# Accelerated Natural Language Processing 2018

# Lecture 12: More CF rules for English, agreement

Henry S. Thompson
Drawing on slides by Philip Koehn, Jurafsky and Martin 2009
11 October 2018


## 1. Verb Phrases

English verb phrases consist of

- some optional pre-modifiers
- a main verb
  - which we will once again call the **head**
- and zero or more **complements**
  - Divided into **arguments**
  - and **adjuncts**

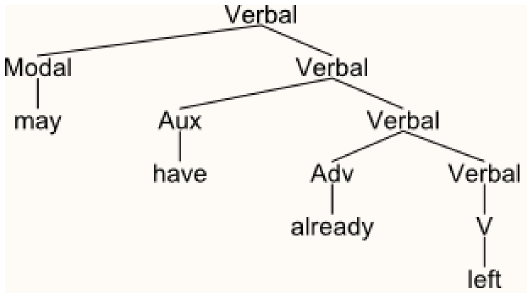
## 2. Verb Phrases: pre-modifiers

We have to account for a range of structures ahead of the main verb

- Including adverbs, modals and auxiliary verbs

| | |
|---|---|
| leave | Verbal → V |
| may leave | Verbal → Modal Verbal |
| has left | Verbal → Aux Verbal |
| suddenly left | Verbal → Adv Verbal |

We get a familiar-looking right-branching structure when these combine




# 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 *always* optional
- And you can have lots of them

Adjuncts include

- adverbs
- prepositional phrases that are *like* adverbs
    - Expressing time, place, manner, . . .

Adjuncts come *after* arguments


# 4. Arguments and subcategorisation

We need some rules for different patterns of arguments:

| | | | |
|---|---|---|---|
| **intransitive** | (no complements) | disappear | VP → Verbal |
| **transitive** | (one NP complement) | prefer a morning flight | VP → Verbal NP |
| **transitive + indirect object** | (one NP and one PP) | give a book to Ro | VP → Verbal NP PP |
| **ditransitive** | (two NP complements) | buy Robin a ticket | VP → Verbal NP NP |

Not all verbs are allowed to participate in all the 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

- Sometimes called subcategorisation **frames**

# 5. Subcat examples and counterexamples

Some examples of the diversity of complement patterning

```
John sneezed
Please find a flight to Edinburgh
Can you help me with a flight
Give me a cheaper fare
Give a cheaper fare to my children
I prefer to leave earlier
I was told (that) KLM has a flight
```

And some counterexamples

```
*John sneezed the book
*I prefer KLM has a flight
*Give with a flight
```

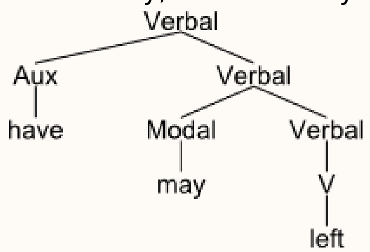
As with agreement phenomena, we need a way to formally express the constraints

# 6. Overly complicated, and wrong as well?

[Before we go on to agreement, a brief diversion]

You might feel that all these (mostly binary) rules are missing the point

- Particularly, because they allow all kinds of wrong orders




Why don't we just make the order explicit?

```
CNP → Det? Card? Ord? Quant? AP* Nominal
```

```
Verbal → Modal? Aux? AdvP? V
```

where by e.g. "Det?" is meant the Det is optional
and the "*" is a Kleene star, i.e. 0 or more APs are allowed

That is, why not, for the right hand side of rules,

- instead of *sequences* drawn from $T \cup NT$

- allow *regular expressions* over $T \cup NT$

We could, and people have

- Either as an extension to CFGs
- Or as an extension to FSAs, called **Pushdown Automata**
    - Or sometimes **Recursive Transition Networks**

# 7. Extending CFGs

You can understand such an extension to CFGs in one of two ways:

- As a change to the formalism itself, i.e.
    - **rhs** a regular expression whose alphabet is $T \cup NT$
    - corresponding (non-trivial) changes to the rewriting and node-admissibility definitions
- As an extention to the notation *only*, not to the formalism as such
    - I.e., we treat rules notated like so:
    - $X \rightarrow \ldots_1\ Y?\ \ldots_2$
    - As just shorthand notation for the more verbose pair of notations

      $X \rightarrow \ldots_1\ Y \ldots_2$

      $X \rightarrow \ldots_1\ \ \ldots_2$

On this account, our VP 'rule' on the previous slide is a shorthand notation for *eight* actual rules

What about the NP rule, with its Kleene star?

