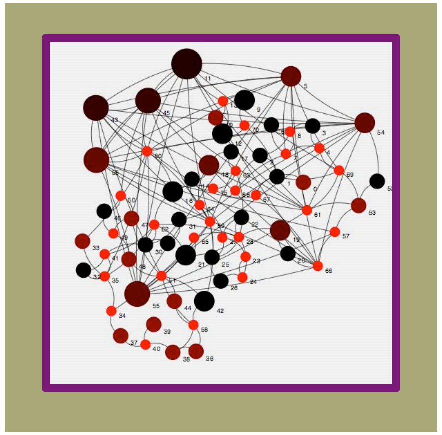
COMP40020 Human Language Technologies

Grammars, Rules and Parsers

February 2019



Prof. Julie Berndsen
School of Computer Science

Julie.Berndsen@ucd.ie

+ HLT6

Contents:

- Introduction to Syntactic Parsing
- Definitions and Examples

Aim:

- To give a brief introduction to language parsing strategies based on constituency and dependency. We will look at practical examples using NLTK at Workshop 2 on Wednesday.

+ HLT6

Contents:

- Introduction to Syntactic Parsing
- Definitions and Examples

Aim:

- To give a brief introduction to linguistic parsing strategies based on constituency and dependency. We will look at practical examples using NLTK at Workshop 2 next Monday. There will be a short assignment based on the workshop [illegible] worth 5%.

Primarily knowledge-based approach

Analysis & Generation

- Analysis
  - a representation constructed while checking whether a sentence can be accepted or rejected is an **analysis** (→ derivational history)
- Generation
  - a string is said to be generated by a grammar if there is a derivation of the grammar leading to it (→ can be several distinct derivations)

Grammar & Lexicon

- Grammar
  - representation
    (→ for now, CF-phrase structure rules)
  - acquisition

- Lexicon
  - representation
    (→ for now, terminal vocabulary V_T)
  - acquisition

+ Grammar & Lexicon

- Grammar
  - representation

    (→ for now, CF-phrase stru
  - acquisition

Syntactic Rules

S → NP VP

NP → Det N

NP → Det Adj N

VP → V NP

- Lexicon
  - representation

    (→ for now, termina
  - acquisition

Lexicon

Det → the

N → author | novel |orange | rabbit | cat

Adj → orange

V → wrote | chased

+ Parsing as Search

- Syntactic parsing is often regarded as searching through the space of all possible parse trees to find the correct parse tree for the sentence.

→Search space of possible parses is defined by the grammar

→The goal is to find all trees whose root is S and which cover exactly the words in the input

→Regardless of the search algorithm, there are two kinds of constraints that can help guide the search

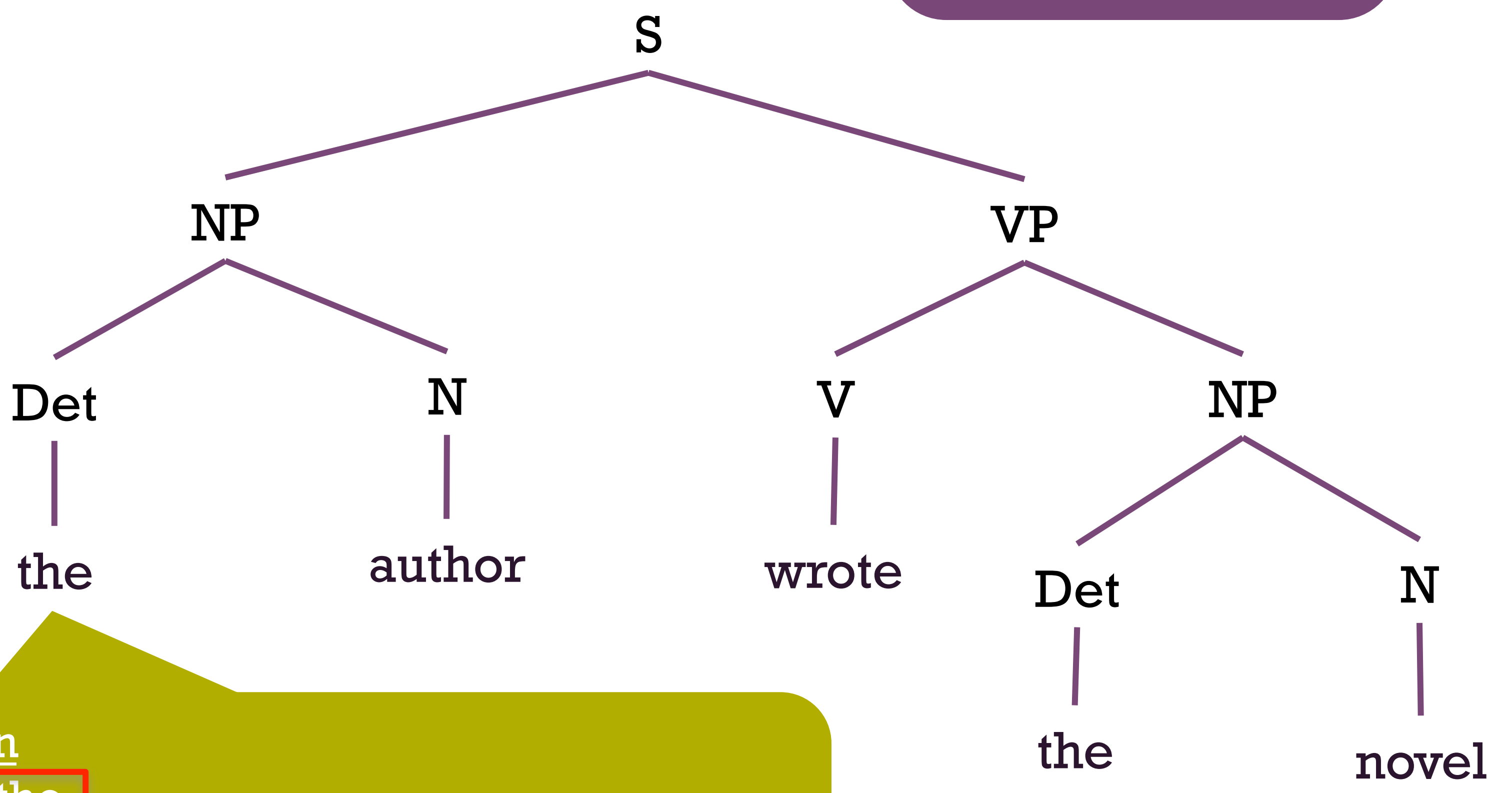
  → Using the input (→ bottom-up → input-driven)

  → Using the grammar (→ top-down → goal-driven)

(Jurafsky & Martin, 2008)

