61A Lecture 37

Wednesday, April 29

Announcements

Announcements

- Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm

Announcements

- Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
- Quiz 4 due Thursday 4/30 @ 11:59pm

Announcements

- Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
- Quiz 4 due Thursday 4/30 @ 11:59pm
- No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)

Announcements

- Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
- Quiz 4 due Thursday 4/30 @ 11:59pm
- No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)
  - If at least 60% of students respond, everyone gets an extra credit point

Announcements

- Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
- Quiz 4 due Thursday 4/30 @ 11:59pm
- No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)
  - If at least 60% of students respond, everyone gets an extra credit point
- Next week: 18 hours of review sessions Monday, Tuesday, & Wednesday 11–5 in 271/273 Soda

Announcements

- Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
- Quiz 4 due Thursday 4/30 @ 11:59pm
- No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)
  - If at least 60% of students respond, everyone gets an extra credit point
- Next week: 18 hours of review sessions Monday, Tuesday, & Wednesday 11–5 in 271/273 Soda
  - Two TAs are available every hour

Announcements

- Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
- Quiz 4 due Thursday 4/30 @ 11:59pm
- No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)
  - If at least 60% of students respond, everyone gets an extra credit point
- Next week: 18 hours of review sessions Monday, Tuesday, & Wednesday 11–5 in 271/273 Soda
  - Two TAs are available every hour
  - One room will be a review session going over topic-specific problems

Announcements

- Homework 9 (4 pts) due Wednesday 4/29 @ 11:59pm
- Quiz 4 due Thursday 4/30 @ 11:59pm
- No videos on Friday 5/1; Come to lecture (and fill out the HKN course survey at the end)
  - If at least 60% of students respond, everyone gets an extra credit point
- Next week: 18 hours of review sessions Monday, Tuesday, & Wednesday 11–5 in 271/273 Soda
  - Two TAs are available every hour
  - One room will be a review session going over topic-specific problems
  - The other room is unstructured; staff will answer any questions you have

Ambiguity

Syntactic Ambiguity in English

Programs must be written for people to read

Syntactic Ambiguity in English

Programs must be written for people to read[1]

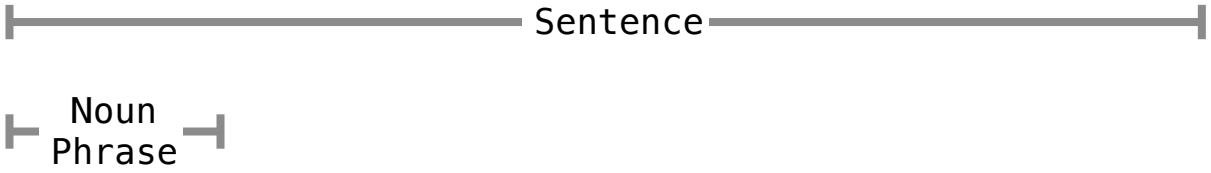
[1]Preface of **Structure and Interpretation of Computer Programs**
by Harold Abelson and Gerald Sussman with Julie Sussman

Syntactic Ambiguity in English

Programs must be written for people to read[1]

[1]Preface of **Structure and Interpretation of Computer Programs**
by Harold Abelson and Gerald Sussman with Julie Sussman

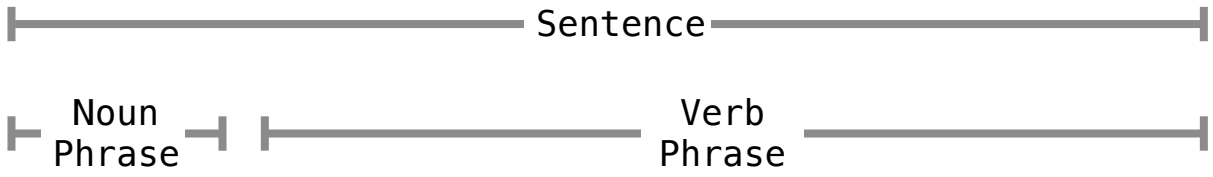
Syntactic Ambiguity in English



Programs must be written for people to read[1]

[1]Preface of **Structure and Interpretation of Computer Programs**
by Harold Abelson and Gerald Sussman with Julie Sussman

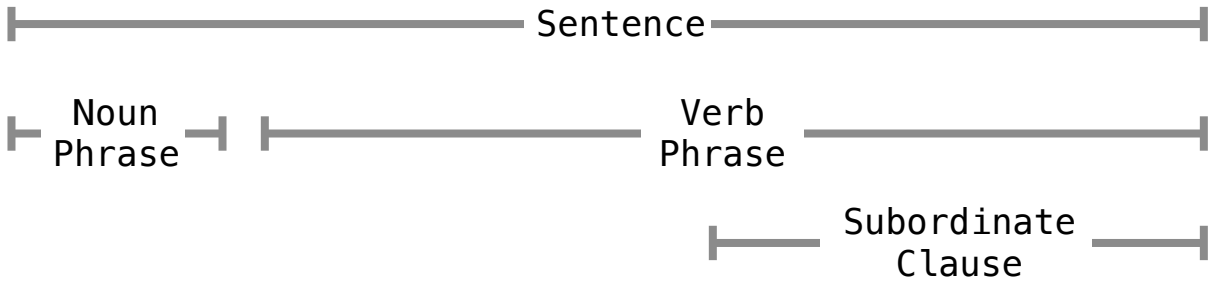
Syntactic Ambiguity in English



Programs must be written for people to read[1]

[1]Preface of **Structure and Interpretation of Computer Programs**
by Harold Abelson and Gerald Sussman with Julie Sussman

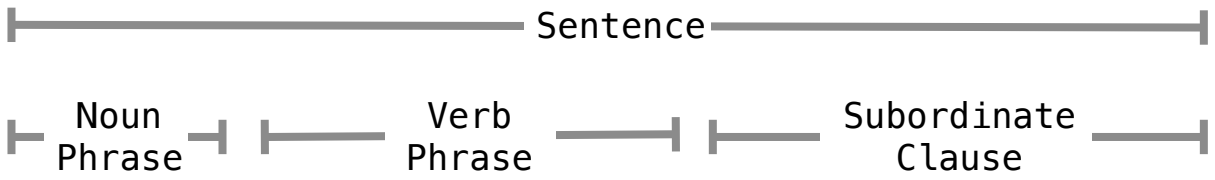
Syntactic Ambiguity in English



Programs must be written for people to read[1]

[1]Preface of **Structure and Interpretation of Computer Programs**
by Harold Abelson and Gerald Sussman with Julie Sussman

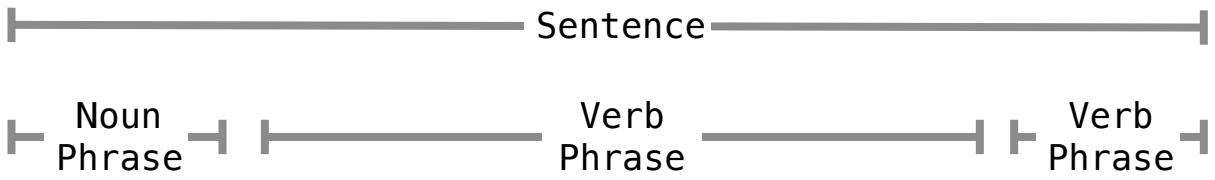
Syntactic Ambiguity in English



Programs must be written for people to read[1]

[1]Preface of **Structure and Interpretation of Computer Programs**
by Harold Abelson and Gerald Sussman with Julie Sussman

Syntactic Ambiguity in English



Programs must be written for people to read[1]

[1]Preface of **Structure and Interpretation of Computer Programs**
by Harold Abelson and Gerald Sussman with Julie Sussman

Syntactic Ambiguity in English

pro•gram (noun)
  a series of coded software instructions

pro•gram (verb)
  provide a computer with coded instructions

Programs must be written for people to read

must (verb)
  be obliged to

must (noun)
  dampness or mold

Definitions from the New Oxford American Dictionary

Syntax Trees

Representing Syntactic Structure

Representing Syntactic Structure

cows

Representing Syntactic Structure

Representing Syntactic Structure

Noun

cows

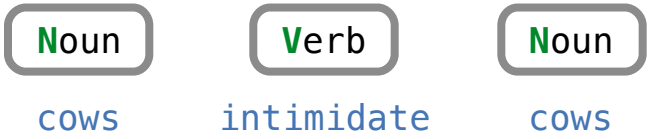
Representing Syntactic Structure

Noun

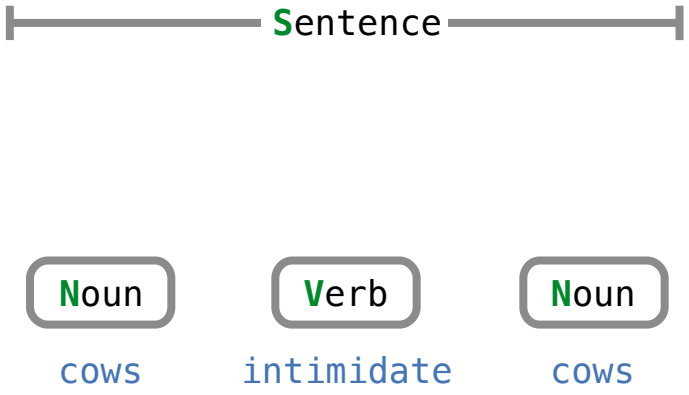
cows intimidate

Representing Syntactic Structure

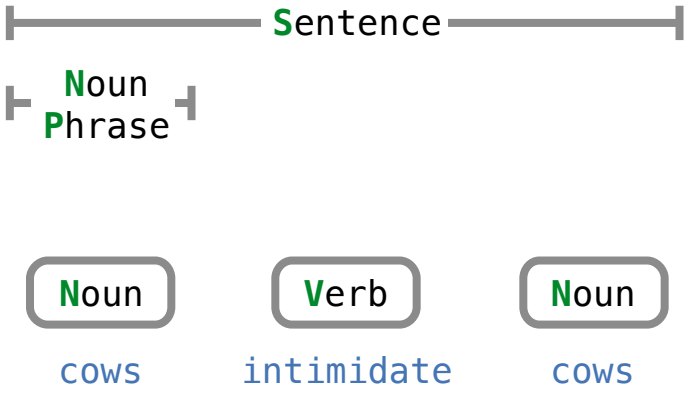
Representing Syntactic Structure



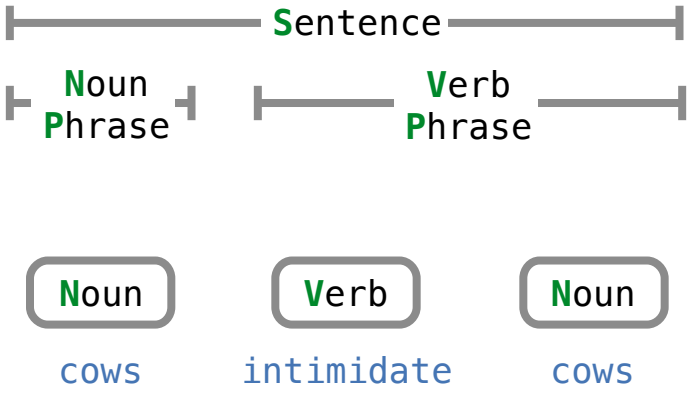
Representing Syntactic Structure



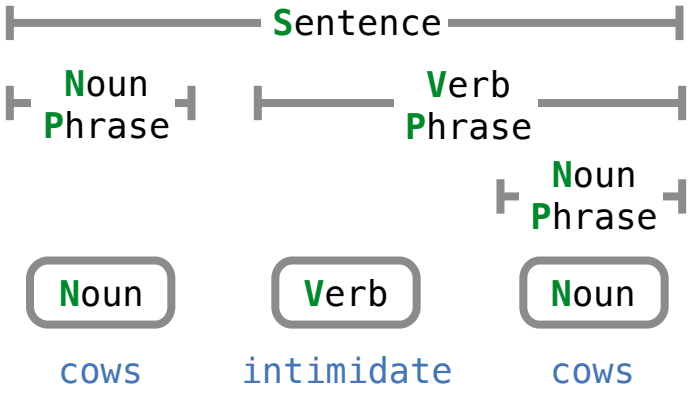
Representing Syntactic Structure



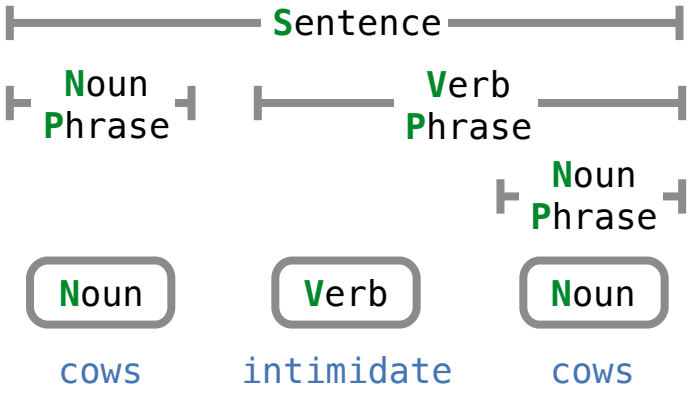
Representing Syntactic Structure



Representing Syntactic Structure

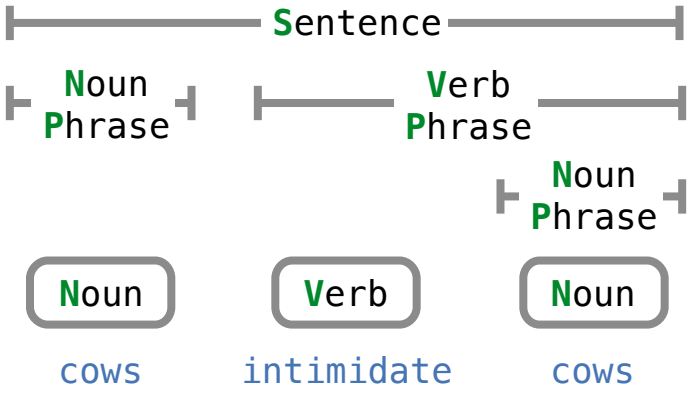


Representing Syntactic Structure



A **Tree** represents a phrase:

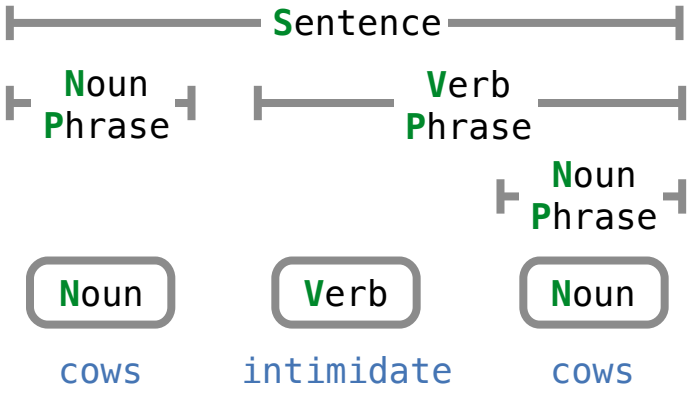
Representing Syntactic Structure



A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)

