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

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

Programs must be written for people to read
Syntactic Ambiguity in English

Programs must be written for people to read

Preface of *Structure and Interpretation of Computer Programs* by Harold Abelson and Gerald Sussman with Julie Sussman
Programs must be written for people to read

\(^1\)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

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

\[
cows = \text{Leaf}('N', 'cows')
\]
\[
\text{intimidate} = \text{Leaf}('V', 'intimidate')
\]
\[
S, \ NP, \ VP = 'S', 'NP', 'VP'
\]
\[
\text{Tree}(S, [\text{Tree}(NP, [\text{cows}]),
\text{Tree}(VP, [\text{intimidate},
\text{Tree}(NP, [\text{cows}])]])])
\]
Grammars
Context-Free Grammar Rules

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

A Sentence ...

... can be expanded as ...

... a Noun Phrase then a Verb Phrase.

Grammar

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>NP</th>
<th>VP</th>
<th>N</th>
<th>V</th>
<th>buffalo</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>→</td>
<td>NP</td>
<td>VP</td>
<td>N</td>
<td>V</td>
<td>buffalo</td>
</tr>
<tr>
<td>NP</td>
<td>→</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>buffalo</td>
</tr>
<tr>
<td>N</td>
<td>→</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>buffalo</td>
</tr>
<tr>
<td>V</td>
<td>→</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>buffalo</td>
</tr>
</tbody>
</table>
Parsing
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

```
0
1
2
3
4

buffalo  buffalo  buffalo  buffalo

NP       S       VP
```
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

Constraint: A Leaf must match the input word

buffalo buffalo buffalo buffalo buffalo
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input

```
S
NP      VP
```

```
0  buffalo  1  buffalo  2  buffalo  3  buffalo  4
```
Exhaustive Parsing

Expand all tags recursively, but constrain words to match input
Learning

(Demo)
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

teacher strikes idle kids

Rule frequency per 100,000 tags

\[
\begin{align*}
S &\rightarrow NP \; VP & 25372 & NN &\rightarrow & teacher & 5 \\
NP &\rightarrow NN \; NNS & 1335 & NNS &\rightarrow & strikes & 25 \\
VP &\rightarrow VB \; NP & 6679 & VB &\rightarrow & idle & 26 \\
NP &\rightarrow NNS & 4282 & NNS &\rightarrow & kids & 32
\end{align*}
\]
Scoring a Tree Using Relative Frequencies

Not all syntactic structures are equally common

```
S → NP VP
NP → NN
VP → VBZ NP
NP → JJ NNS
```

```
teacher strikes idle kids
```

<table>
<thead>
<tr>
<th>Rule frequency per 100,000 tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow \text{NP} \ \text{VP}$</td>
</tr>
<tr>
<td>$\text{NP} \rightarrow \text{NN}$</td>
</tr>
<tr>
<td>$\text{VP} \rightarrow \text{VBZ} \ \text{NP}$</td>
</tr>
<tr>
<td>$\text{NP} \rightarrow \text{JJ} \ \text{NNS}$</td>
</tr>
</tbody>
</table>

(Demo)