+ Top-Down Parsing

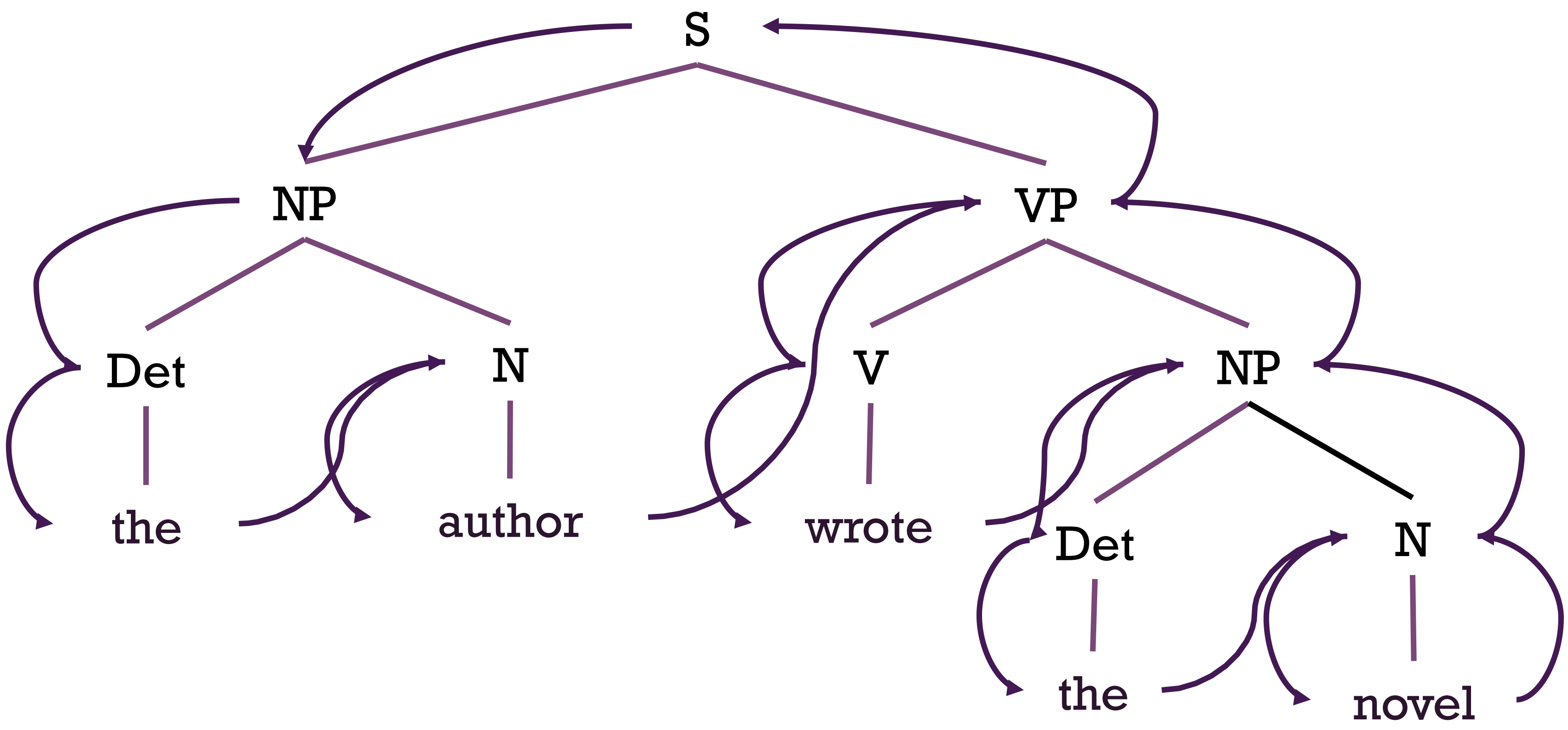
Syntactic Rules
S → NP VP
NP → Det N
NP → Det Adj N
VP → V NP



Lexicon
Det → the
N → author | novel | orange | rabbit | cat
Adj → orange
V → wrote | chased

Top-Down Parsing

- Top-down, depth-first, left-to-right



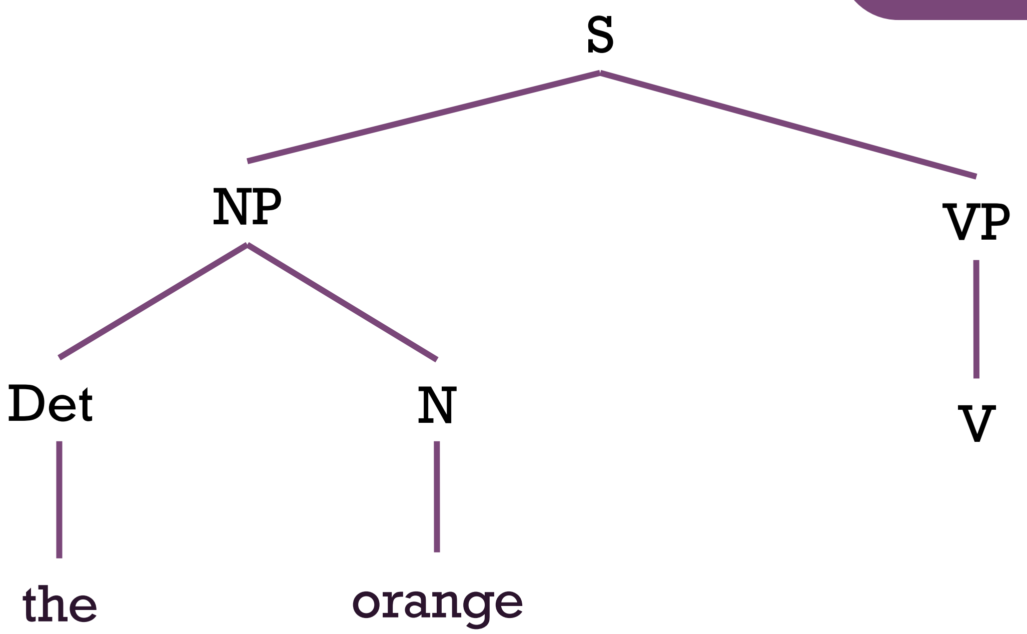
- Starting with the grammar

+ Top-Down Parsing

- *the orange rabbit* chased th

Syntactic Rules

S → NP VP
NP → Det N
NP → Det Adj N
VP → V NP

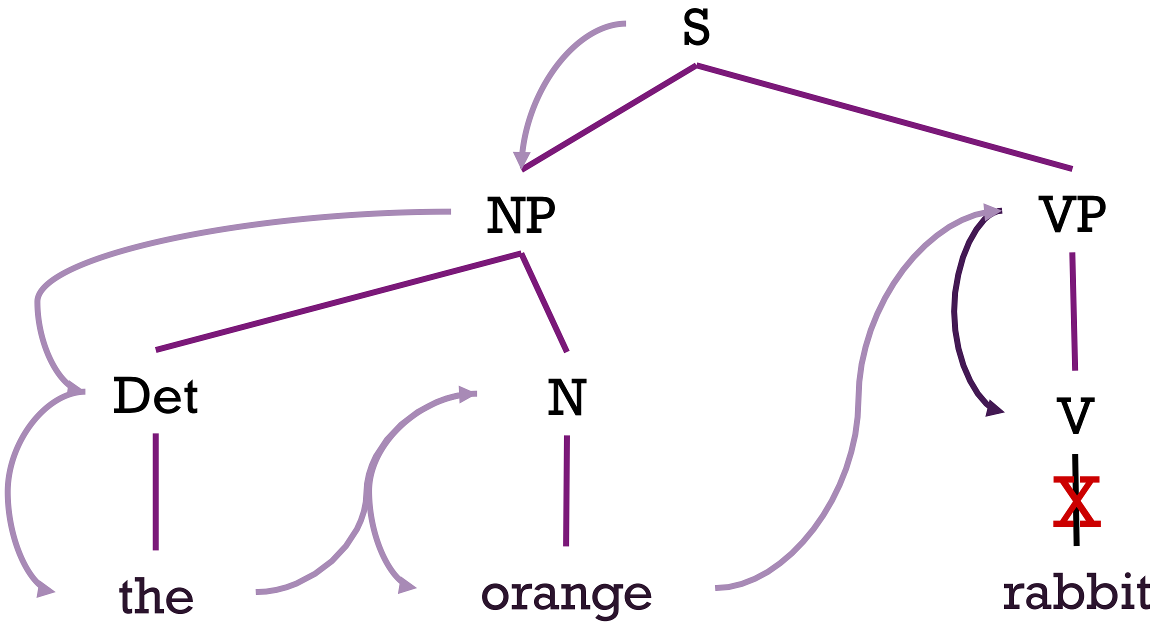


Lexicon

Det → the
N → author | novel | orange | rabbit | cat
Adj → orange
V → wrote | chased

