Syntactic representation
Further reading
Beatrice Santorini
An introduction to syntactic theory
http://www.ling.upenn.edu/~beatrice/150/toc-long.html
Chapters 1–5.

Syntactic rules
So far we’ve been describing individual utterances one-at-a-time.

- Cats detest peas.
- Children eat tomatoes.
- Cheetahs chase gazelles.

Syntactic rules
But clearly there are more sentences than we can describe individually.

- Sentence template: Ns V Ns.
- Vocabulary size: N: 1000, V: 100
- Number of sentences: 100 million.

Syntactic rules
Anyway, we can understand sentences we’ve never heard before.

- Peas detest cats.
- Tomatoes eat children.
- Gazelles chase cheetahs.
Syntactic rules

At the same time, we know that other potential sentences are not part of English.

*Cats black detest peas green.
*Adopt which cat did your friend?
*Over there is the who I went to the party with guy.

In linguistics * labels ungrammatical sentences.

Our goal is to come up with representations and algorithms that characterize the sentences in a natural language, and what they mean.

Syntactic rules must be formulated in linguistically meaningful ways

To make a question in English:
Take the main auxiliary and move it to the beginning of the sentence (swap it with the subject of the sentence).

The girl is tall ->
Is the girl tall?
The red pig can stand on the house ->
Can the red pig stand on the house?

Impossible rule:
Take the first auxiliary and move it to the beginning of the sentence.

The boy who was holding the plate is crying
Was the boy who holding the plate is crying?
Is the boy who was holding the plate crying?

Linguistically meaningful rules must refer to

Syntactic constituency –
the natural ways words group together in sentences.

Syntactic relationships –
the kinds of meaningful combinations that link words together in sentences.

Constituency

What groups of words go together.

Evidence: substitution

A constituent is any syntactic unit; a single word is the smallest possible such unit; if a single word can substitute for a string, that’s evidence that the word and the string are constituents of the same category.
Constituency and substitution

The little boy fed the cat. -> He fed the cat.
Black cats detest green peas. -> They detest them.

Constituency

Constituency is a judgment about the intended analysis of particular utterance.

The little boy from next door fed the cat without a tail. -> *He from next door fed her without a tail.*
-> He fed her.
The little boy is sometimes a noun phrase, but not always.

Other kinds of constituents

There: prepositional phrases.
Put it on the table. -> Put it there.
Put it over on the table. -> Put it over there.
Put it over on the table. -> Put it there.

Put it over on the table that’s by the door -> *Put it there that’s by the door.*

Other kinds of constituents

So or it: full sentences
I know that they’re invited. -> I know it.
I imagine that they’re invited -> I think so.

Other kinds of constituents

So or it: full sentences
I know that they’re invited. -> I know it.
I imagine that they’re invited -> I think so.

Other kinds of constituents

do so: verb phrases
She will give you a back rub ->
She will do so.
The two boys could order tuna salad sandwiches ->
The two boys could do so.
Representing constituenthood

Tree diagrams – the graphic structure of a tree is a statement about how the speaker intends to group together the words into constituents (mentally).

Syntactic constituents are represented graphically as nodes in the tree.

Representing constituenthood

The secretary drafted the letter.

Need:
sentence: the secretary drafted the letter.
noun phrase: the secretary, the letter
verb phrase: drafted the letter.

Other constituent tests

Can something function as a sentence fragment in response to a question?

What do you like?
The cats.
Cats with long, fluffy tails.
The cats with long, fluffy tails.

Other constituent tests

Can you construct an it-cleft focusing a particular string of words?

Ordinary cats detest the smell of citrus fruits.
It is ordinary cats that detest the smell of citrus fruits.
It is the smell of citrus fruits that ordinary cats detest.

Caveats about constituency

Tests so far deal with phrasal constituents (which stand on their own).
Single words are lexical constituents (you can switch words, etc.), this can be different.
Caveats about constituency

Two kinds of verb phrase: finite and nonfinite.
Finite verb phrases carry tense, modality. She gave you a backrub.
Nonfinite verb phrases She will give you a backrub.
Only nonfinite verb phrases move, etc.

Constituenthood and rules

We’ve seen a close connection between substitution and constituency.

Syntactic rules should allow constituents to substitute, where possible.

Constituenthood and rules

Substitution sites in bold.

Constituenthood and rules

Substitution sites in bold.

Constituenthood and rules

This kind of structure is called an elementary tree.

Talking about trees

Anchor or head.
Talking about trees

Spine.