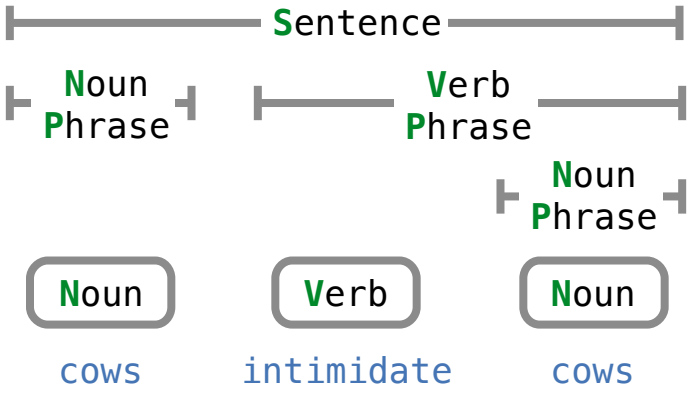
Representing Syntactic Structure



A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

Representing Syntactic Structure

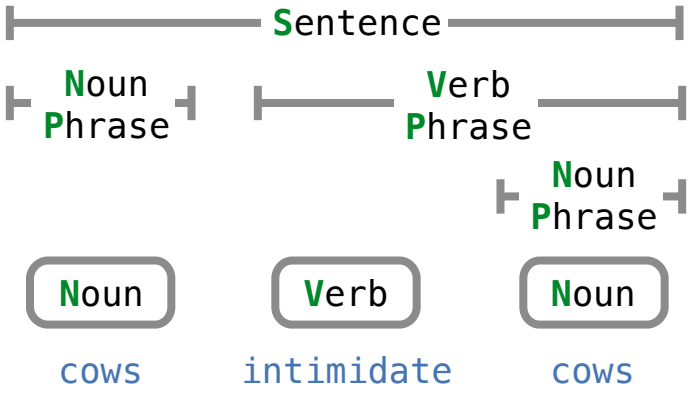


A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

Representing Syntactic Structure



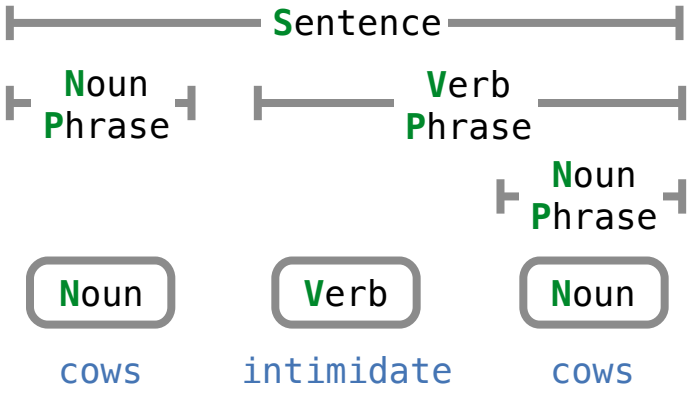
A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)

Representing Syntactic Structure



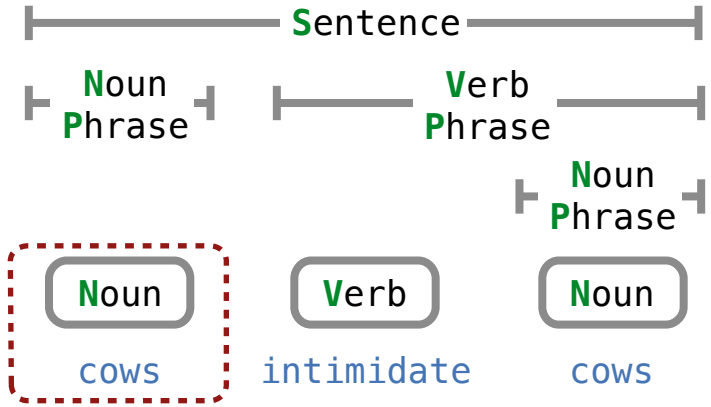
A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

Representing Syntactic Structure



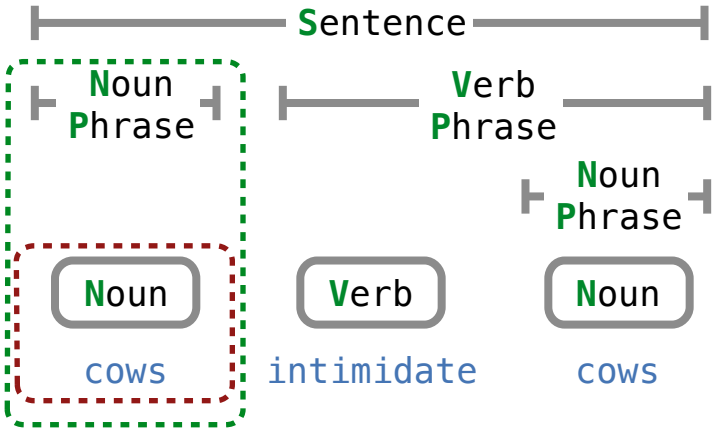
A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

Representing Syntactic Structure



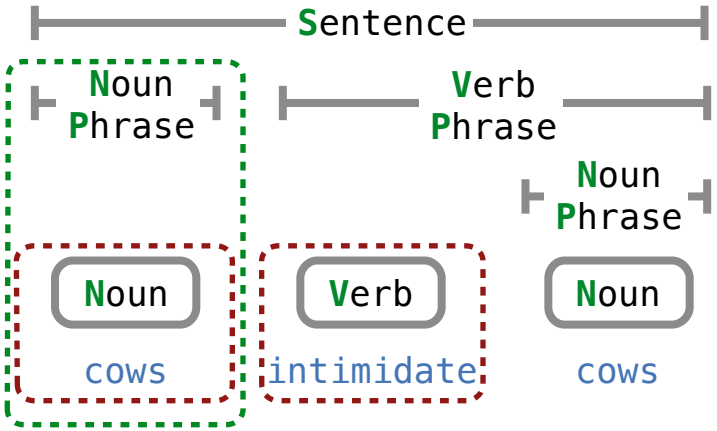
A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

Representing Syntactic Structure



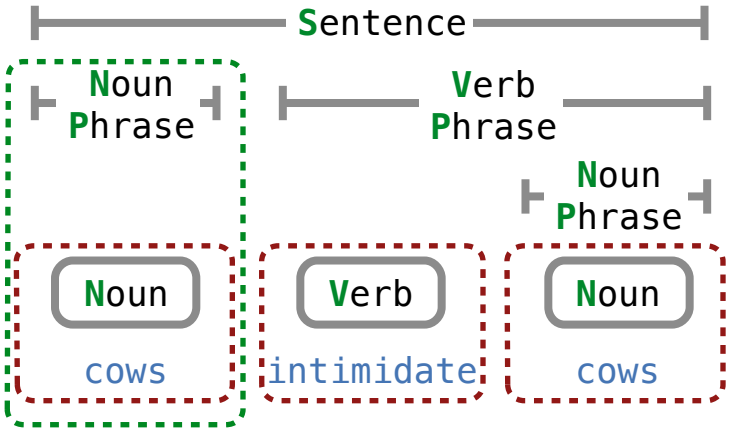
A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

Representing Syntactic Structure



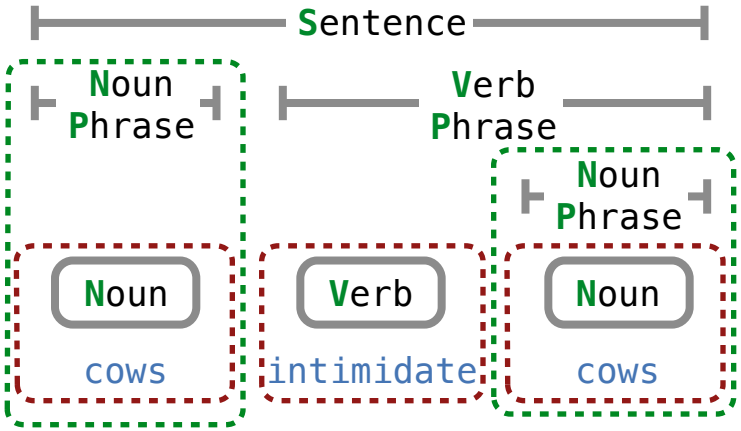
A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

Representing Syntactic Structure



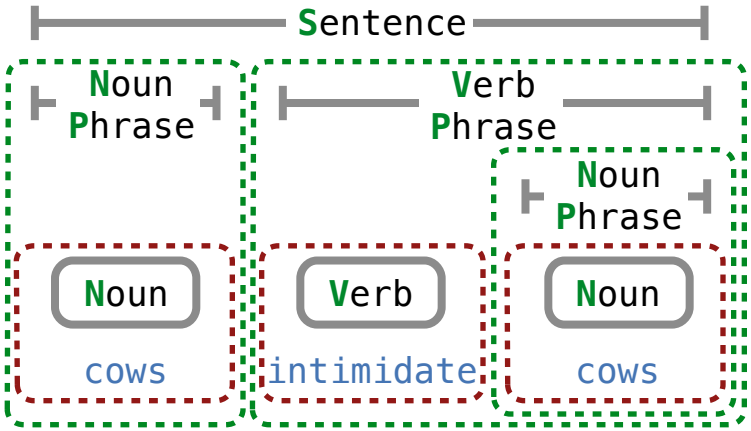
A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

Representing Syntactic Structure



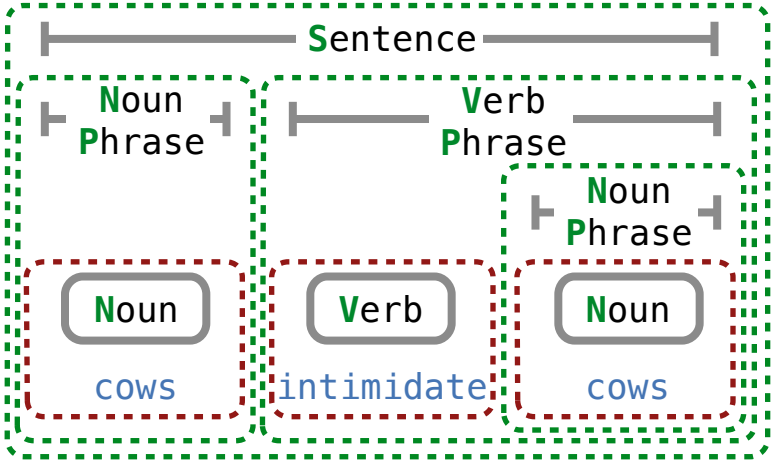
A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

Representing Syntactic Structure



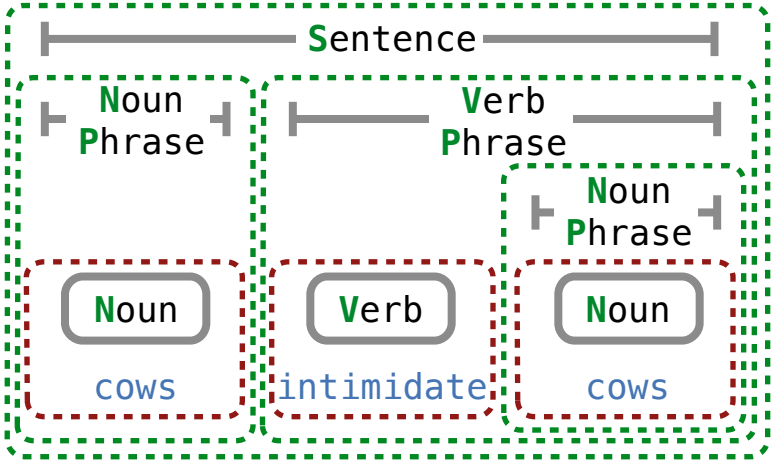
A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

Representing Syntactic Structure



A **Tree** represents a phrase:

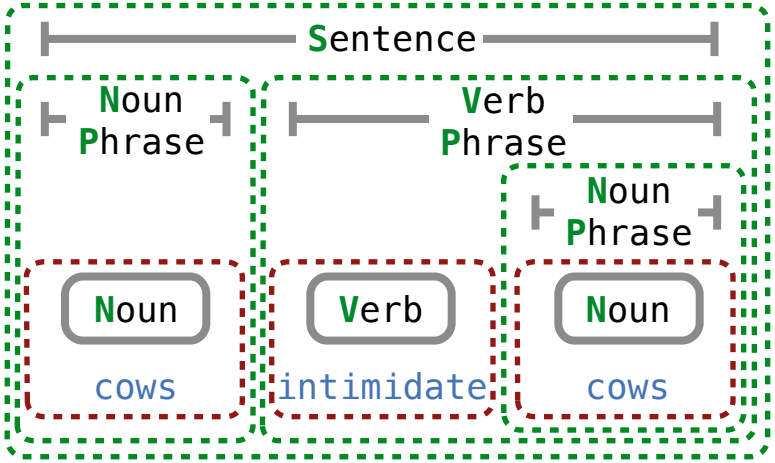
- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

```
cows = Leaf('N', 'cows')
```

Representing Syntactic Structure



A **Tree** represents a phrase:

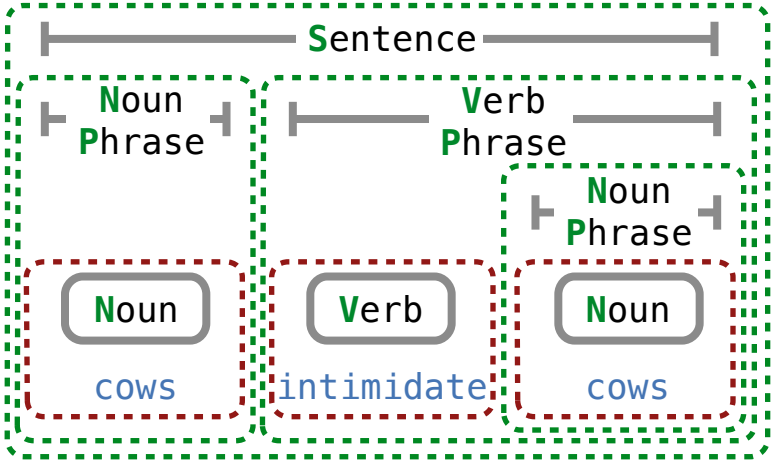
- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

```
cows = Leaf('N', 'cows')
intimidate = Leaf('V', 'intimidate')
```

Representing Syntactic Structure



A **Tree** represents a phrase:

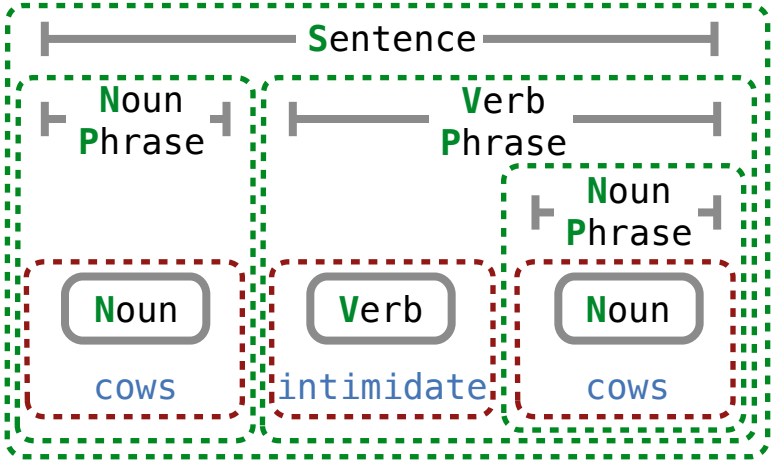
- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

```
cows = Leaf('N', 'cows')
intimidate = Leaf('V', 'intimidate')
S, NP, VP = 'S', 'NP', 'VP'
```

Representing Syntactic Structure



A **Tree** represents a phrase:

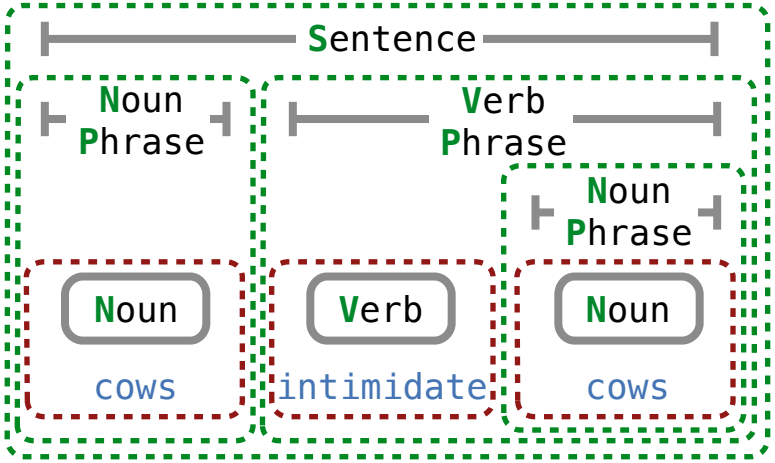
- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

```
cows = Leaf('N', 'cows')
intimidate = Leaf('V', 'intimidate')
S, NP, VP = 'S', 'NP', 'VP'
Tree(S, [Tree(NP, [cows]),
```

Representing Syntactic Structure



A **Tree** represents a phrase:

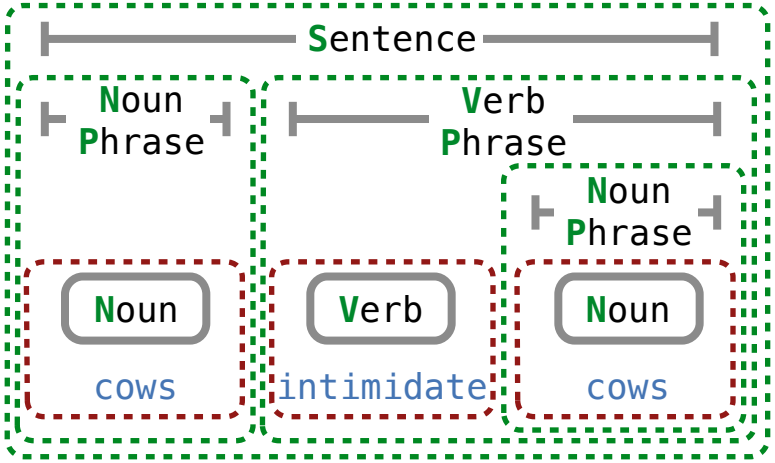
- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

```
cows = Leaf('N', 'cows')
intimidate = Leaf('V', 'intimidate')
S, NP, VP = 'S', 'NP', 'VP'
Tree(S, [Tree(NP, [cows]),
         Tree(VP, [intimidate,
```

Representing Syntactic Structure



A **Tree** represents a phrase:

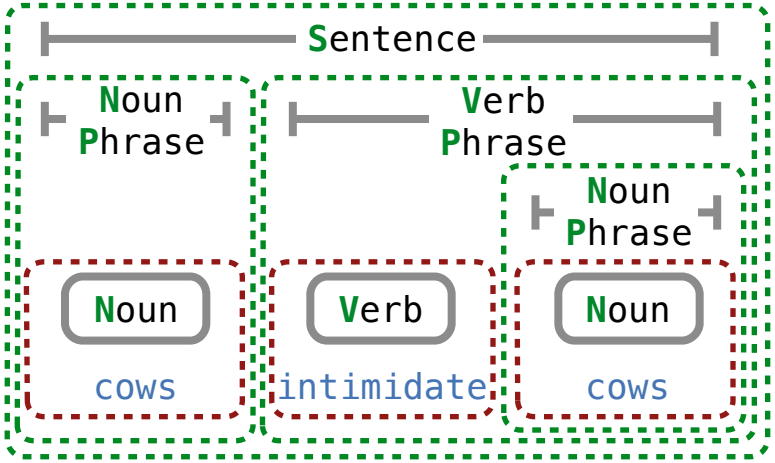
- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

```
cows = Leaf('N', 'cows')
intimidate = Leaf('V', 'intimidate')
S, NP, VP = 'S', 'NP', 'VP'
Tree(S, [Tree(NP, [cows]),
         Tree(VP, [intimidate,
                   Tree(NP, [cows])])])
```

Representing Syntactic Structure



A **Tree** represents a phrase:

- **tag** -- What kind of phrase (e.g., **S**, **NP**, **VP**)
- **branches** -- Sequence of **Tree** or **Leaf** components

A **Leaf** represents a single word:

- **tag** -- What kind of word (e.g., **N**, **V**)
- **word** -- The word

```
cows = Leaf('N', 'cows')
intimidate = Leaf('V', 'intimidate')
S, NP, VP = 'S', 'NP', 'VP'
Tree(S, [Tree(NP, [cows]),
         Tree(VP, [intimidate,
                   Tree(NP, [cows])])])
```

(Demo)

Grammars

Context-Free Grammar Rules

Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

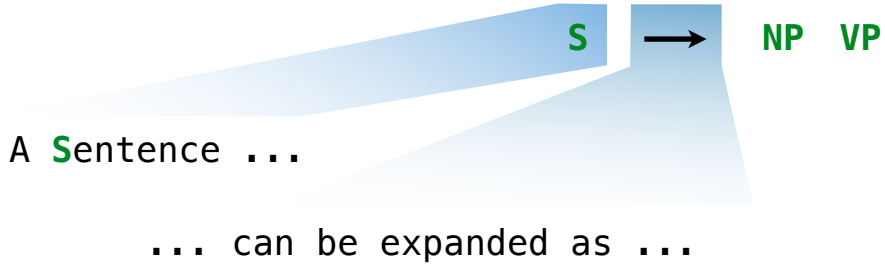
$$\mathbf{S} \longrightarrow \mathbf{NP}\ \mathbf{VP}$$

Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

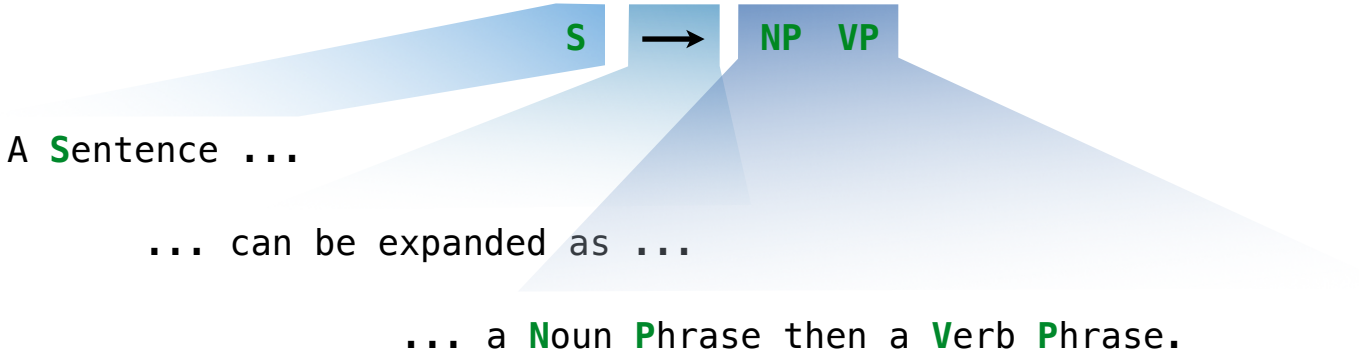
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words



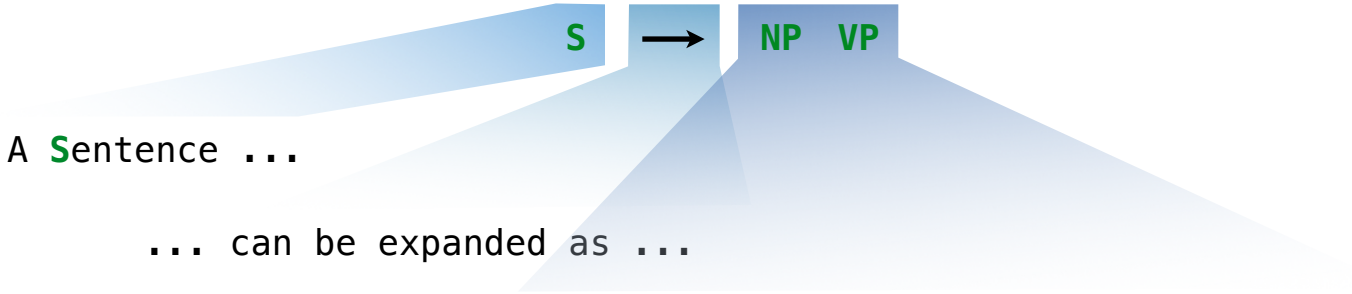
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words



Context-Free Grammar Rules

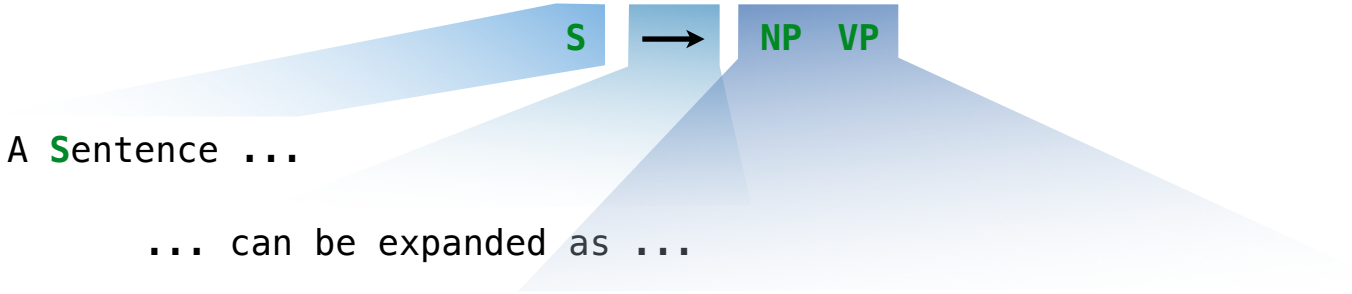
A grammar rule describes how a tag can be expanded as a sequence of tags or words



Grammar

Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

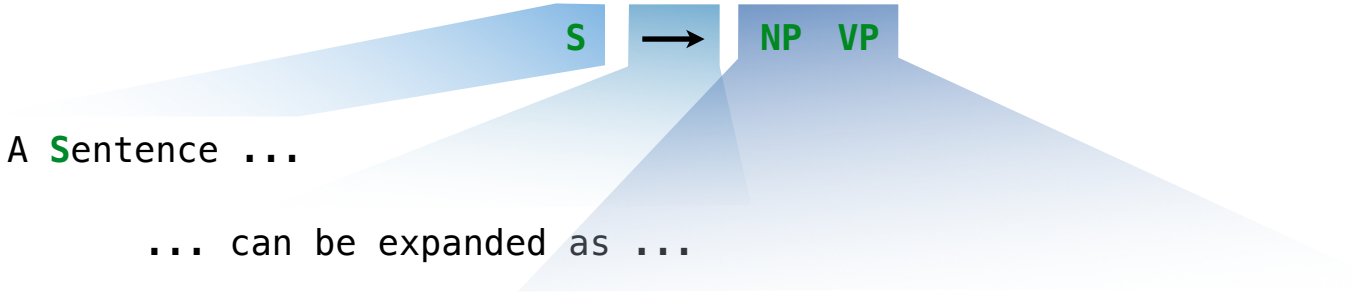


Grammar

S ⟶ NP VP

Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

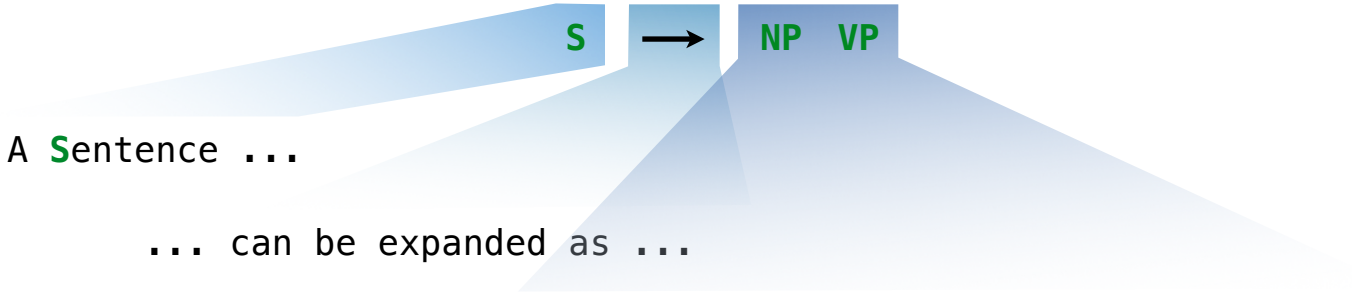


Grammar

S → NP VP

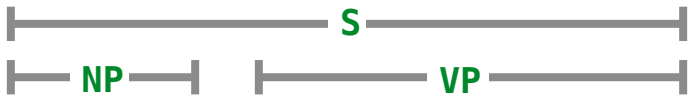
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words