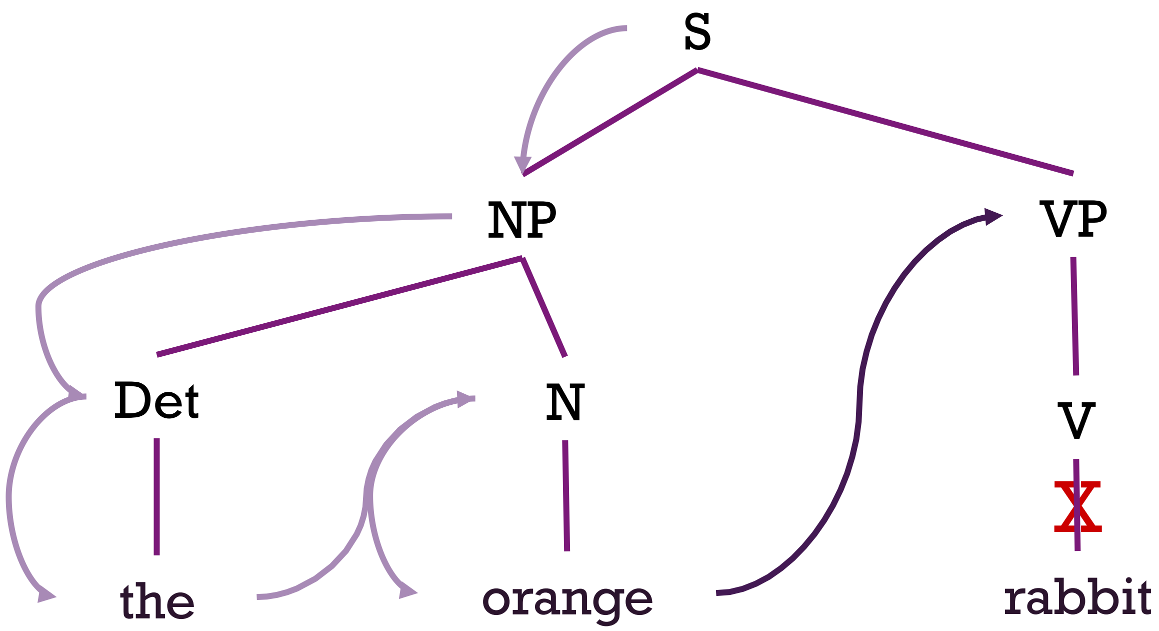
+ Top-Down Parsing

- *the orange rabbit* chased the cat



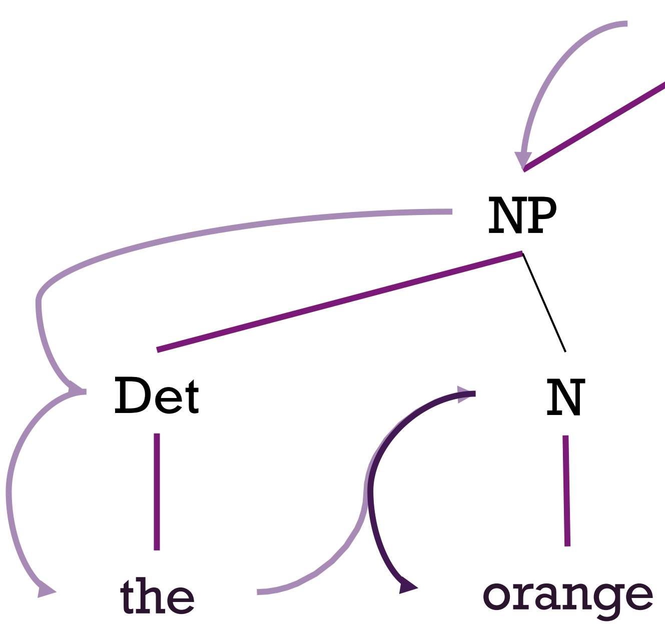
Top-Down Parsing

- *the orange rabbit* chased the cat



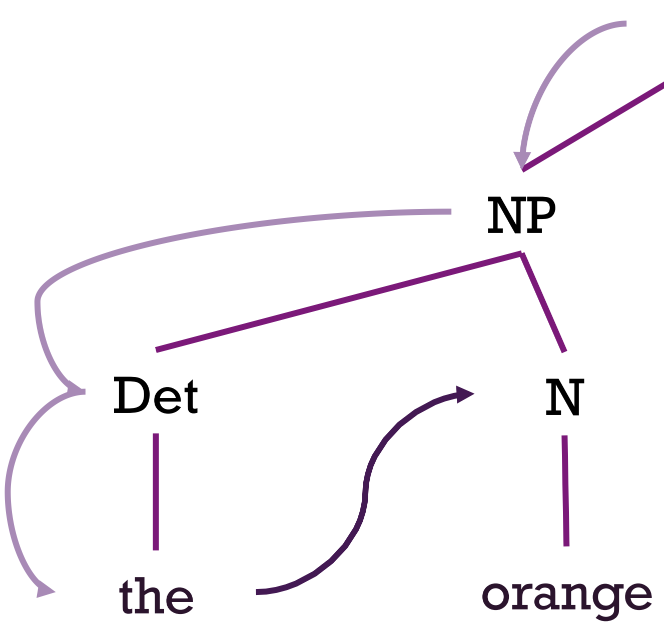
+ Top-Down Parsing

- *the orange rabbit* chased the cat



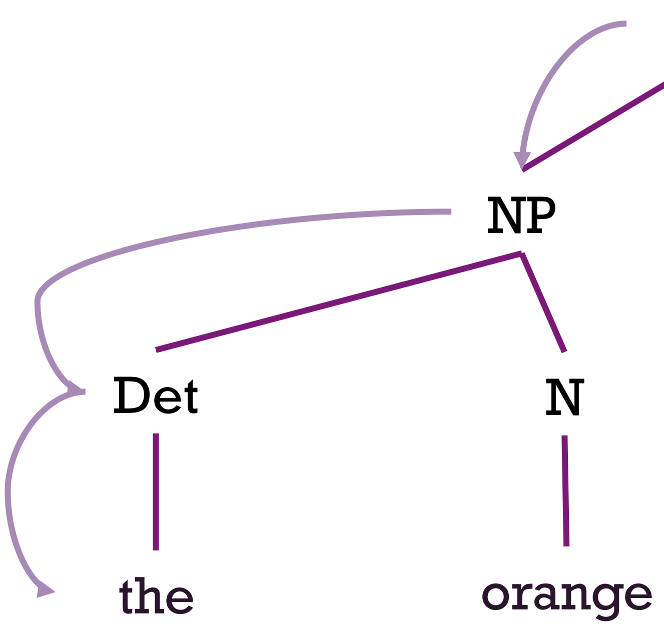
+ Top-Down Parsing

- *the orange rabbit* chased the cat



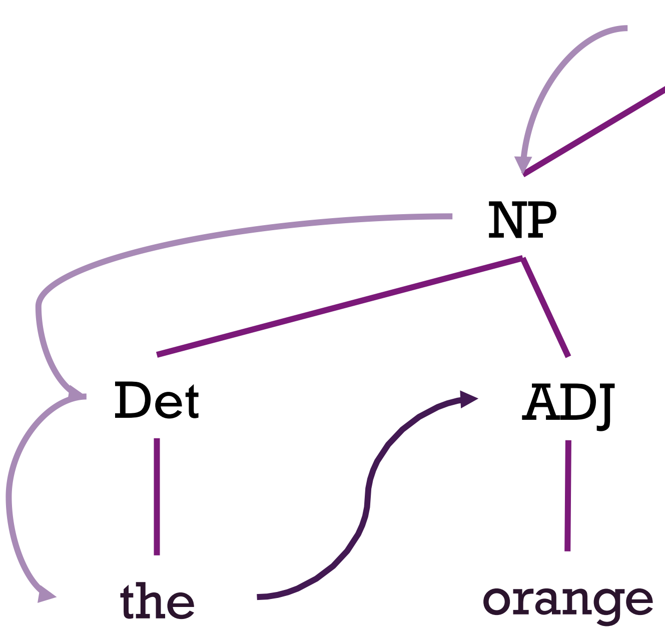
Top-Down Parsing

- *the orange rabbit* chased the cat



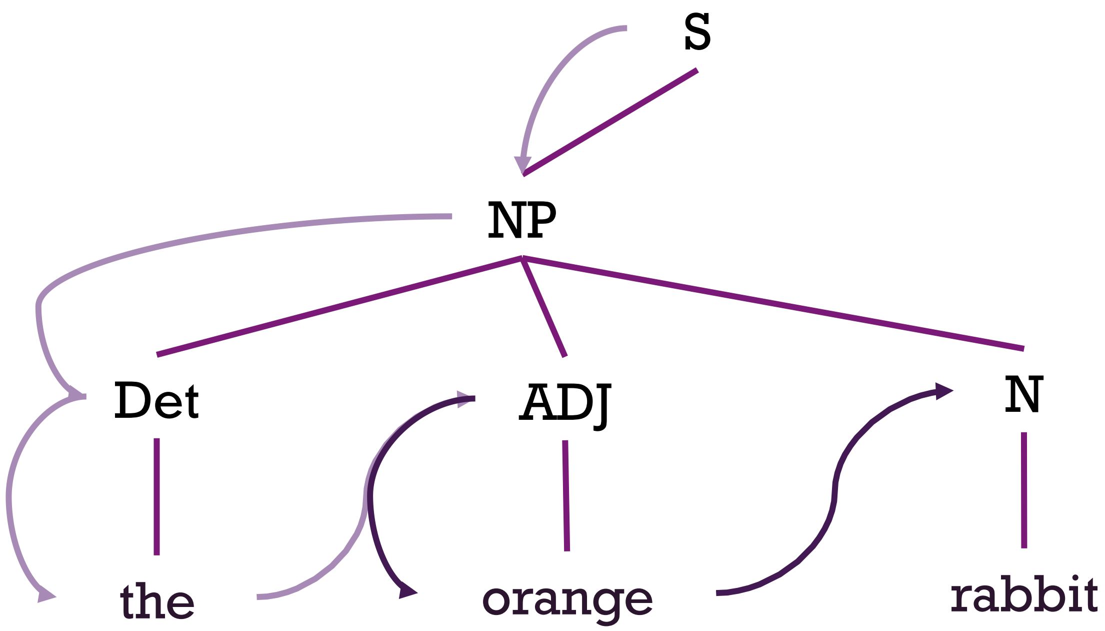
Top-Down Parsing

- *the orange rabbit* chased the cat



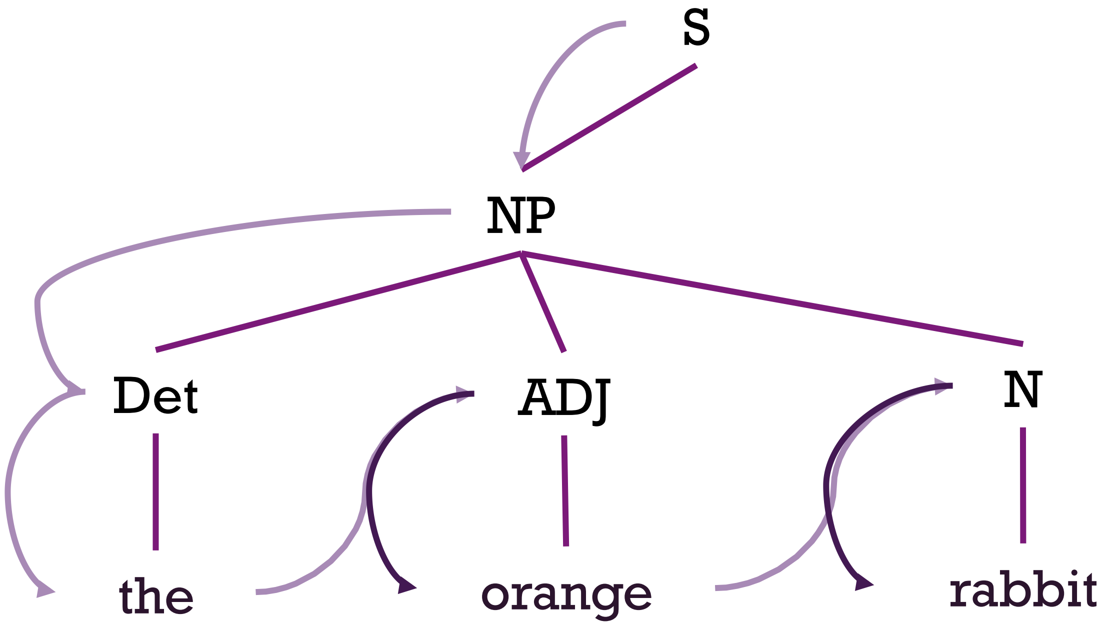
+ Top-Down Parsing

- *the orange rabbit* chased the cat



Top-Down Parsing

- *the orange rabbit* chased the cat



→ backtracking

Top-Down Parsing