# 8. Infinite CFGs

Including Kleene star in our notation for the right-hand side of rules turns out to have a surprising consequence

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

- I.e., we treat rules notated like so:
- $X \rightarrow \ldots_1\ Y^*\ \ldots_2$
- As just shorthand notation for the more verbose pair of notations

  $X \rightarrow \ldots_1\ \ \ldots_2$

  $X \rightarrow \ldots_1\ Y\ Y^*\ \ldots_2$

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 formalism
    - Its **weak generative capacity**
    - AKA its position on the **Chomsky hierarchy**

[End of diversion]

# 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

- this flight *this flights
- *those flight those flights

# 10. The agreement problem for CFGs

Our earlier NP rules are clearly deficient since they don't capture this constraint

```
NP → Det Nominal
```

- That rule accepts, and assigns correct structures, to grammatical examples (this flight)
- But also accepts incorrect examples (*these flight)

Such a rule is said to **overgenerate**

# 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 often introduce **spurious ambiguity**
- We'll come back to that when we talk more about ambiguity and parsing

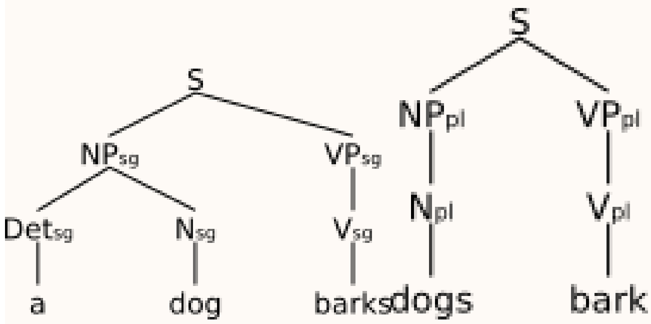
# 12. Possible CFG Solution for Agreement

We could try to address our agreement problems by expanding the non-terminal categories to encode agreement:

$NP_{sg} \rightarrow CNP_{sg}$
$CNP_{sg} \rightarrow Det_{sg} CNP_{sg}$
$NP_{pl} \rightarrow CNP_{pl}$
$CNP_{pl} \rightarrow Det_{pl} CNP_{pl}$
$S_{sg} \rightarrow NP_{sg} VP_{sg}$
$S_{pl} \rightarrow NP_{pl} VP_{pl}$
$VP_{pl} \rightarrow V_{pl}\ NP$
$VP_{sg} \rightarrow V_{sg}\ NP$

- Where we've used 'sg' and 'pl' for singular and plural
- And the above isn't enough: more doubling of rules would be needed
    - E.g. for Det

This gives us trees for *a dog barks* and *dogs bark*, but not for e.g. *dogs barks*




We could use the same approach for all the verb/VP classes

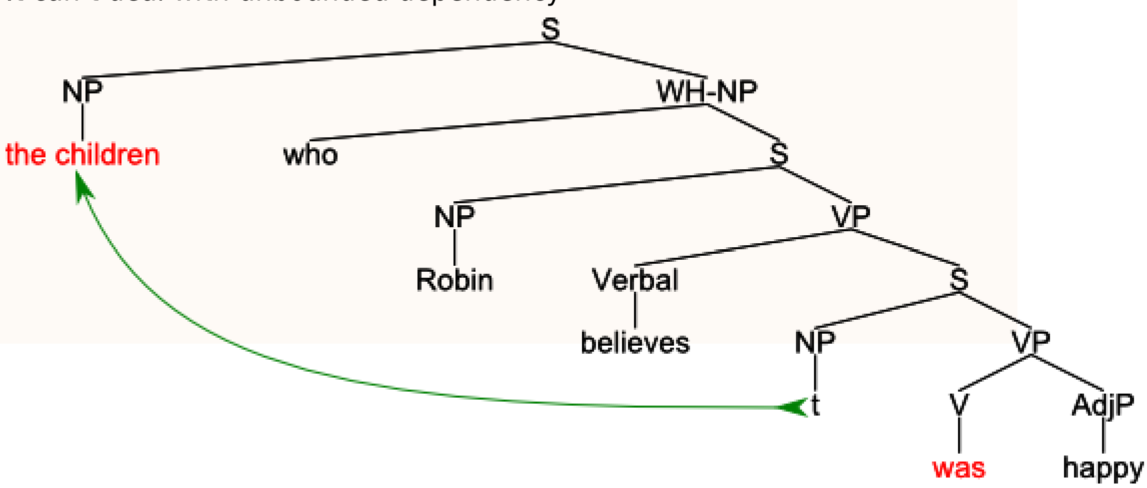
- But this clearly has become quite obscure
- And the (multiplicative) interaction *between* number agreement and subcategorisation will make things *much* worse

# 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 can't deal with unbounded dependency



    - Where we use 't' (for "trace") as the missing plural subject of "was happy"

# 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
    - Regular expressions on the RHS
    - Sign-based theories (GPSG, HPSG)
    - Tree-adjoining grammars

A compromise approach is to expand our approach to categories

- By adding **features**