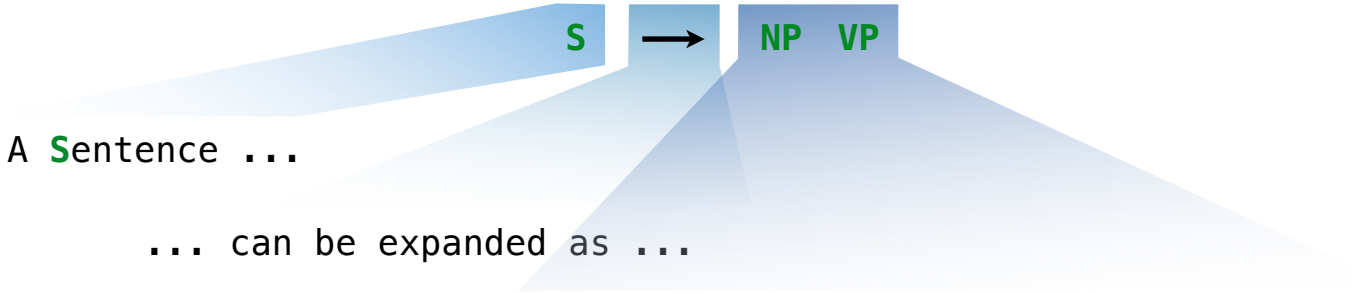
Grammar

S → NP VP



Context-Free Grammar Rules

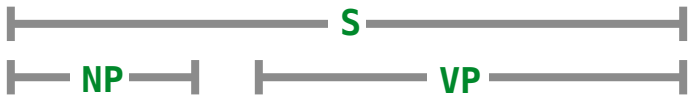
A grammar rule describes how a tag can be expanded as a sequence of tags or words



Grammar

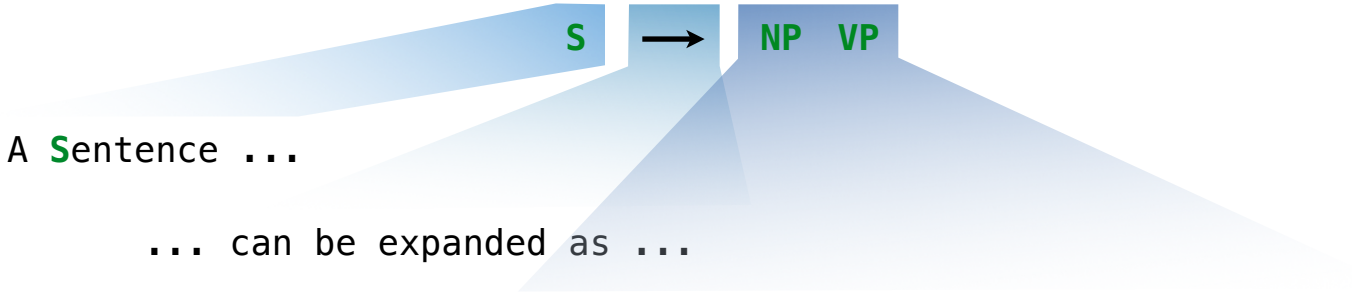
S → NP VP

NP → N



Context-Free Grammar Rules

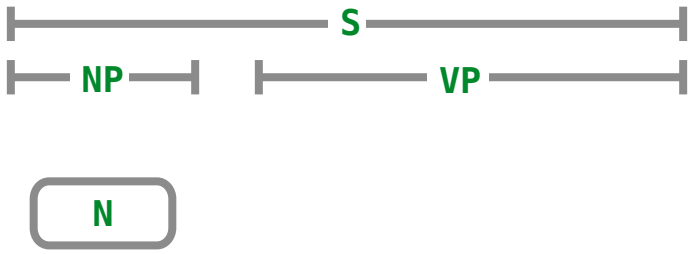
A grammar rule describes how a tag can be expanded as a sequence of tags or words



Grammar

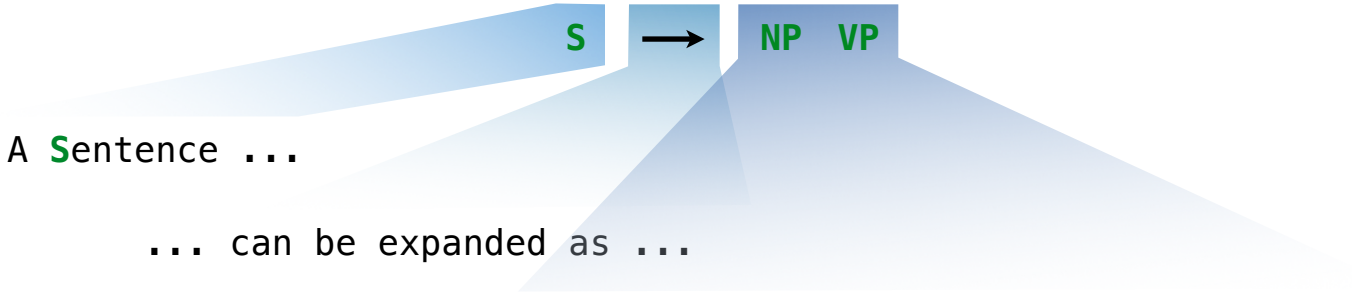
S → NP VP

NP → N



Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

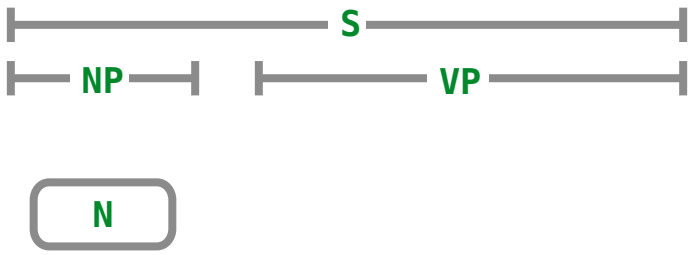


Grammar

S → NP VP

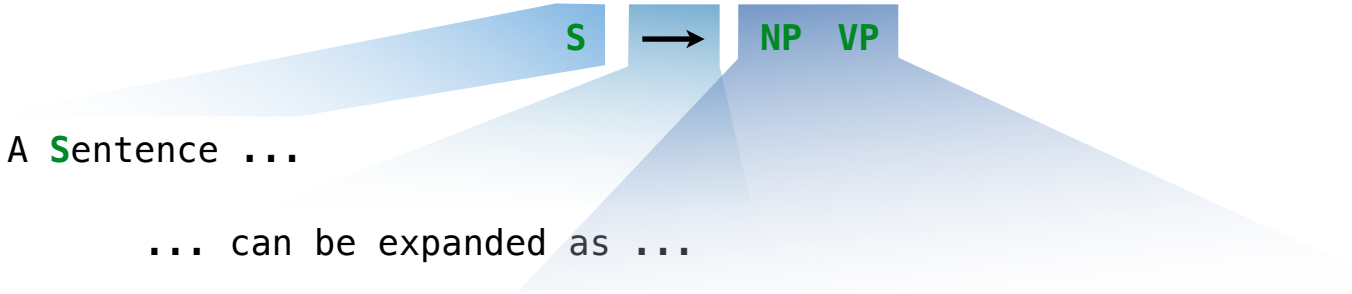
NP → N

N → buffalo



Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

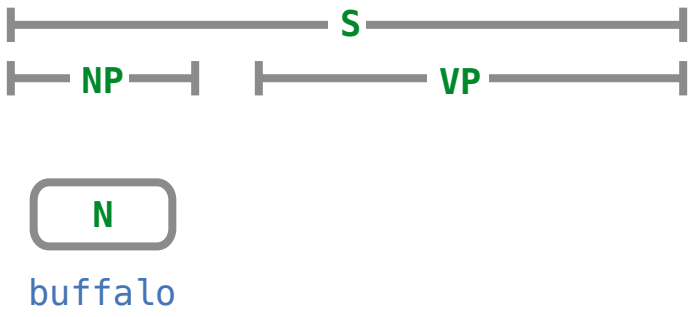


Grammar

S → NP VP

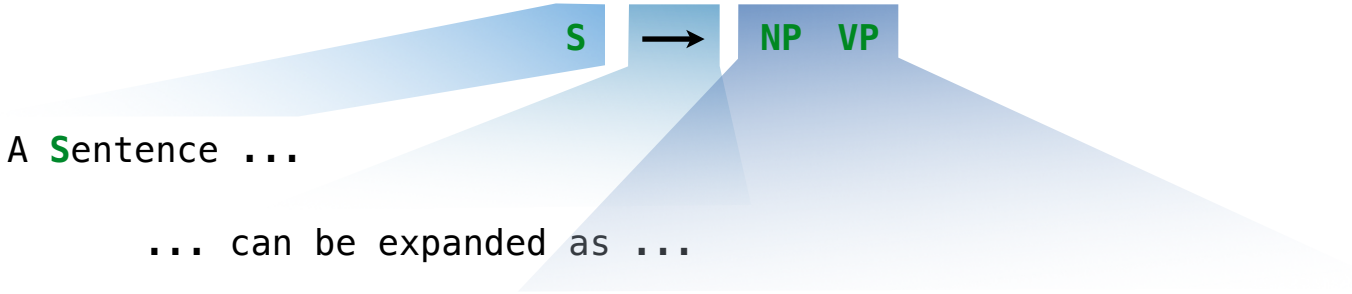
NP → N

N → buffalo



Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

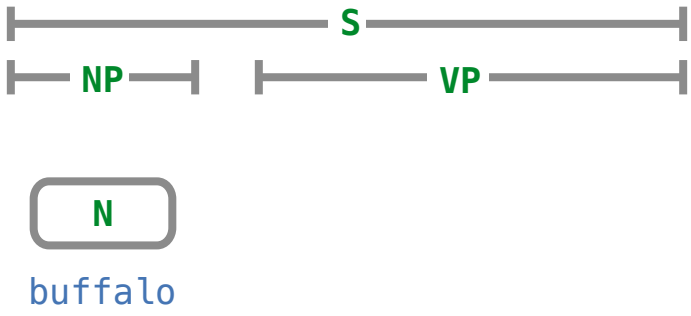


Grammar

S ⟶ NP VP

NP ⟶ N

N ⟶ buffalo



Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

S → NP VP

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

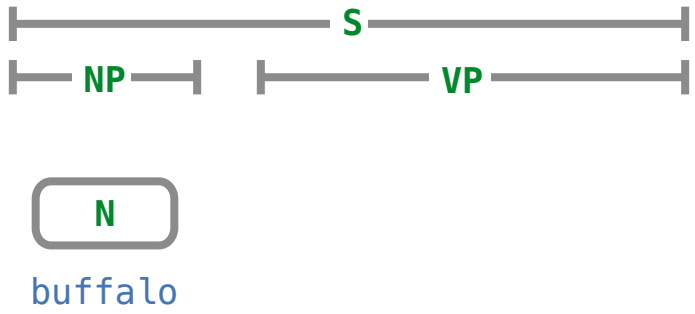
Grammar

S → NP VP

NP → N

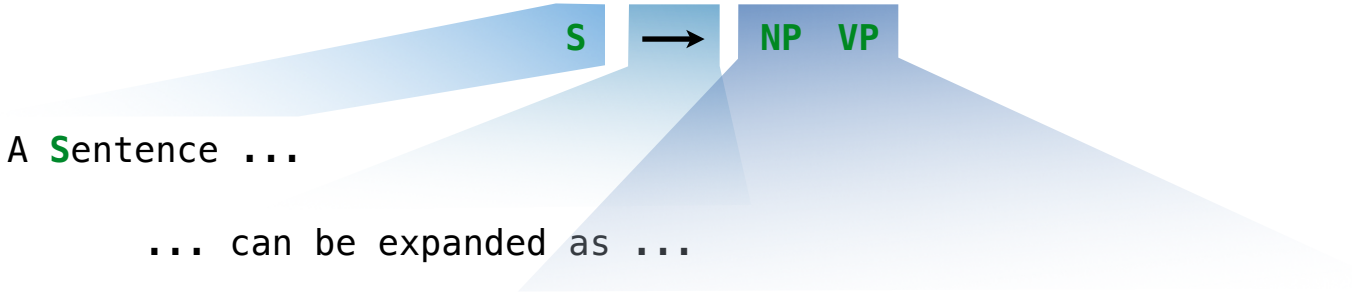
N → buffalo

VP → V NP



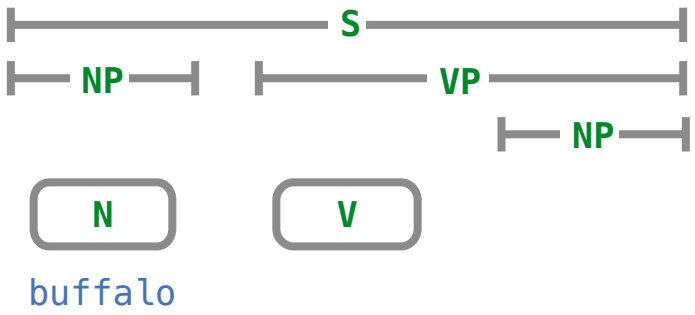
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words



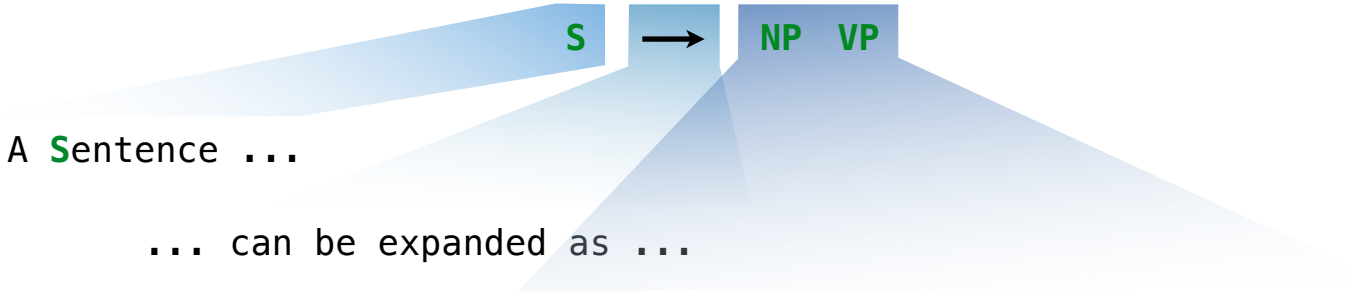
Grammar

S → NP VP
NP → N
N → buffalo
VP → V NP



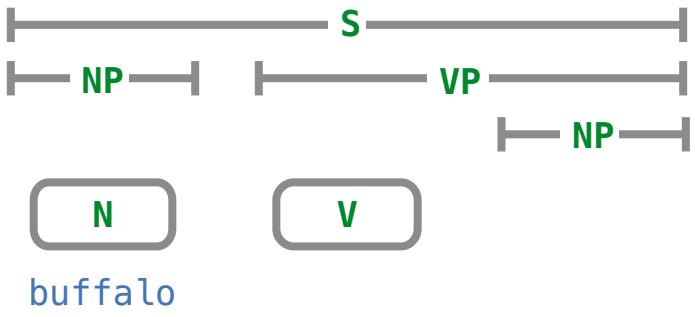
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words