Advantages:

- Never wastes time exploring trees that cannot result in an S
- Never explores trees which cannot find a place in some S-rooted tree

But:

- Spends considerable time on S trees which are not consistent with the input
- Has problems with coordination ambiguity
  - John talked to the novelist and the politician last week
  - John talked to the novelist and the politician wasn’t happy

+ Top-Down Parsing

Problems:

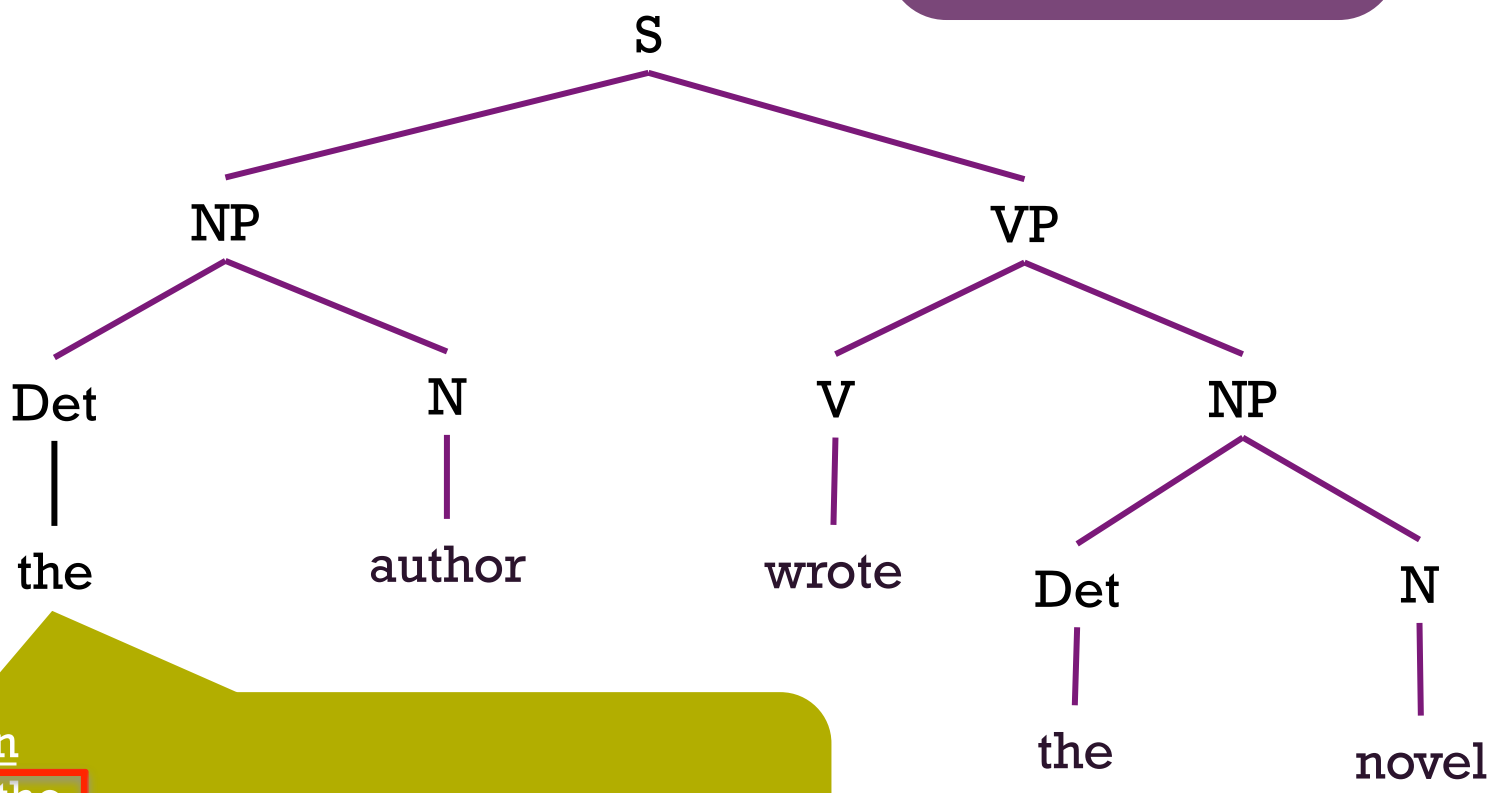
- ***recursive*** rules e.g. NP → NP CONJ NP
  - Options:
    - rewrite the rules
    - change the strategy to one which is not troubled by them (→ bottom-up)
- processor commits itself to a particular analysis before attempting to connect up words of the input string
  - Option
    - change the strategy (→ left-corner)