Sentence

Noun Phrase

Verb Phrase

Transitive Verb

drafted

Talking about trees

Root.

Sentence

Noun Phrase

Verb Phrase

Transitive Verb

drafted

Constituency and rules

We can define grammatical operations that combine together elementary trees.

Substitution: unify the root of one tree with a substitution site in another tree (with the same label).

Illustrating substitution

One elementary tree

Noun Phrase

Determiner

Noun

secretary

Illustrating substitution

Another elementary tree

Sentence

Noun Phrase

Verb Phrase

Transitive Verb

drafted

Illustrating substitution

Unify root of one tree with substitution site in another.

Sentence

Noun Phrase

Determiner

Noun

secretary

Transitive Verb

drafted
Tree Substitution Grammar

```
S               S
/ \           /  \\
NP  VP        NP  VP
|    |        |    |
|    |        |    |
\    \      \    \
V  D  N      V  D  N
|  barks     |  dog  barks
```

```
T₁ T₂ = + T₃
```

Dependency

```
S
/  \\
NP  VP
|    |
|    |
\    \\
D  N  V
|  the  |
|  dog  barks
```

```
T₁ T₂ = + T₃
```

Modification operations

There are lots of modifiers – modifying a constituent does not change its category.

The children ate the pizza.
The children ate the pizza with gusto.

Modification operations

You can add modifiers indefinitely.

We would drink lemonade.
We would drink lemonade in the summer.
We would drink lemonade in the summer on the porch.
We would drink lemonade in the summer on the porch with friends.
Possible modifications

Sister-adjunction

\[
\begin{array}{c}
X'' \\
\downarrow \\
X \\
\downarrow \\
word \\
\end{array}
\quad
\begin{array}{c}
X' \\
\downarrow \\
Y'' \\
\downarrow \\
W'' \\
\end{array}
\quad
\begin{array}{c}
X'' \\
\downarrow \\
X' \\
\downarrow \\
Z'' \\
\end{array}
\]

Possible modifications

General adjunction

\[
\begin{array}{c}
X'' \\
\downarrow \\
X' \\
\downarrow \\
Y'' \\
\downarrow \\
W'' \\
\end{array}
\quad
\begin{array}{c}
X' \\
\downarrow \\
X' \\
\downarrow \\
Z'' \\
\end{array}
\]

Modification issues

General adjunction:
more syntactically-motivated

The children ate the pizza with gusto.
The parents did so also.
The parents did so reluctantly.

Sister-adjunction
easier to implement
context-free
X-Bar theory

You need to generalize over elementary trees with similar structure.

(XTAG has thousands of trees for English!)

X-Bar theory

Start from here:

Sentence
  Noun Phrase  Verb Phrase
    Transitive Verb
      Noun Phrase
        eat

X-Bar Theory

This handles the sentence
The children ate the pizza.
But what about
The children ate.

X-Bar theory

Keep the structure in common:

Sentence
  Noun Phrase  Verb Phrase
    Transitive Verb
      Noun Phrase
        eat

X-Bar theory

The head of the tree determines the category of projections along the spine.

$V \Rightarrow V' \Rightarrow V''$

Complements come in at one bar level
Specifiers come in at two bar levels

X-Bar theory

Example

Sentence
  Noun Phrase  Verb Phrase
    Noun Phrase
      eat
**Parallels across categories**

**Predicate-argument structure**

The army destroyed the city.
the army’s destruction of the city.
The city was destroyed (by the army)
the city’s destruction (by the army)

**Modification**

She gives money to the organization on a regular basis.
She regularly gives money to the organization on a regular basis.
her gifts of money to the organization on a regular basis.
her regular gifts of money to the organization.

**General X–Bar schema**

Common underlying pattern for English

```
X''
Z'
X
X'

word
```

**Movement**

Elementary trees localize semantic and syntactic dependencies within a single constituent.

```
V'' / NP
V

Noun Phrase
```

**Movement – standard analysis**

Trees with gaps (marked \( f \)) that show dependency but indicate extraction, and slashes (categories \( X/Y \)) that remember what’s missing.

```
NP

V'' / NP
V

NP
```

In some constructions, though, dependent elements appear outside this constituent.

Contrast:
Bill knows Mary said the children *ate pizza*.
Bill knows *what* Mary said they *ate*.