Grammar

S → NP VP
NP → N
N → buffalo
VP → V NP
V → buffalo



Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

S → NP VP

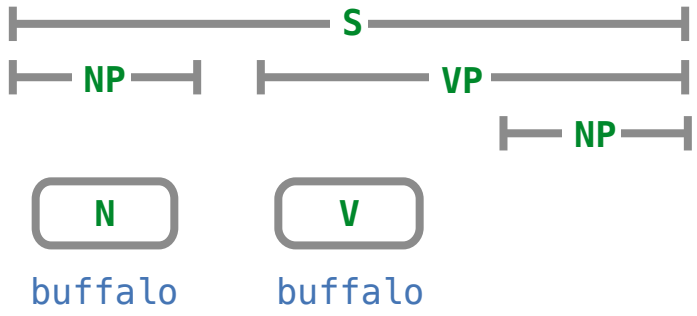
A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

Grammar

S → NP VP
NP → N
N → buffalo
VP → V NP
V → buffalo



Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

S → NP VP

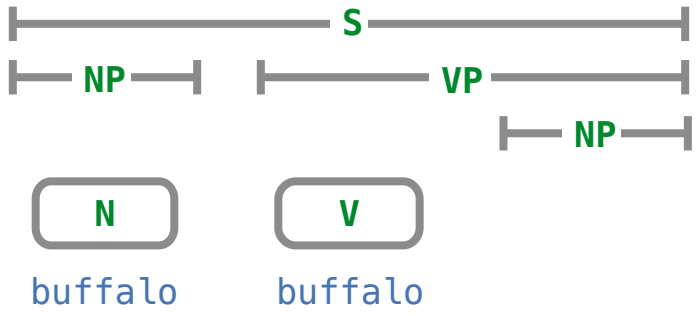
A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

Grammar

S → NP VP
NP → N
N → buffalo
VP → V NP
V → buffalo



Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words

S → NP VP

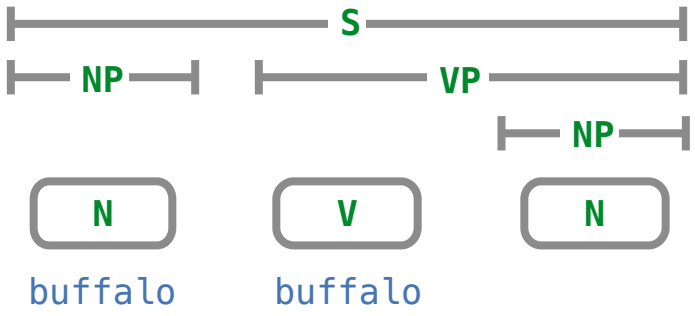
A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

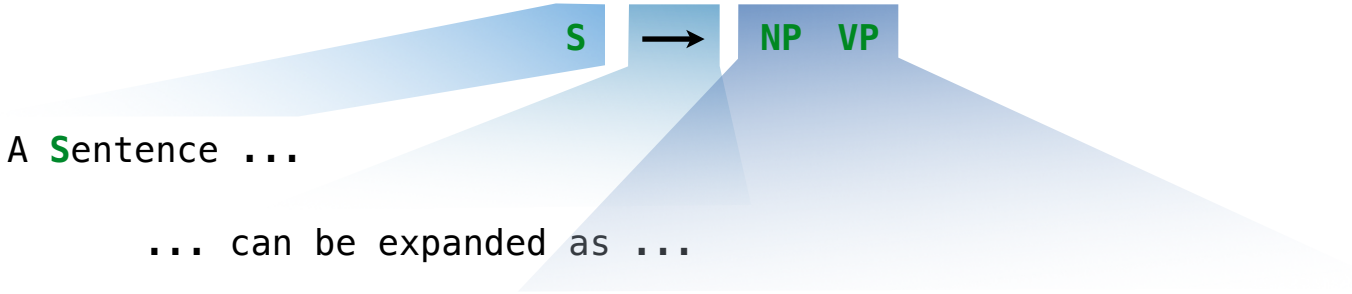
Grammar

S → NP VP
NP → N
N → buffalo
VP → V NP
V → buffalo



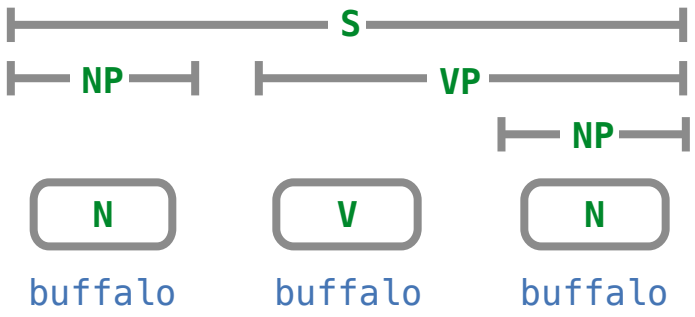
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words



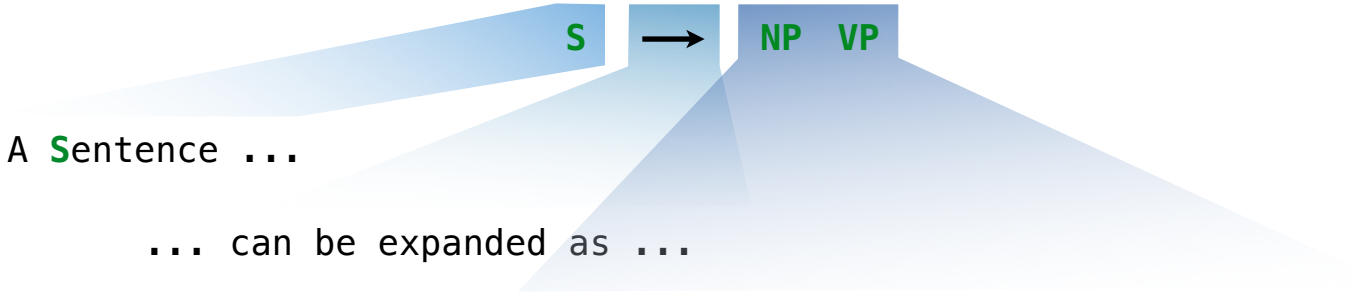
Grammar

S → NP VP
NP → N
N → buffalo
VP → V NP
V → buffalo



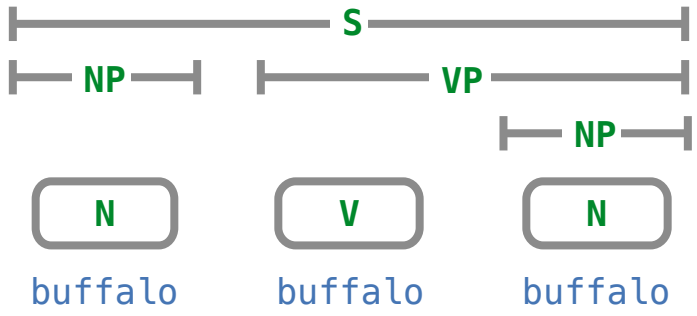
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words



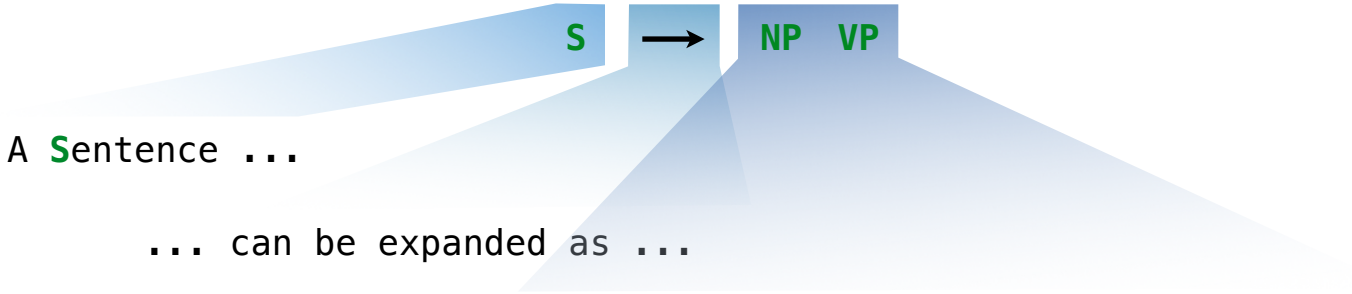
Grammar

S → NP VP
NP → N
N → buffalo
VP → V NP
V → buffalo



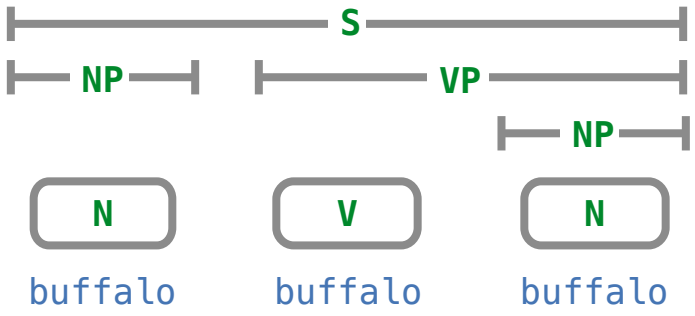
Context-Free Grammar Rules

A grammar rule describes how a tag can be expanded as a sequence of tags or words



Grammar

S → NP VP
NP → N
N → buffalo
VP → V NP
V → buffalo



(Demo)

Parsing

Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

buffalo buffalo buffalo buffalo

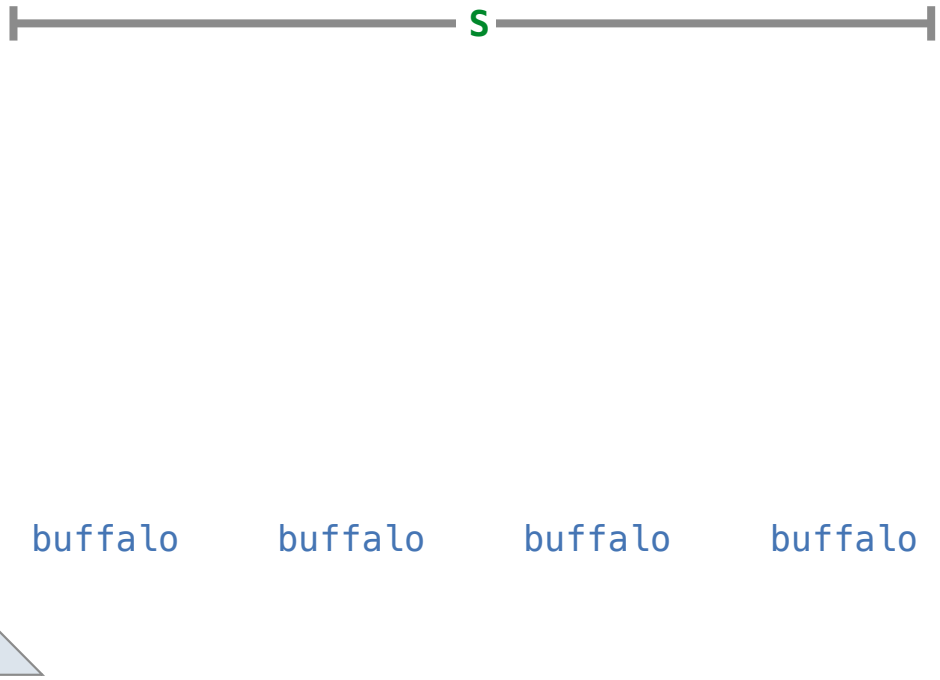
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

buffalo buffalo buffalo buffalo

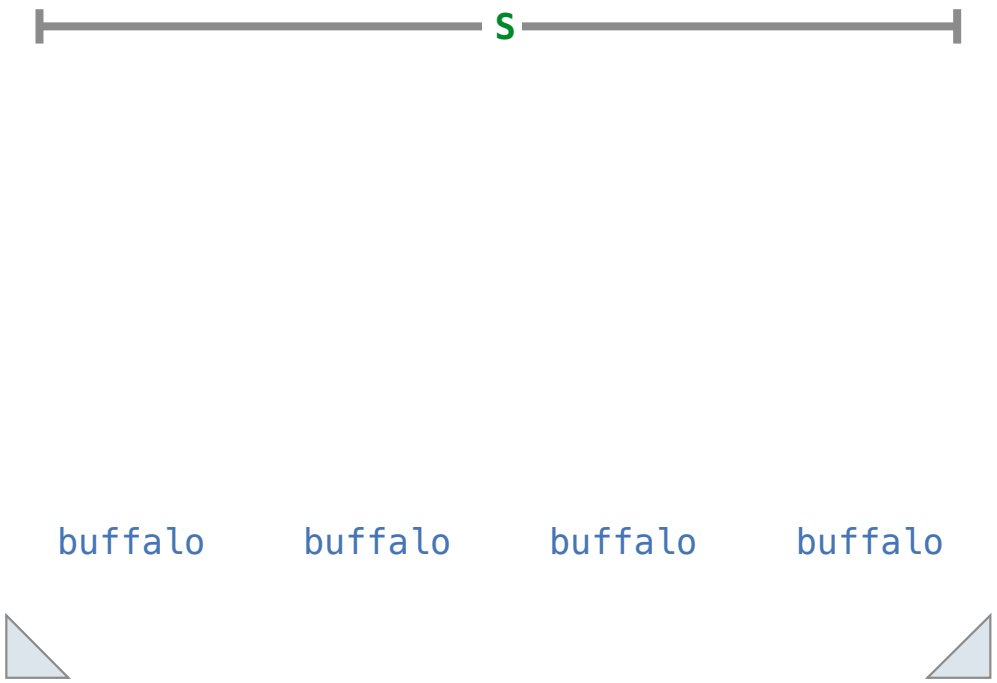
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