+ Bottom-Up Parsing

- Bottom-up

Syntactic Rules

S → NP VP
NP → Det N
NP → Det Adj N
VP → V NP

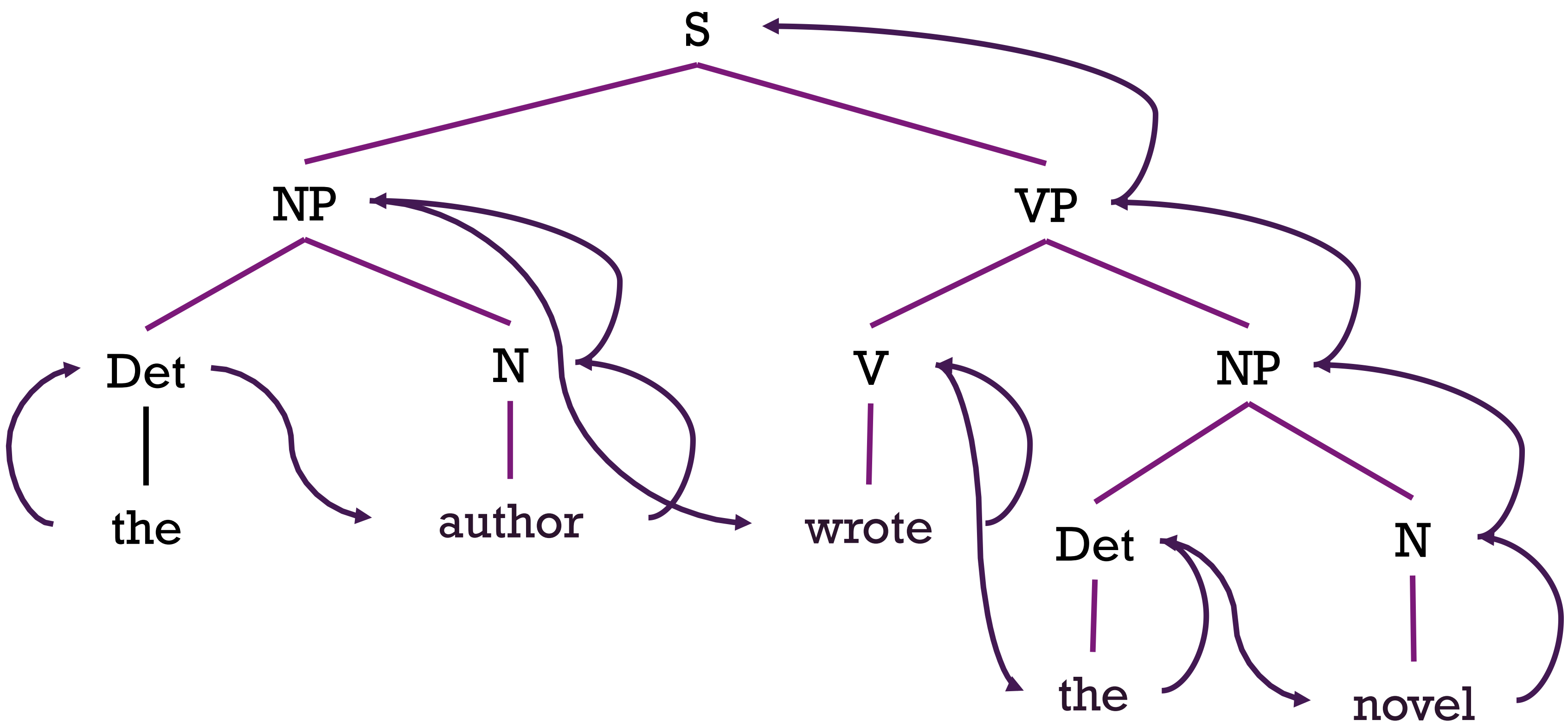


Lexicon

Det → the
N → author | novel | orange | rabbit | cat
Adj → orange
V → wrote | chased

+ Bottom-Up Parsing

- Bottom-up



- Starting with the words

+ Top-Down vs. Bottom-Up

- One problem with top-down parsing strategy was that the parser commits to a particular analysis before connecting up the words of the input string.

- Incorrect guesses can be recovered through backtracking but this process can involve a great deal of futile computation as structures are pursued that could never be supported by the input.

