/NP, CompC/NP; and the corresponding derived rules
- We can now define a relative clause as introducing the derived category **S/NP**!

Unbounded dependencies with metarules

S $\rightarrow$ NP VP
NP $\rightarrow$ Det N
NP $\rightarrow$ NP RC
VP $\rightarrow$ V (NP)
VP $\rightarrow$ V CompC
VP $\rightarrow$ VP (NP) PP

CompC $\rightarrow$ (that) S
S/NP $\rightarrow$ NP/NP VP
S/NP $\rightarrow$ NP VP/NP
NP/NP $\rightarrow$ ϵ
VP/NP $\rightarrow$ V/NP NP
VP/NP $\rightarrow$ V NP/NP

VP/NP $\rightarrow$ V/NP CompC
VP/NP $\rightarrow$ V CompC/NP
CompC/NP $\rightarrow$ (that) S/NP
RC $\rightarrow$ that S/NP

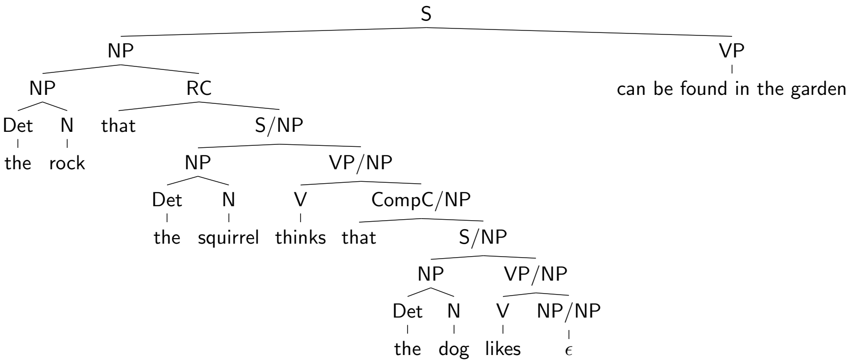


Unbounded dependencies with metarules

S → NP VP
NP → Det N
NP → NP RC
VP → V (NP)
VP → V CompC
VP → VP (NP) PP

CompC → (that) S
S/NP → NP/NP VP
S/NP → NP VP/NP
NP/NP → ϵ
VP/NP → V/NP NP
VP/NP → V NP/NP

VP/NP → V/NP CompC
VP/NP → V CompC/NP
CompC/NP → (that) S/NP
RC → that S/NP

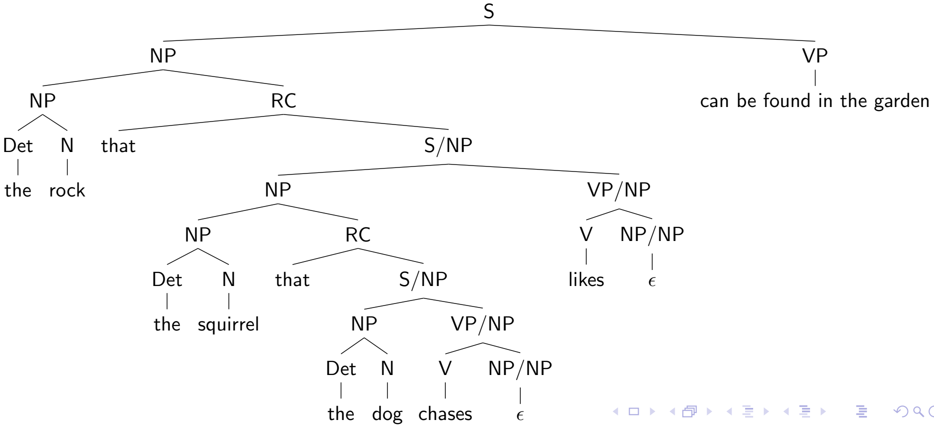


Unbounded dependencies with metarules

S → NP VP
NP → Det N
NP → NP RC
VP → V (NP)
VP → V CompC
VP → VP (NP) PP

CompC → (that) S
S/NP → NP/NP VP
S/NP → NP VP/NP
NP/NP → ϵ
VP/NP → V/NP NP
VP/NP → V NP/NP

VP/NP → V/NP CompC
VP/NP → V CompC/NP
CompC/NP → (that) S/NP
RC → that S/NP



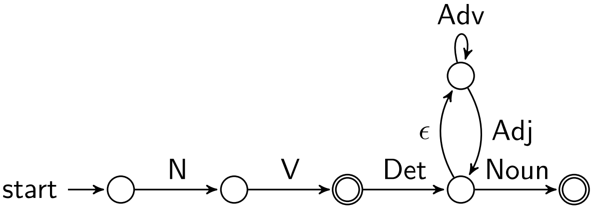
Three types of "long-distance dependency" I