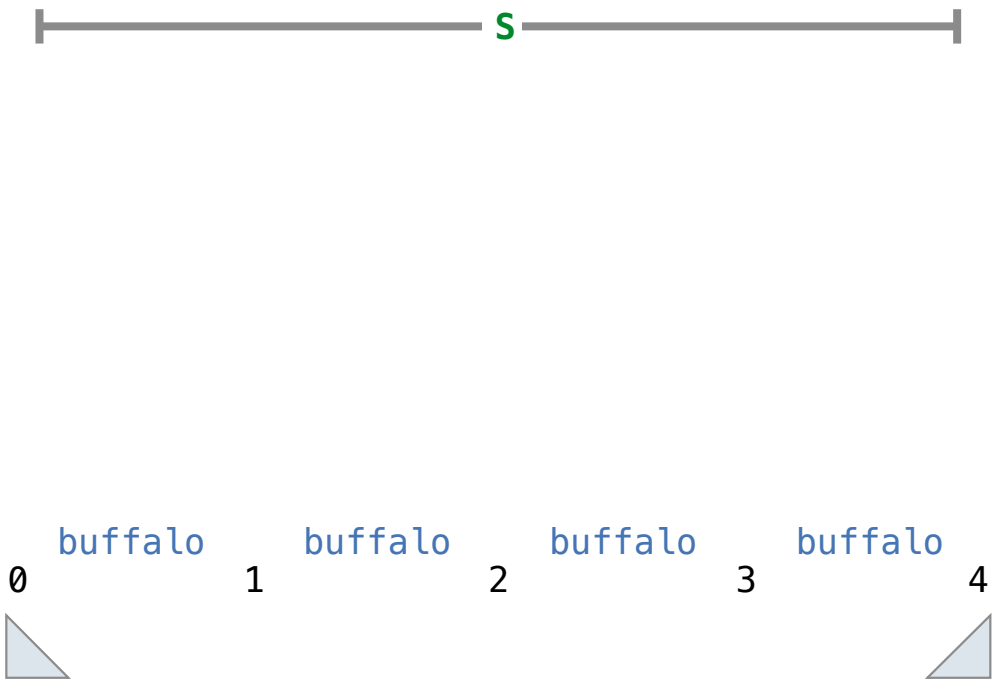
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



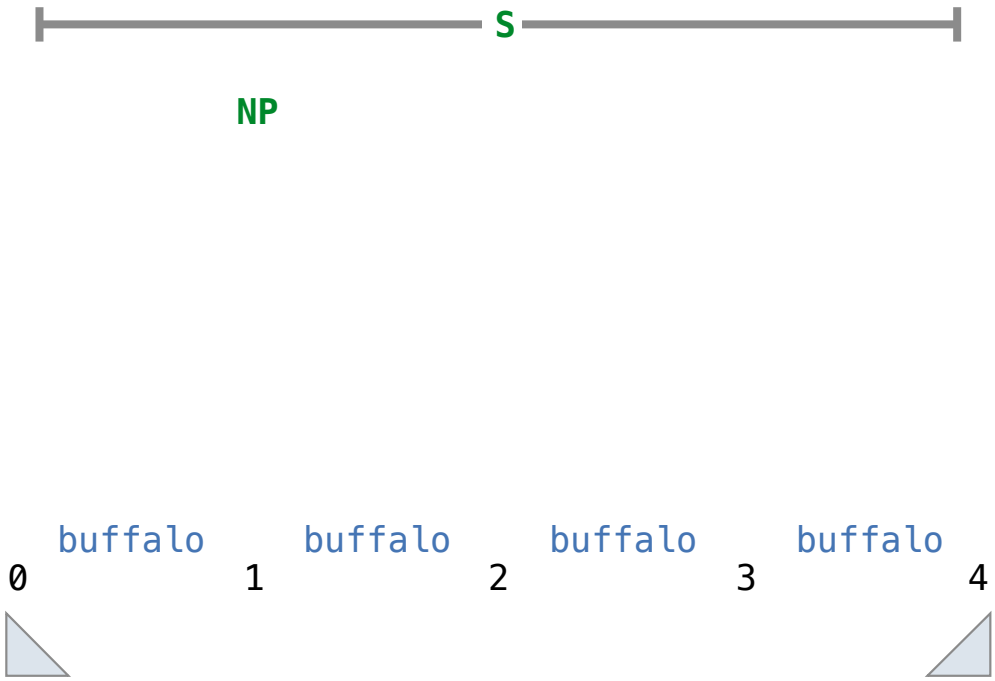
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



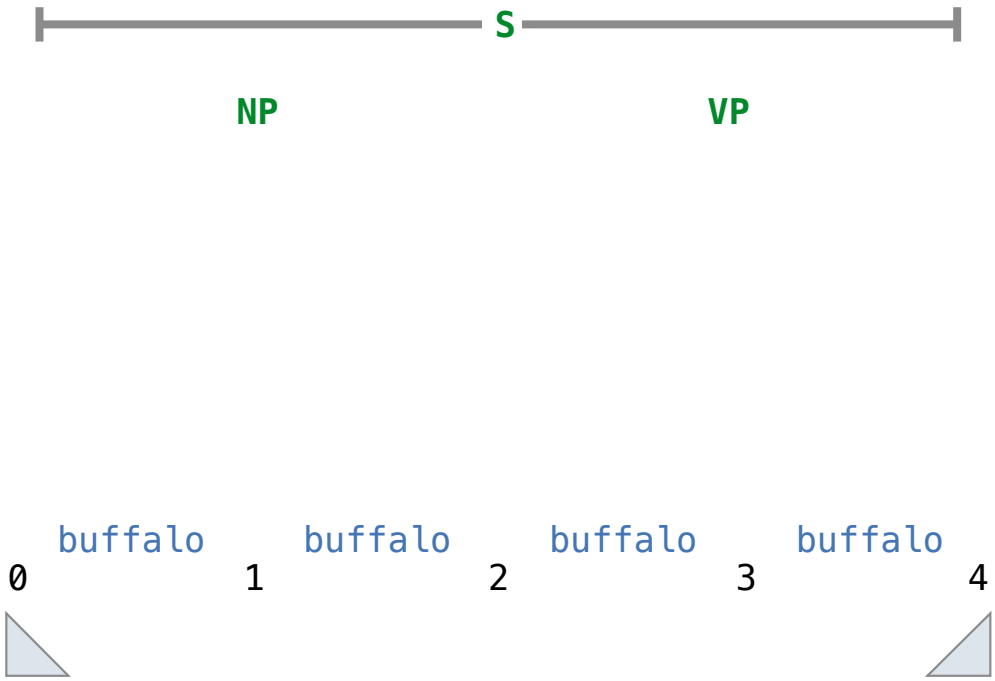
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



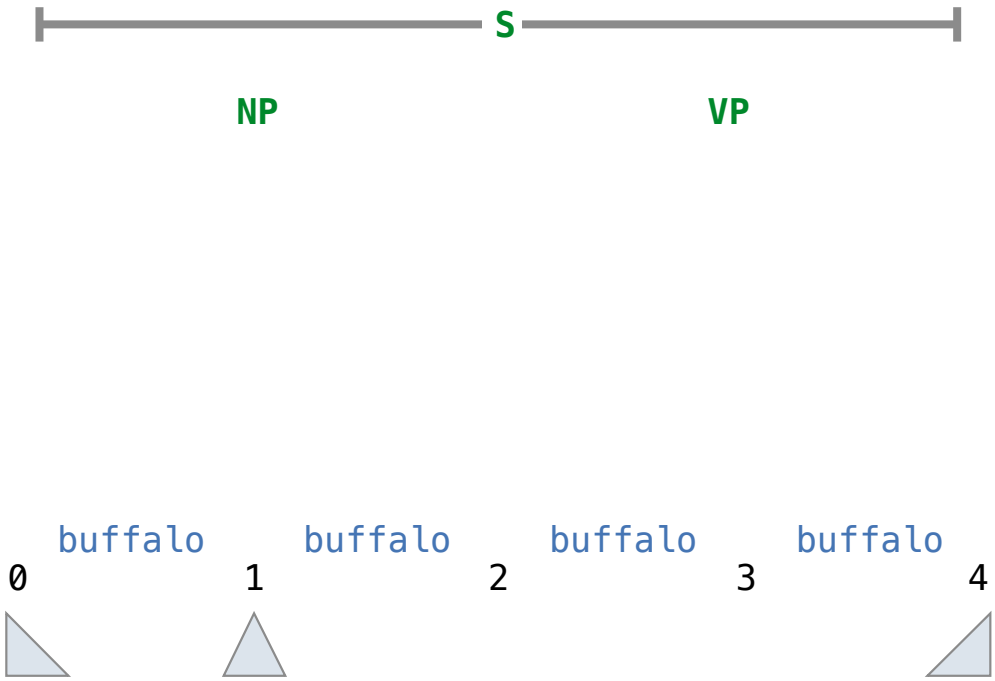
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



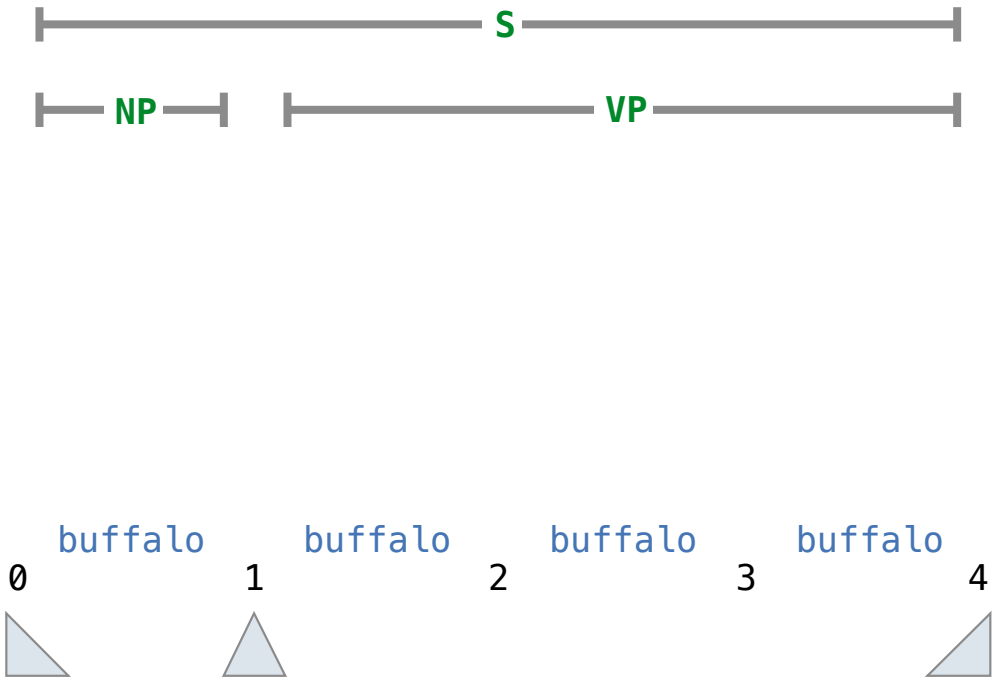
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



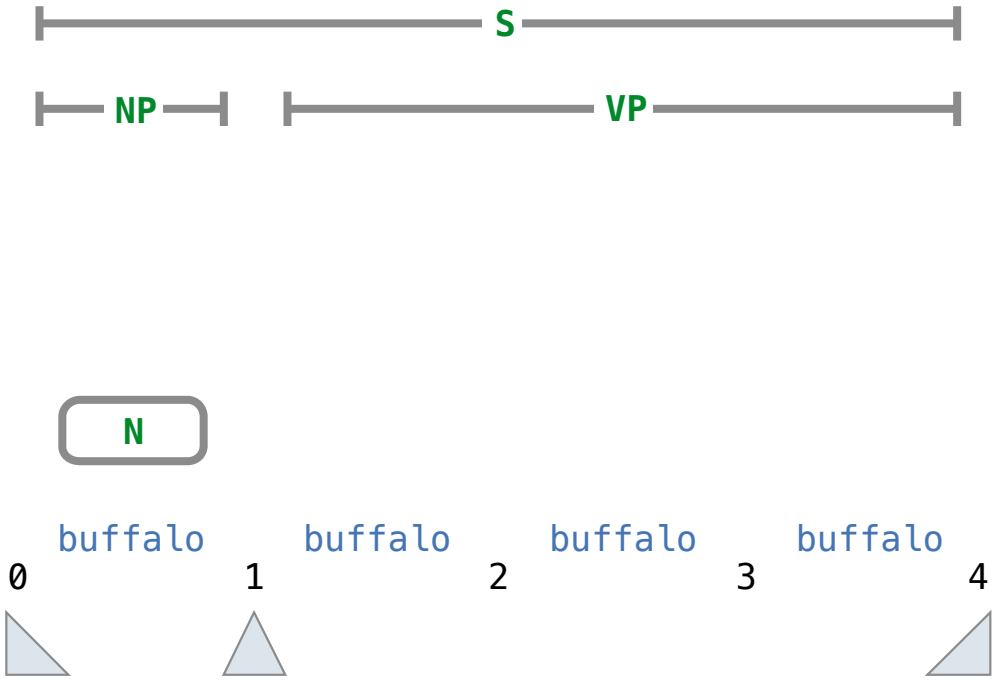
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



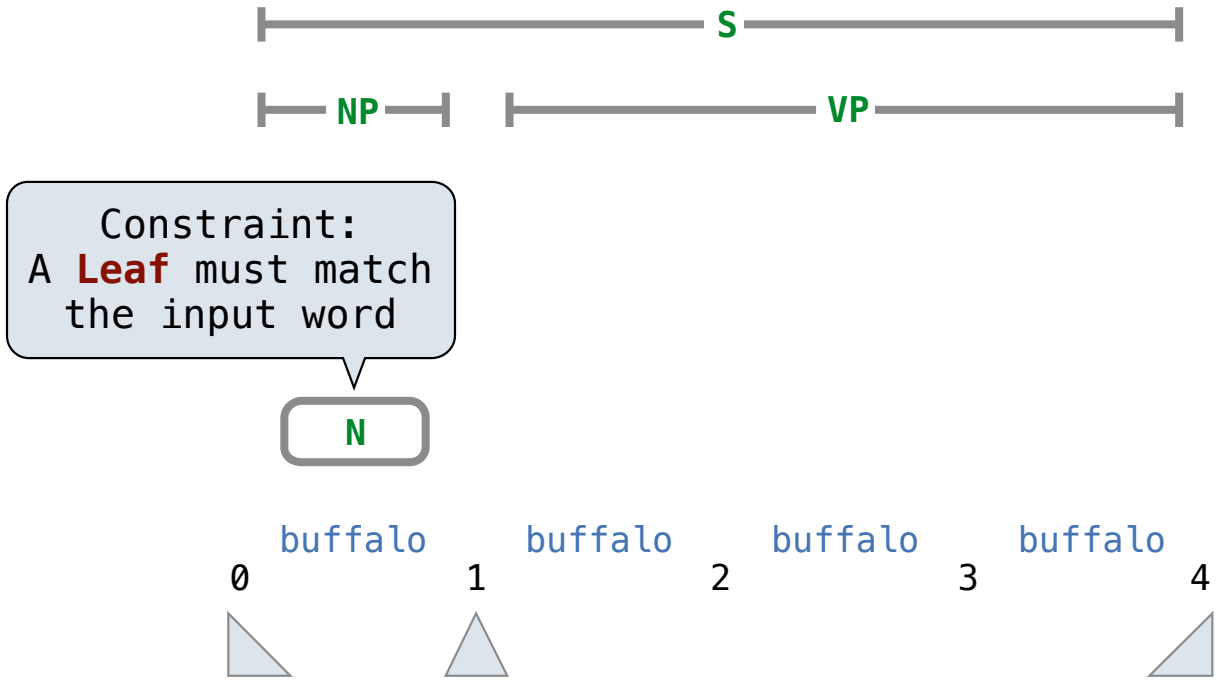
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