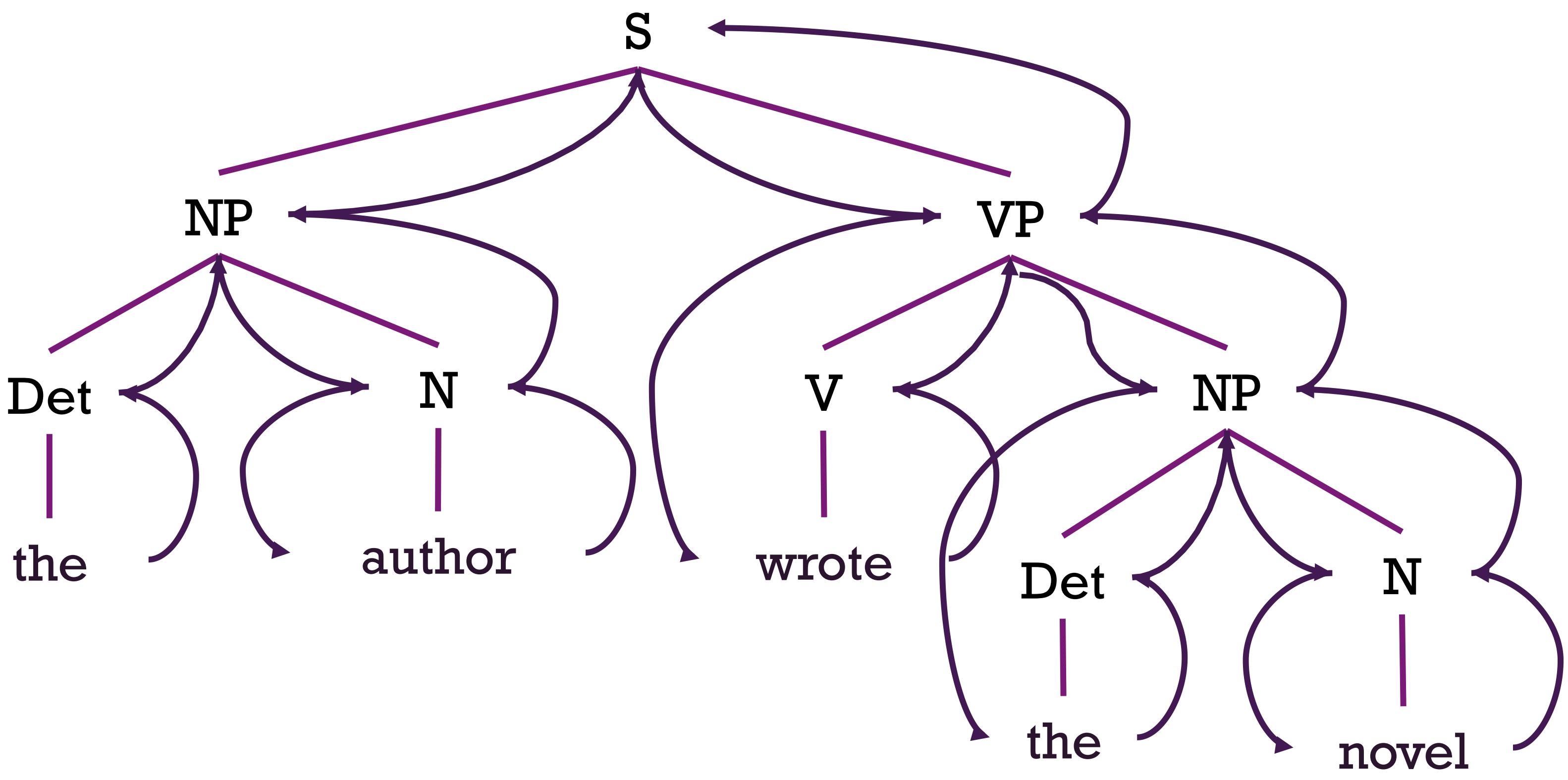
Top-Down vs. Bottom-Up

- **Bottom-up** parsing does not get misled this way as it is input-driven.

- But a bottom-up strategy can also result in much fruitless processing if analyses are attempted which could never be true **in that particular context.** (→ ambiguous words: can, play, orange)

- It is possible to temper a blind bottom-up strategy with a **degree of top-down control** (bottom-up parsing with top-down lookahead → left-corner)

+ Left-Corner Parser



Left-Corner Parsing

- Behaves sensibly on left- and right branching structures

But

- still has problems with centre-embedding (as do humans!)
  - E.g. a man that a woman that a child that a bird that I heard saw knows loves…
- unlike the top-down parser, the structure it builds is not always completely connected.
- unlike the bottom-up parser, it predicts phrases just on the basis of the left corner which can be wrong.

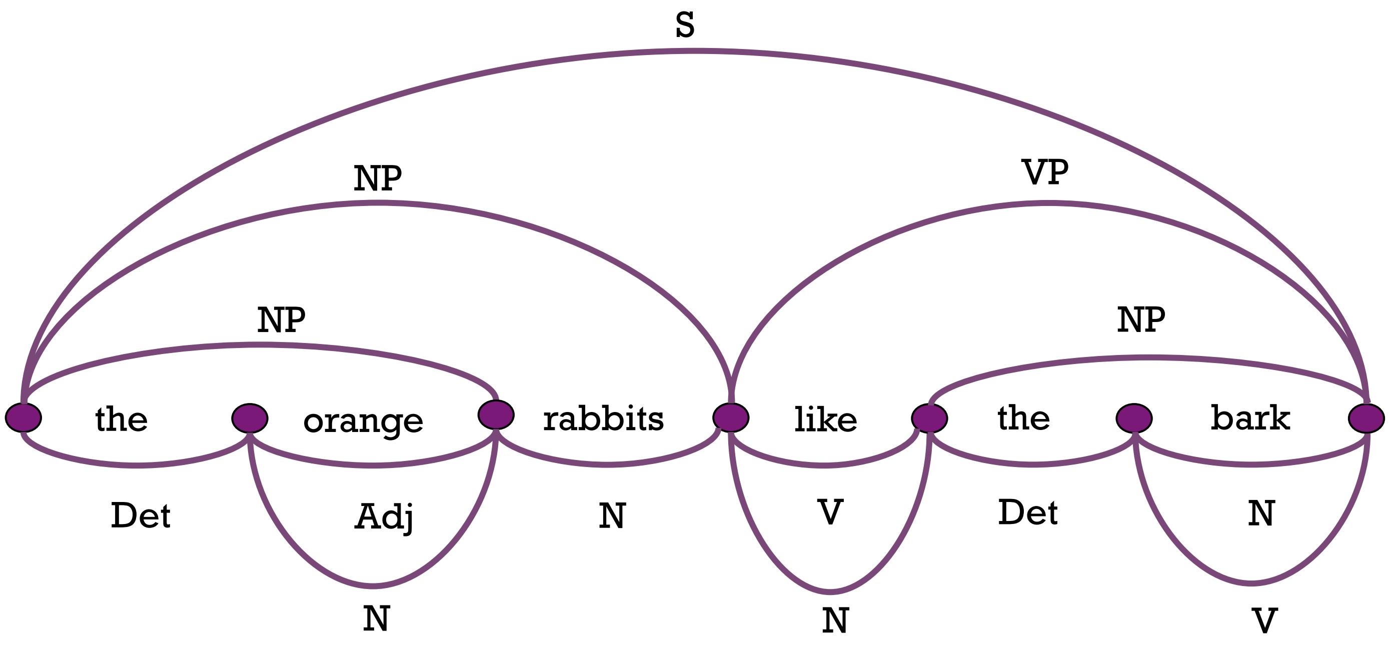
Chart Parsing

- One of the problems of relying on backtracking to recover from errors parsing is that the parser forgets everything it has found between the backtracking points

→ rediscovering previously recognised lexical items or phrases

Chart Parsing

- A solution is to keep track of whatever has already been processed in a **chart**.



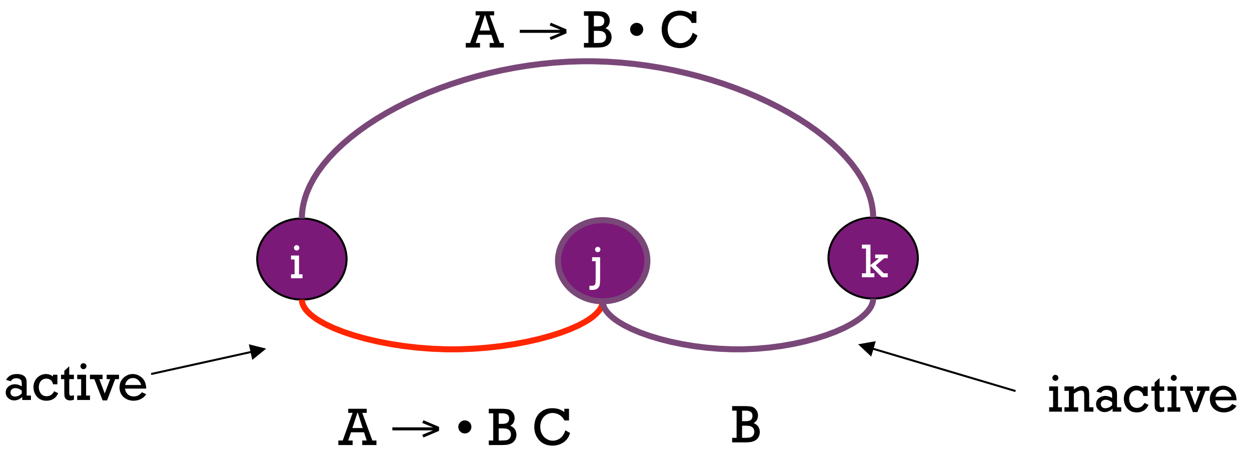
- The nodes (vertices) represent the gaps between the words

- The edges (arcs) represent the lexical or phrasal categories.