We have now seen three types of "long-distance dependency" in language:

1. A pair of categories separable by an unbounded number of **tokens** (here, words):

   `I ate`
   `I ate a sandwich`
   `I ate a big sandwich`
   `I ate a very big, freshly prepared, extremely tasty sandwich`
   ⋮

   This type of dependency can be modeled with finite-state methods:



Three types of "long-distance dependency" II

2. A pair separable by **an unboundedly deep nesting of phrases**:

   `if students work hard, then they generally do well in class.`

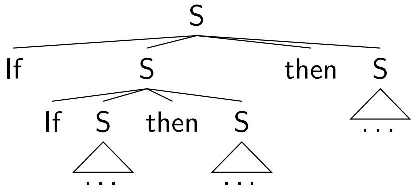
   `if it is the case that if students work hard, then they generally do well in class, then the teacher is rewarded.`

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

   ⋮

   This type of dependency requires context-free grammars:

   S → NP VP   VP → V NP   S → If (it is the case that) S then S

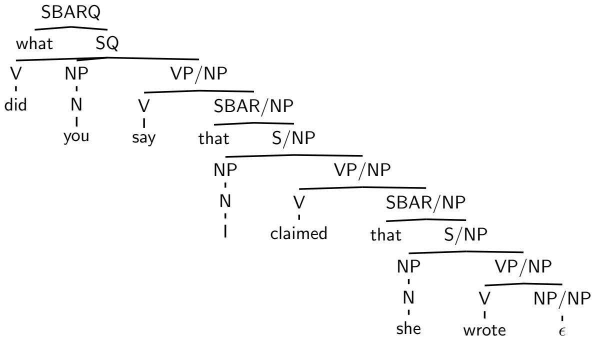


Three types of "long-distance dependency" III

3. A pair separable by **An unboundedly long chain of tree nodes**:

```
Who did Kim invite ___ to the party?
Who did you say [Kim invited ___ to the party]?
Who did you say [Pat suspected Kim invited ___ to the party]]?
Who did you say [Terry texted that [Pat suspected [Kim invited ___ to the party]]]?
⋮
```

This type of dependency can be modeled with context-free grammars that have **feature passing** through the categories:



Context-free languages and their closure properties

- The **context-free languages** are the set of languages that can be characterized by a context-free grammar

Context-free languages and their closure properties

- The **context-free languages** are the set of languages that can be characterized by a context-free grammar
- Like the regular languages, the context-free languages are closed under:

Context-free languages and their closure properties

- The **context-free languages** are the set of languages that can be characterized by a context-free grammar
- Like the regular languages, the context-free languages are closed under:
  - **union**, **concatenation**, and **Kleene closure**

Context-free languages and their closure properties

- The **context-free languages** are the set of languages that can be characterized by a context-free grammar
- Like the regular languages, the context-free languages are closed under:
  - **union**, **concatenation**, and **Kleene closure**
- The context-free languages are also closed under **intersection with a regular language**. If L is context-free and R is regular, then $L \cap R$ is context-free.

Context-free languages and their closure properties

- The **context-free languages** are the set of languages that can be characterized by a context-free grammar
- Like the regular languages, the context-free languages are closed under:
  - **union**, **concatenation**, and **Kleene closure**
- The context-free languages are also closed under **intersection with a regular language**. If L is context-free and R is regular, then $L \cap R$ is context-free.
- But unlike the regular languages, the context-free languages are *not* closed under intersection: if L_1 and L_2 are context-free, then $L_1 \cap L_2$ is not necessarily context-free.

Normal forms for grammars

- Any given context-free language will have multiple (in fact, infinitely many!) context-free grammars that generate it

Normal forms for grammars

- Any given context-free language will have multiple (in fact, infinitely many!) context-free grammars that generate it
- Various NORMAL FORMS pose constraints on the structure of a grammar's rules

Normal forms for grammars

- Any given context-free language will have multiple (in fact, infinitely many!) context-free grammars that generate it
- Various NORMAL FORMS pose constraints on the structure of a grammar's rules
- Of particular interest for us is CHOMSKY NORMAL FORM (CNF), in which all rules take one of the following three forms (S: the start symbol; A, B, and C: non-terminals; X: a terminal):
  - S $\rightarrow$ ϵ (the grammar generates the empty string)
  - A $\rightarrow$ B C (binary non-terminal rewrite)
  - A $\rightarrow$ x (unary terminal rewrite)

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.