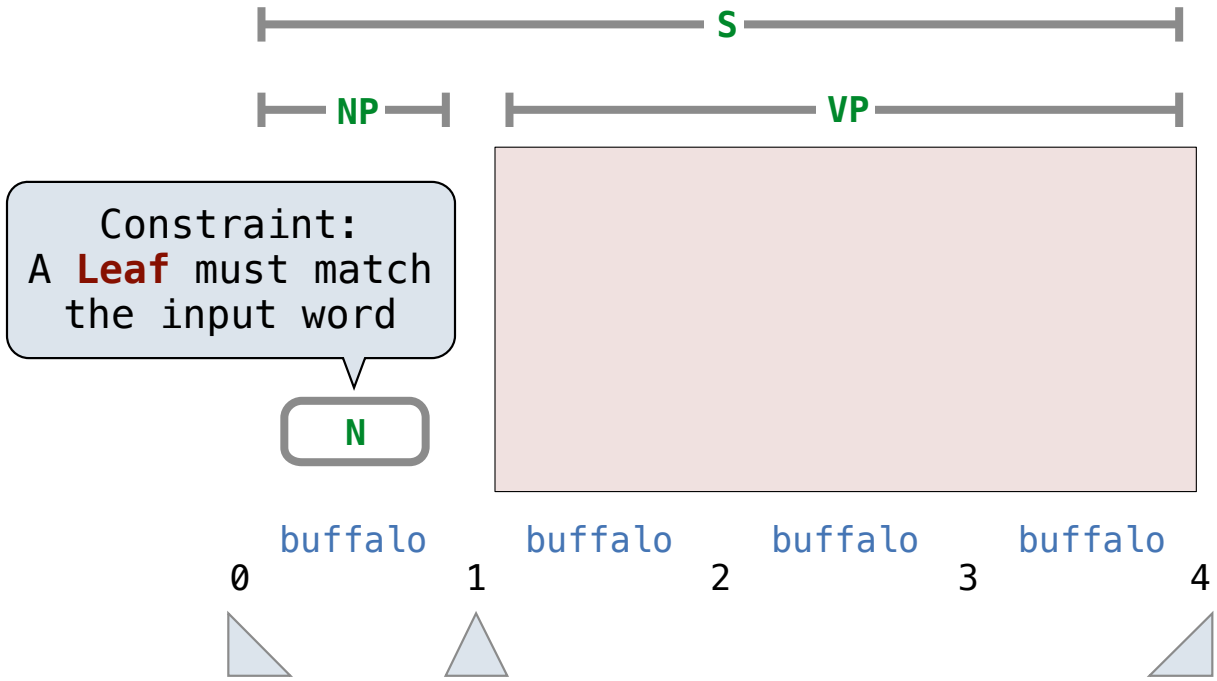
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



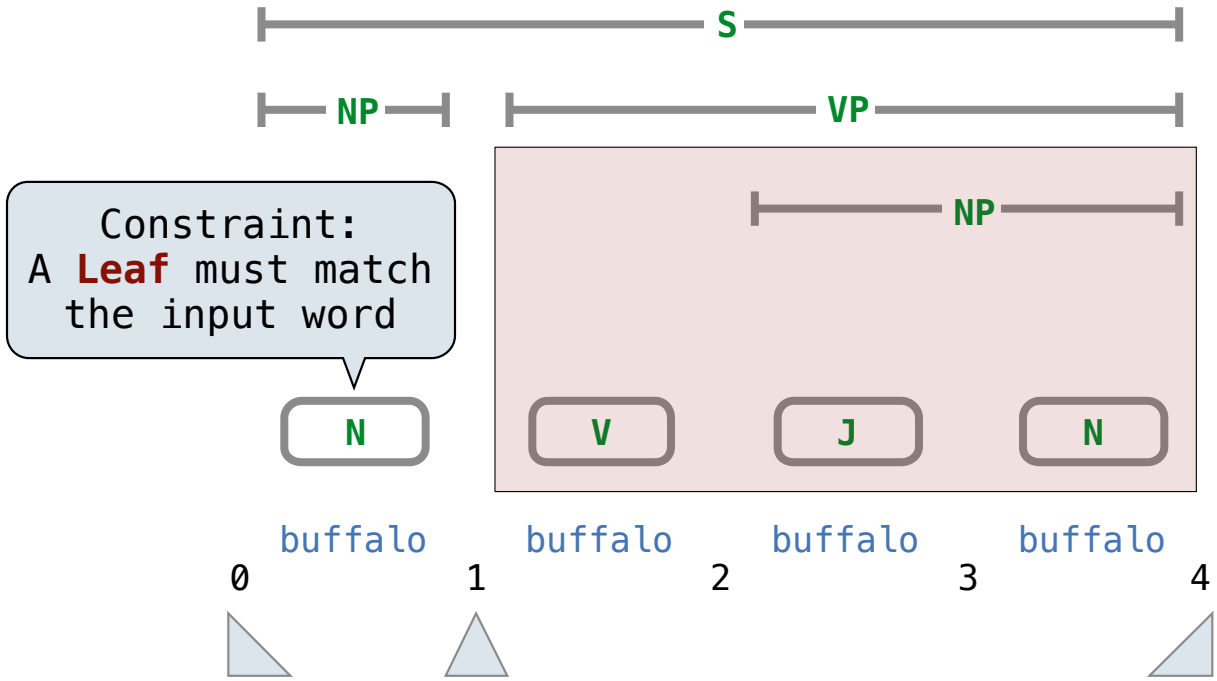
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



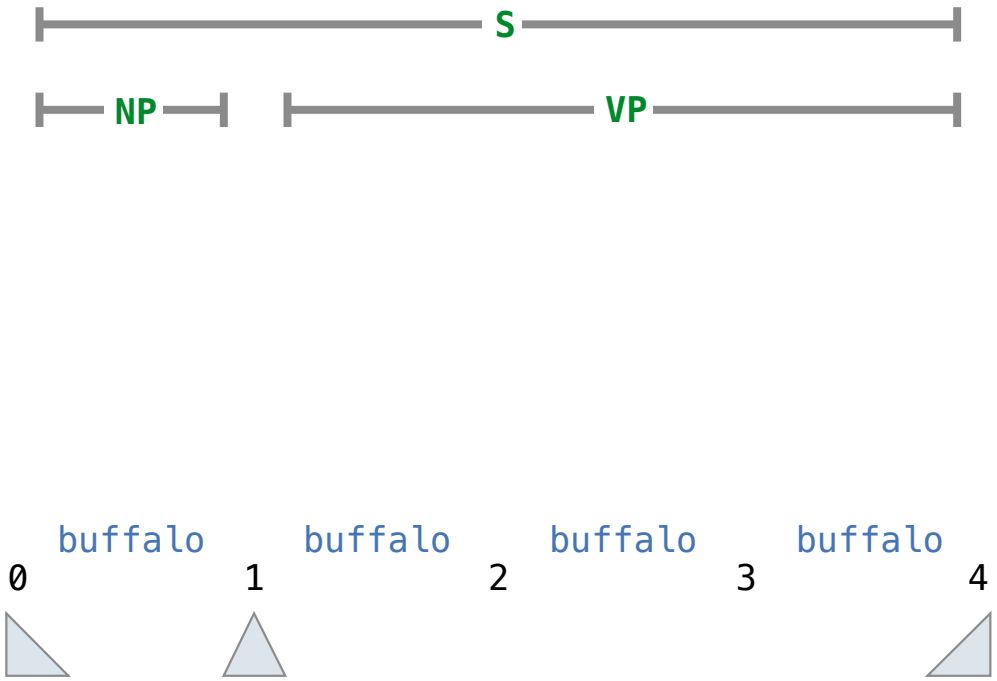
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



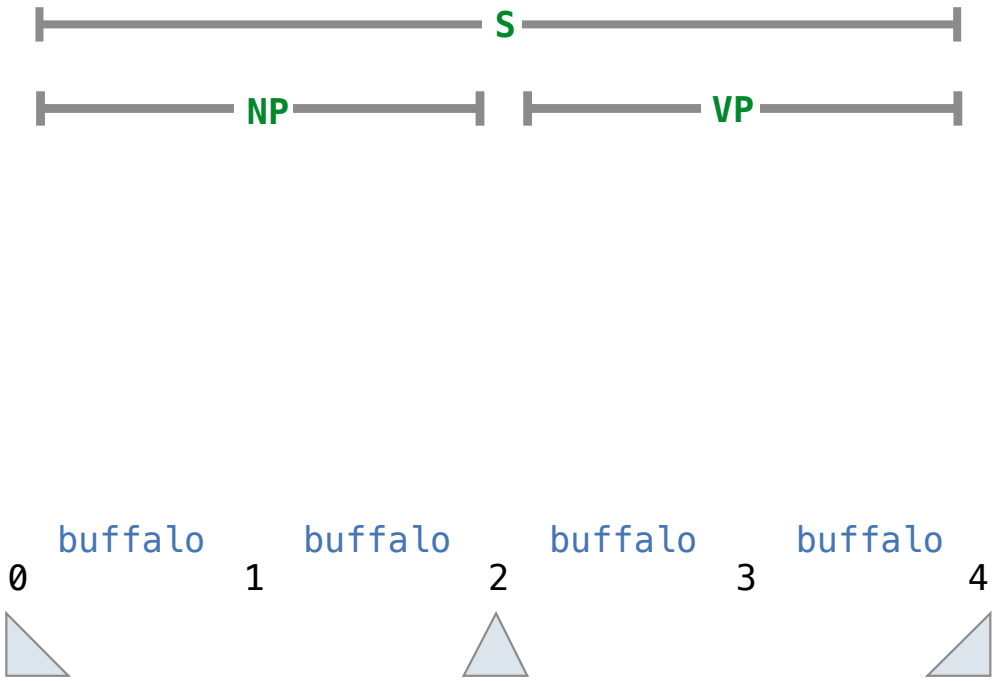
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



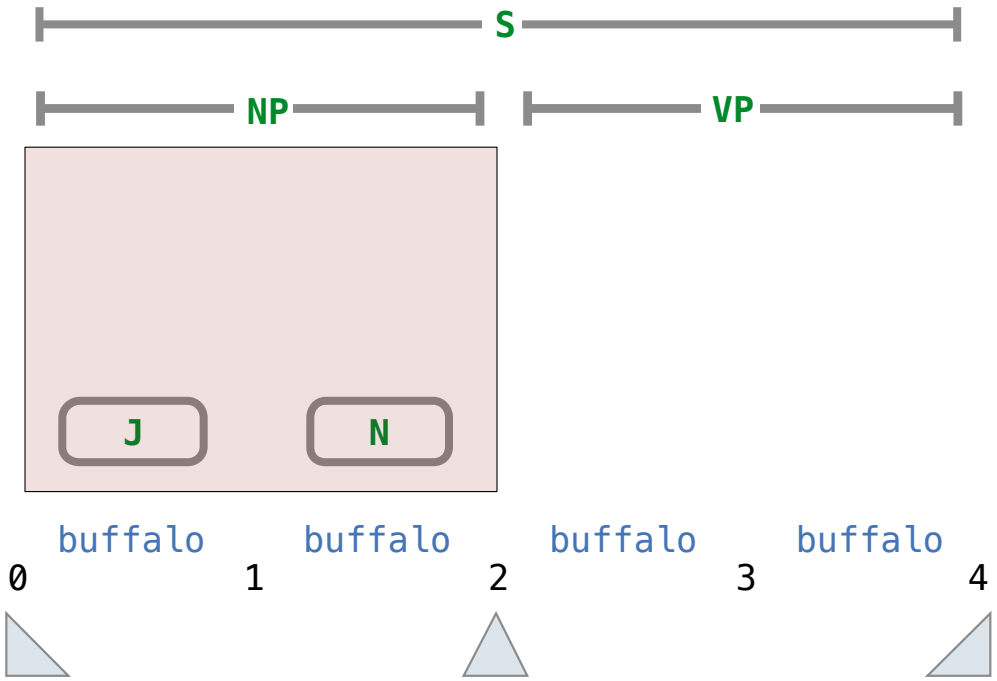
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



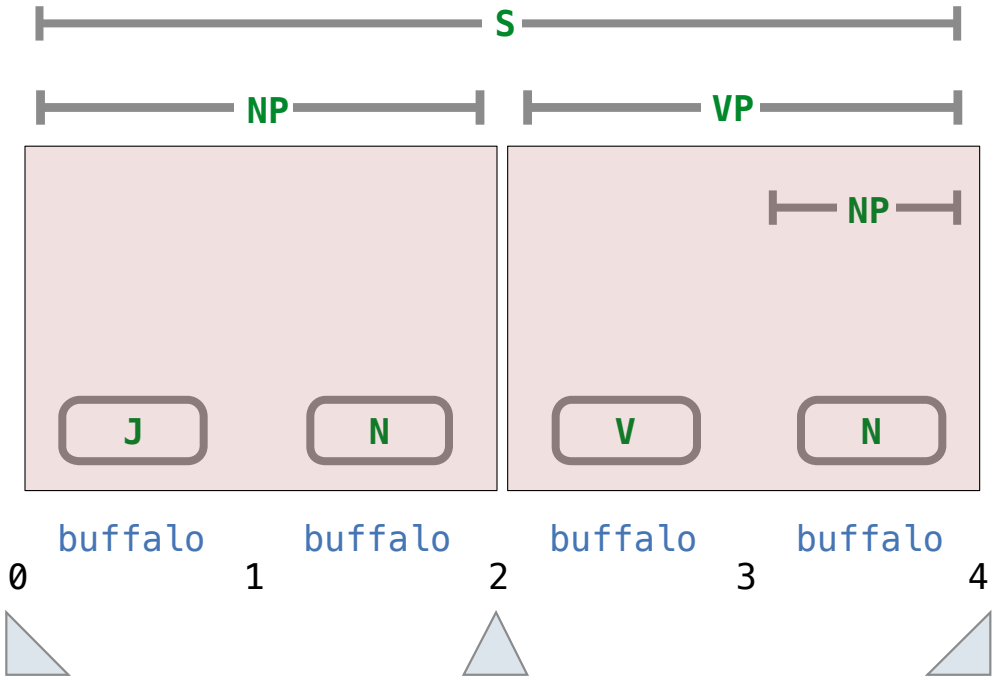
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