Chart Parsing

- Processing using a chart involves interaction between active (incomplete) and inactive (complete) edges which leads to the introduction of a new edge into the chart

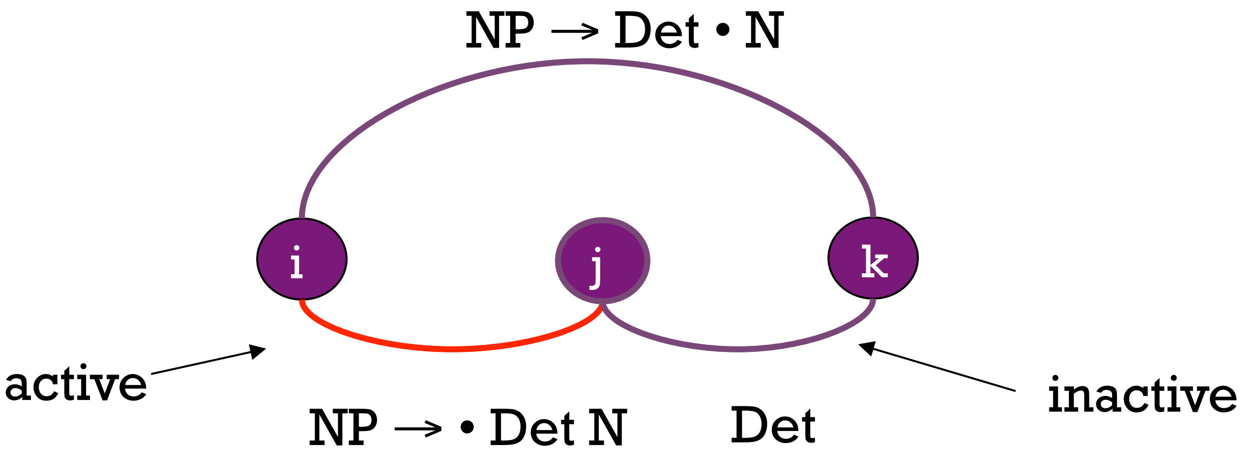
→ Fundamental Rule



+ Chart Parsing

- Processing using a chart involves interaction between active (incomplete) and inactive (complete) edges which leads to the introduction of a new edge into the chart

→ Fundamental Rule



+ Chart Parsing

- Assuming the following grammar and lexicon:

Syntactic Rules

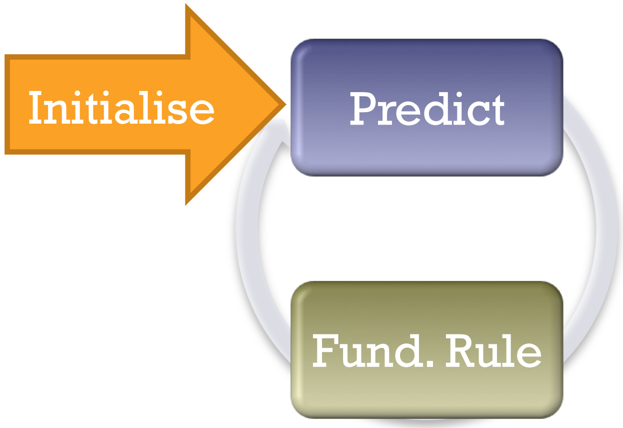
S → NP VP
NP → Det N
NP → Det Adj N
VP → V NP

Lexicon

Det → the
N → orange | rabbits | bark | like
Adj → orange
V → like | bark

+ Chart Parsing

- Bottom-Up
  - Initialisation rule
  - Predict rule
  - Fundamental rule



- Top-Down
  - Initialisation rule
  - Expand rule
  - Match rule
  - Fundamental rule

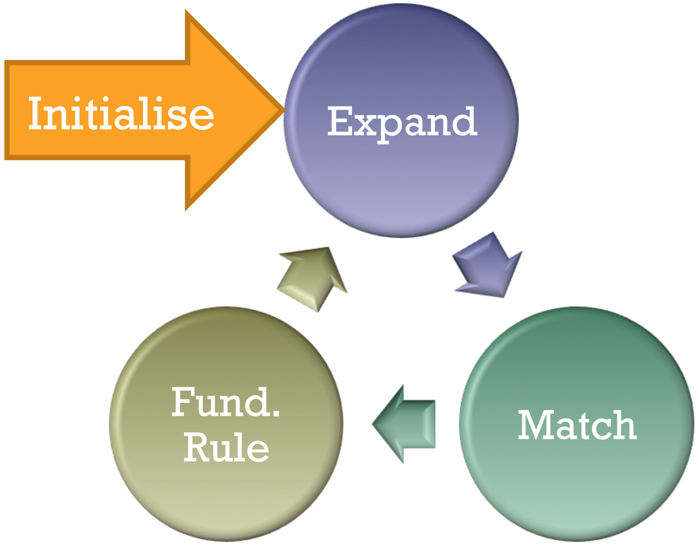


Chart Parsing

- Bottom-Up Initialisation:
  - add an inactive (completed) edge for each of the lexical items

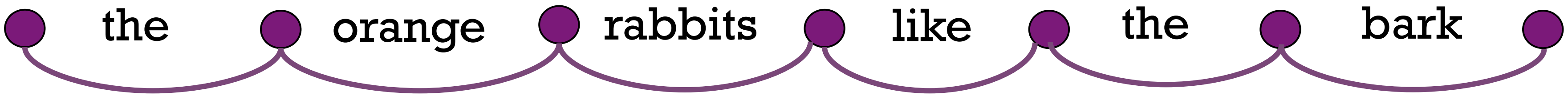


Chart Parsing

- Bottom-Up Prediction:
  - For each complete edge, add an active (incomplete) edge based on the lexicon and grammar

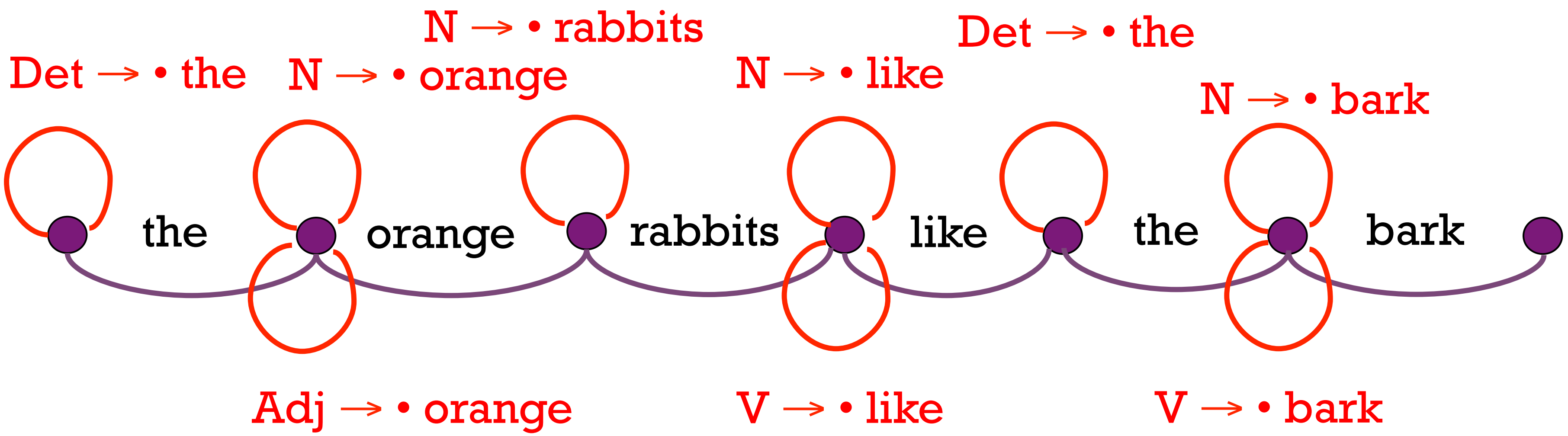
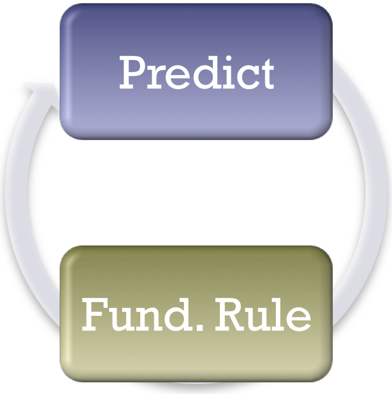