Syntactic/semantic representations need to localize dependencies and encode word order.
Movement – standard analysis

*Filler* trees that require slash categories, and eliminate them:

```
      Q
     / \
    NP  V' / NP
     |   /  |
    N  NP  V  / NP
            /  |
            N  V
            /  |
            NP  / NP
            /  |
            N  V
            /  |
            NP
```

Movement – gap threading analysis

*Gap threading* trees that take slashes and pass them up:

```
      Q
     / \
    NP  V' / NP
     |   /  |
    V  NP  V' / NP
          /  |
          V  / NP
          /  |
          V  / NP
```

Movement – putting it all together

```
      Q
     / \  
    NP  V' / NP
     |   /  |
    N  NP  V  / NP
            /  |
            N  V
            /  |
            NP  / NP
            /  |
            N  V
            /  |
            NP
    Mary  said
    /    /  |
   N    V  / NP
       /  |
       N  / NP
       /  |
       V  / NP
```

Prolog Programming as Knowledge Representation

- Start by understanding the world.
  - Determine the **objects** that you need to consider; create Prolog terms for them.
  - Determine the **relationships** that you need to know about those objects; create Prolog predicates for them.
  - Develop a precise conceptualization that takes a stand on meaningful choices.
- State the facts.

Case study: linguistic structure

- **Objects:**
  - Categories
    - noun
    - verb
    - phrases
  - Words
    - the
    - dog
    - barks

          S
         /  
        NP  VP
        /  
      D    N  V
          /  
         the  dog  barks

Linguistic data structures

- **Lexical nodes:**
  - leaf(the)
  - leaf(dog)
  - leaf(barks)

          S
         /  
        NP  VP
        /  
      D    N  V
          /  
         the  dog  barks
**Linguistic data structures**

- **Internal nodes:**
  - `node(d, [leaf(the)])`
  - `node(n, [leaf(dog)])`
  - `node(v, [leaf(barks)])`
  - `node(np, [node(d, [leaf(the)]), node(n, [leaf(dog)])])`

**Computing with Trees**

- **New object**
  - Substitution site or gap with specific category
  - `gap(np)`

- **Sample structure**
  - `node(s, [gap(np), node(vp, [node(v, [leaf(barks)])])])`

**Computing with Trees**

- **New object**
  - Substitution site or gap with specific category
  - `gap(np)`

- **Sample structure**
  - `node(s, [gap(np), node(vp, [node(v, [leaf(barks)])])])`
Defining Substitution

New idea: two \textit{mutually recursive} procedures
Each represents the base case for the other.

For substitution, one procedure each for
Whole nodes
Lists of children

Defining Substitution

Overall goal:
Define \texttt{subst(Tree,Subtree,Result)}
Replace one gap(C) node in the passed
Tree with a new node(C,L) Subtree to obtain Result

Basic basic case
The passed Tree is itself gap(C)

Defining Substitution

Clause:
\begin{verbatim}
subst(gap(C),node(C,L),node(C,L)).
\end{verbatim}

Otherwise, if Tree is a node replace one of its children.

\begin{verbatim}
subst(node(X,K), S, node(X,R)) :- subst_l(K, S, R).
\end{verbatim}

Defining Substitution

Substitute somewhere in the list.

Base case: substitute in the head.
\begin{verbatim}
subst_l([Head|Tail],Subst,[R|Tail]):- subst(Head,Subst,R).
\end{verbatim}

Defining Substitution

Recursive case: keep the head and substitute elsewhere.

\begin{verbatim}
subst_l([Head|Tail],Subst,[Head|R]):- subst_l(Tail,Subst,R).
\end{verbatim}

Fancy illustration

\begin{verbatim}
? subst(node(s,[gap(np),
node(vp,[node(v,[leaf(saw)]),
gap(np)])),
node(np,[leaf(chris)]),
T).
\end{verbatim}
Step 1: Break down $S$

? subst(node(s, [gap(np), node(vp, [node(v, [leaf(saw)]), gap(np)])]), node(np, [leaf(chris)])), T).

=> ? subst_l([gap(np), node(vp, [node(v, [leaf(saw)]), gap(np)])], node(np, [leaf(chris)]), R).

Step 2, choice 1 – base case

$R = [\text{node}(np, [\text{leaf}(chris)]), \text{node}(vp, [\text{node}(v, [\text{leaf}(saw)]), \text{gap}(np)])]$
Finally, then

\[ R = \text{[gap(np),}\]
\[ \text{node(vp, [node(v, [leaf(saw)]),}\]
\[ \text{node(np, [leaf(chris)])])]}\]
\[ =\]
\[ T = \text{node(s,}\]
\[ \text{[gap(np),}\]
\[ \text{node(vp, [node(v, [leaf(saw)]),}\]
\[ \text{node(np, [leaf(chris)])])]}\]