1. Verb Phrases

English verb phrases consist of:
- a main verb
- any optional pre-modifiers
- and zero or more complements
- included are arguments and adjuncts.

2. Verb Phrases: pre-modifiers

We have to account for a range of structures ahead of the main verb:
- including adverbs, modal and auxiliary verbs.

3. After the verb: arguments vs. adjuncts

Arguments are post-verbal phrases that are tied very closely to particular classes of verbs:
- Different verbs require different numbers and kinds of arguments.

Adjuncts are post-verbal phrases that can occur with pretty much any verb:
- They’re largely optional.
- And if you have lots of them.

Adjuncts include:
- Adverb phrases that are like adverbs.
- Prepositional phrases that are like adverbial prepositions.

As just shorthand notation for the more verbose pair of notations.

4. Arguments and subcategorisation

We need rules for different patterns of arguments:

\begin{align*}
\text{intensive} & : \text{no components} \quad \text{intensive}^* \quad \text{intensive}^{**} \\
\text{transitive} & : \text{one AP complement} \quad \text{transitive}^* \quad \text{transitive}^{**} \\
\text{intransitive} & : \text{zero AP and one PP} \quad \text{intransitive}^* \quad \text{intransitive}^{**} \\
\text{allograph} & : \text{two AP complements} \quad \text{allograph}^* \quad \text{allograph}^{**}
\end{align*}

Not all verbs are allowed to participate in all these VP rules.

We subcategorise verbs in a language according to the sets of VP rules they participate in.

This is a modern take on the traditional notion of transitive/intransitive.

Modern grammars may have 100s of subcategorisation classes.

5. Subcat examples and counterexamples

Some examples of the diversity of complement patterning:

\begin{align*}
\text{I was told (that) KLM has a flight.} \\
\text{I prefer KLM has a flight.} \\
\text{I prefer to leave earlier.} \\
\text{Give me a cheaper fare.} \\
\text{Give a cheaper fare to my children.} \\
\text{Give with a flight.} \\
\text{Can you help me with a flight?} \\
\text{Please find a flight to Edinburgh.}
\end{align*}

You can understand such an extension to CFGs in one of two ways:
- Either as an extension to CFGs.
- Or as an extension to PDA, called Pushdown Automata.

For something Recursive Transition Networks.

6. Overly complicated, and wrong as well?

(If we go on to agreement, a brief diversion)

You might fear that all these (mostly binary) rules are missing the point:
- Syntactically, because they do allow all kinds of wrong orders.

\begin{align*}
\text{I saw (that) John hit (the) ball.} \\
\text{I saw (the) ball hit (by) John.}
\end{align*}

Why don’t we just make the order explicit?

\begin{itemize}
\item \text{vp: \text{John} \text{hit} \text{the} \text{ball}.}
\item \text{vp: \text{the} \text{ball} \text{hit} \text{by} \text{John}.}
\end{itemize}

Where by \text{eg.} “\text{Det}?” is intended the Det is optional.

7. Extending CFGs

You can understand such an extension to CFGs in one of two ways:
- As a change to the formation itself, i.e. the proper description involves adding to a CFG.
- As a change to the notation only, not to the formation as such.

\begin{align*}
\text{Verbal} \rightarrow \text{Modal? Aux? AdvP? V} \\
\text{Verbal} \rightarrow \text{Adjunct?} \text{V} \\
\text{Verbal} \rightarrow \text{Adv Verbal} \\
\text{Verbal} \rightarrow \text{aux? Modal Verbal} \\
\text{Verbal} \rightarrow \text{Adv Verbal}
\end{align*}

But where did we drash? AKA its position on the Chomsky hierarchy.

Modern grammars may have 100s of subcategorisation classes.

8. Infinite CFGs

Including infinite classes in our notation for the right-hand side of rules turns out to have a curious consequence.

If we take the same approach as we did for question-mark:

\begin{align*}
\text{I saw (that) John hit (the) ball.} \\
\text{I saw (the) ball hit (by) John.} \\
\text{I saw (the) ball hit (by) John.}
\end{align*}

we have what amounts to a notation for a CFG with an infinite number of rules.

That actually has the potential to change the status of the formation to weak generative capacity.

And its position in the Chomsky hierarchy.
9. Agreement
Agreement: when constraints hold among constituents that take part in a rule or set of rules. For example, in English, as in many other languages, determiners and the head nouns in NPs have to agree in number.

10. The agreement problem for CFGs
Our earlier NP rules are clearly deficient since they don’t capture this constraint.

11. Overgeneration
The NP and VP rules we've seen so far overgenerate:
- They permit the presence of strings containing:
  - Determiners and nouns that don’t go together
  - Verbs and arguments that don’t go together
This may not seem to be a problem if we’re only ever interested in parsing:
- As opposed to generation
But it has a nasty side-effect even for parsing:
- It will introduce spurious ambiguity
- We’ll come back to that when we talk more about ambiguity and parsing

12. Possible CFG Solution for Agreement
We could try to address our agreement problems by expanding the non-terminal categories to encode agreement:

```
NP & sg → CNP & sg
CNP & sg → Det & sg CNP & sg
NP & pl → CNP & pl
CNP & pl → Det & pl CNP & pl
S & sg → NP & sg VP & sg
S & pl → NP & pl VP & pl
VP & pl → V & pl NP
VP & sg → V & sg NP
```

Where we've used ‘sg’ and ‘pl’ for singular and plural

13. CFG Solution for Agreement
Good thing:
- It works and stays within the power of CFGs

Less good things:
- It’s inelegant
- It doesn’t scale
  - The interaction among various families of constraints explodes the number of categories and rules in the grammar
  - It still overgenerates!
- It still introduces spurious ambiguity

14. CFG conclusions
CFGs appear to be just about what we need to account for a lot of basic syntactic structure in English.

But there are problems:
- Overgeneration
- Agreement
- Unbounded dependencies

There are more elegant solutions:
- But they go beyond the formal power of CFGs
  - Adjunct expansions on VCs
  - Ego-based theories (GPSG, HPSG)
  - Tree-adjoining grammars

A compromise approach is to expand our approach to categories:
- By adding features