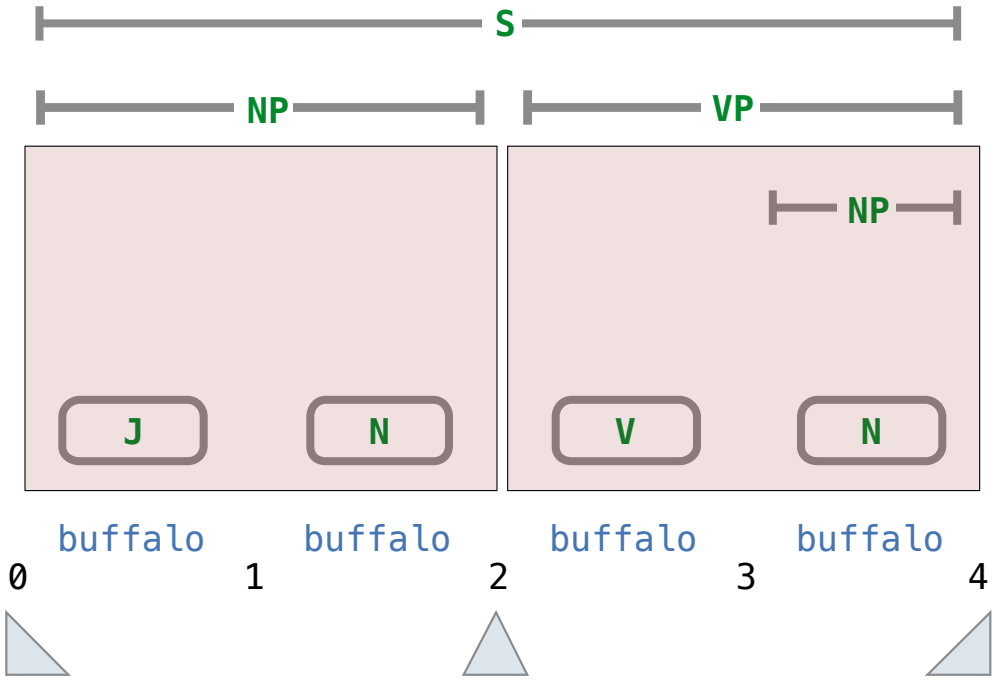
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



Exhaustive Parsing

Expand all tags recursively, but constrain words to match input



(Demo)

Learning

(Demo)

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

teacher strikes idle kids

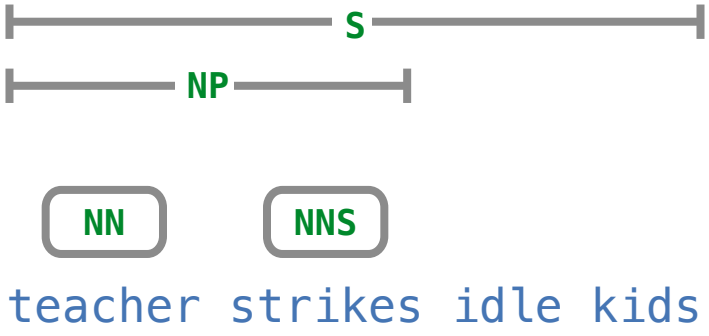
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

teacher strikes idle kids

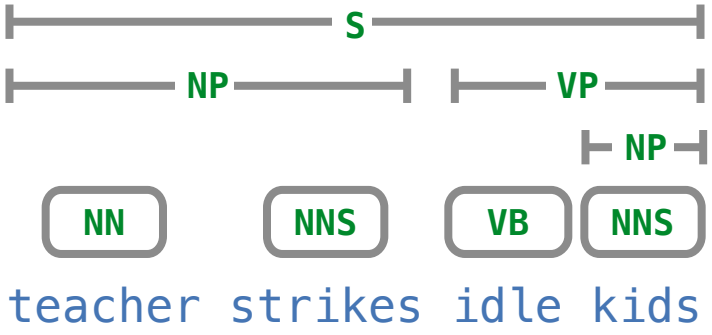
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common



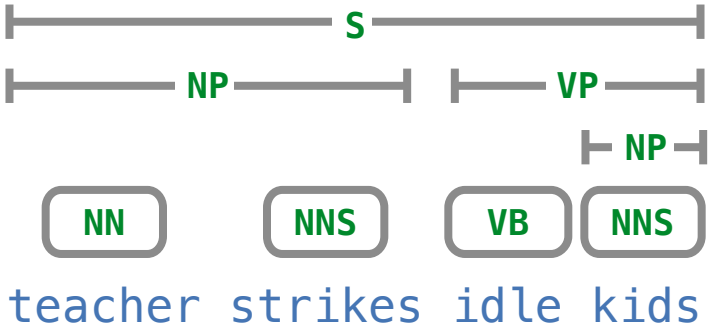
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common



Scoring a Tree Using Relative Frequencies

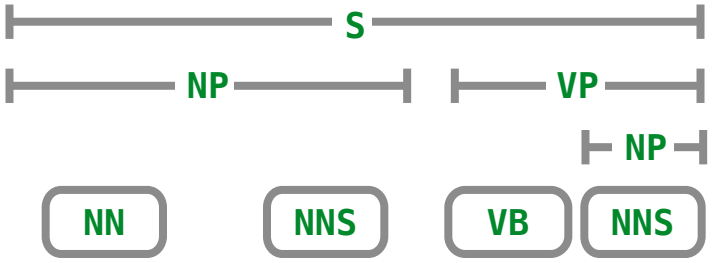
Not all syntactic structures are equally common



S	⟶	NP VP	NN	⟶	teacher
NP	⟶	NN NNS	NNS	⟶	strikes
VP	⟶	VB NP	VB	⟶	idle
NP	⟶	NNS	NNS	⟶	kids

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

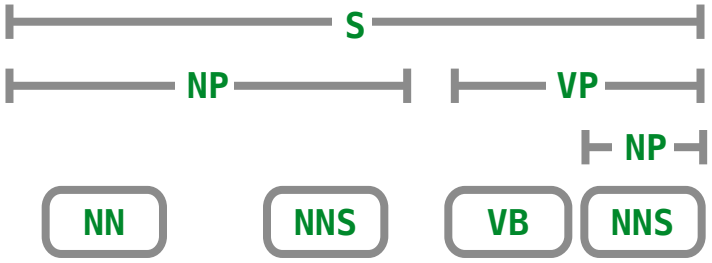


Rule frequency per 100,000 tags

S → NP VP		NN →	teacher
NP → NN NNS		NNS →	strikes
VP → VB NP		VB →	idle
NP → NNS		NNS →	kids

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

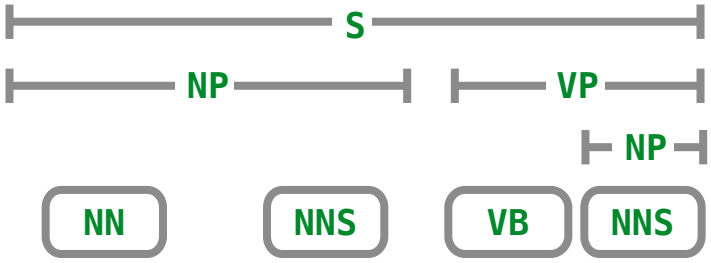


Rule frequency per 100,000 tags

S → NP VP	25372	NN →	teacher
NP → NN NNS		NNS →	strikes
VP → VB NP		VB →	idle
NP → NNS		NNS →	kids

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

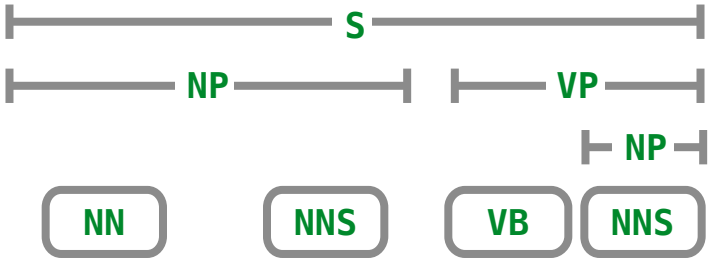


Rule frequency per 100,000 tags

S ⟶ NP VP	25372	NN ⟶	teacher
NP ⟶ NN NNS	1335	NNS ⟶	strikes
VP ⟶ VB NP		VB ⟶	idle
NP ⟶ NNS		NNS ⟶	kids

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

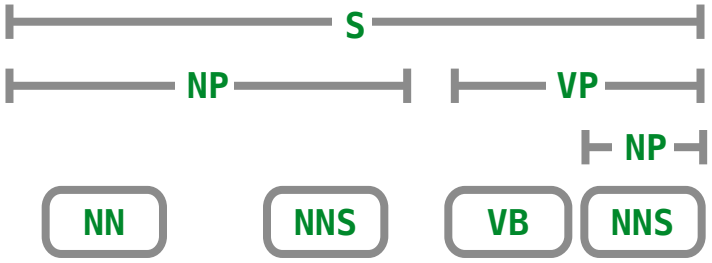


Rule frequency per 100,000 tags

S → NP VP	25372	NN →	teacher
NP → NN NNS	1335	NNS →	strikes
VP → VB NP	6679	VB →	idle
NP → NNS		NNS →	kids

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

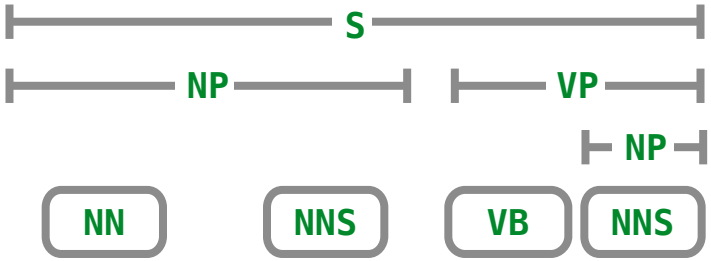


Rule frequency per 100,000 tags

S → NP VP	25372	NN →	teacher
NP → NN NNS	1335	NNS →	strikes
VP → VB NP	6679	VB →	idle
NP → NNS	4282	NNS →	kids

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

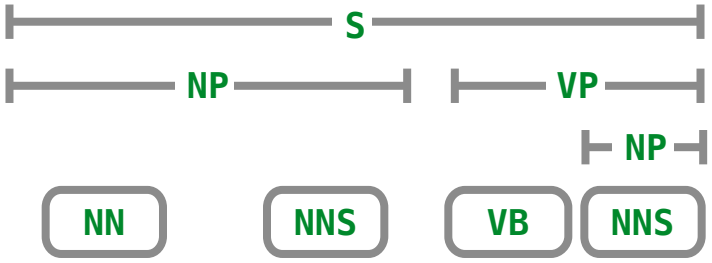


Rule frequency per 100,000 tags

S ⟶ NP VP	25372	NN ⟶	teacher	5
NP ⟶ NN NNS	1335	NNS ⟶	strikes	
VP ⟶ VB NP	6679	VB ⟶	idle	
NP ⟶ NNS	4282	NNS ⟶	kids	

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

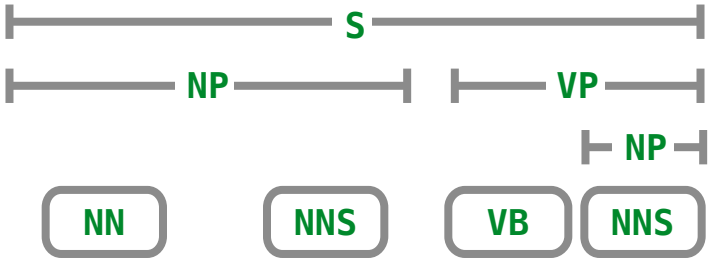


Rule frequency per 100,000 tags

S ⟶ NP VP	25372	NN ⟶	teacher	5
NP ⟶ NN NNS	1335	NNS ⟶	strikes	25
VP ⟶ VB NP	6679	VB ⟶	idle	
NP ⟶ NNS	4282	NNS ⟶	kids	

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

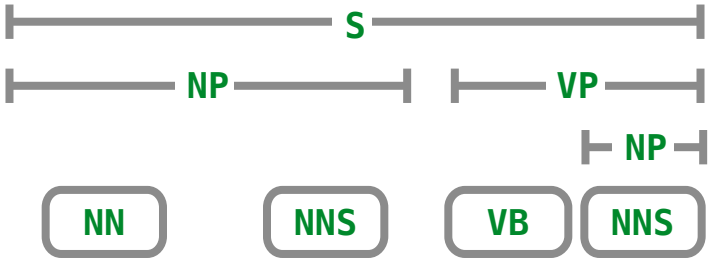


Rule frequency per 100,000 tags

S ⟶ NP VP	25372	NN ⟶	teacher	5
NP ⟶ NN NNS	1335	NNS ⟶	strikes	25
VP ⟶ VB NP	6679	VB ⟶	idle	26
NP ⟶ NNS	4282	NNS ⟶	kids	

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

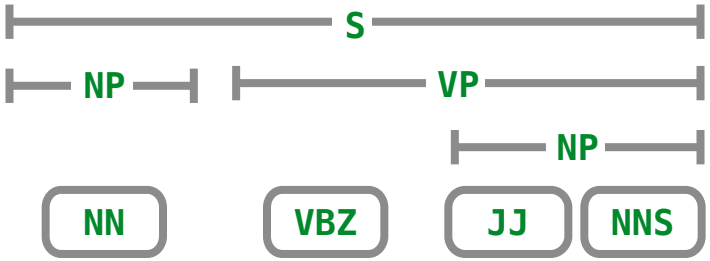


Rule frequency per 100,000 tags

S → NP VP	25372	NN →	teacher	5
NP → NN NNS	1335	NNS →	strikes	25
VP → VB NP	6679	VB →	idle	26
NP → NNS	4282	NNS →	kids	32

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

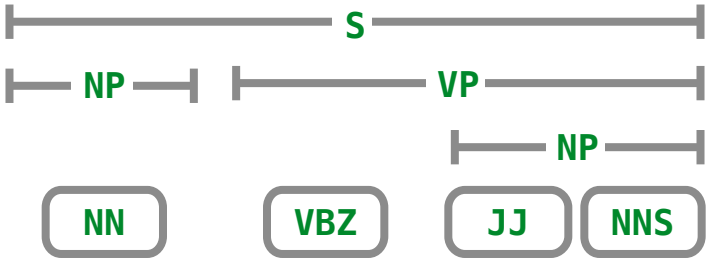


Rule frequency per 100,000 tags

S → NP VP	25372		NN →	teacher	5	
NP → NN	~~1335~~	4358	VBZ →	strikes	~~25~~	19
VP → VBZ NP	~~6679~~	3160	JJ →	idle	~~26~~	18
NP → JJ NNS	~~4282~~	2526	NNS →	kids	32	

Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common



Rule frequency per 100,000 tags

S ⟶ NP VP	25372		NN ⟶	teacher	5	
NP ⟶ NN	~~1335~~	4358	VBZ ⟶	strikes	~~25~~	19
VP ⟶ VBZ NP	~~6679~~	3160	JJ ⟶	idle	~~26~~	18
NP ⟶ JJ NNS	~~4282~~	2526	NNS ⟶	kids	32	

(Demo)