Chart Parsing

- Apply Fundamental Rule



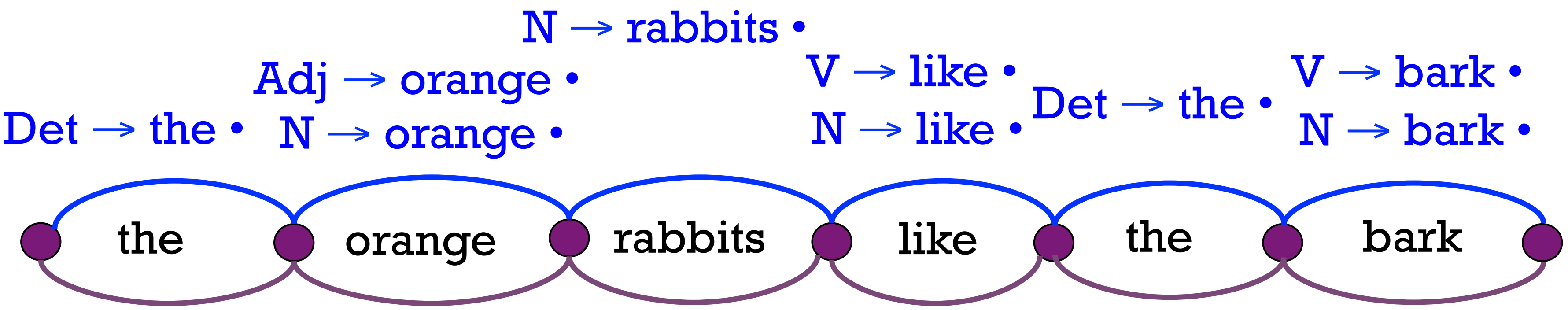
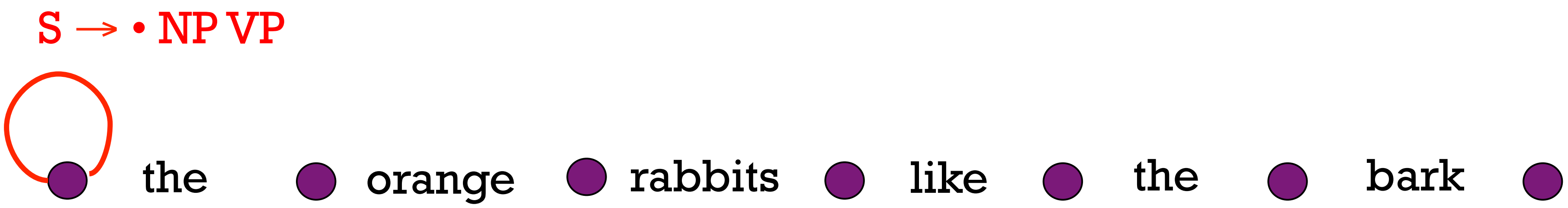


Chart Parsing

- Top-Down Initialisation:
  - add an **active (incomplete) edge for the S category**



S → • NP VP

the orange rabbits like the bark

Chart Parsing

- Top-Down Expand Rule:
  - expand by adding an **active (incomplete) edge for all left corners (i.e. top-down predict)**

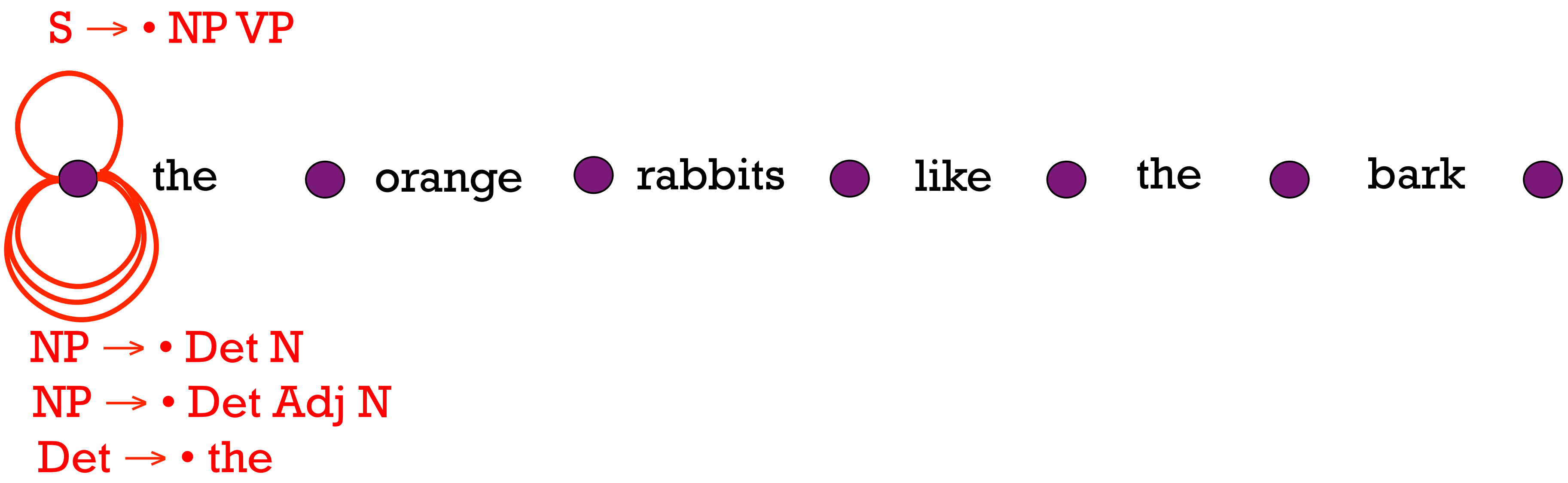


Chart Parsing

- Top-Down Match Rule:
  - Allows the grammar predictions to be matched against the input by adding a complete edge for the lexical items

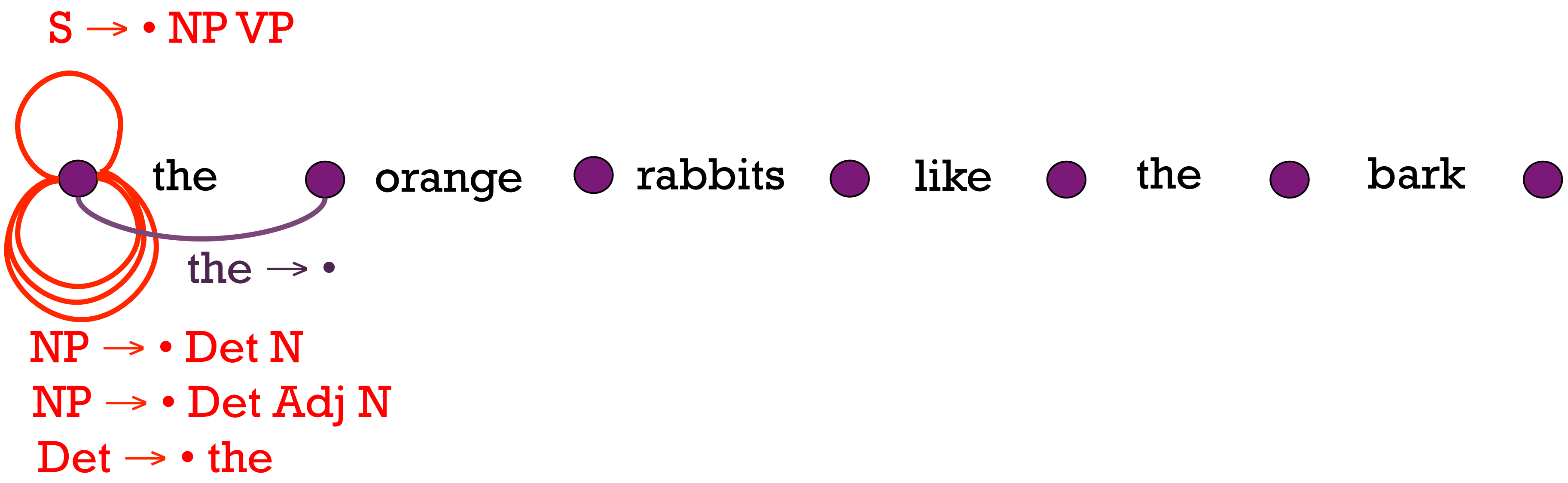
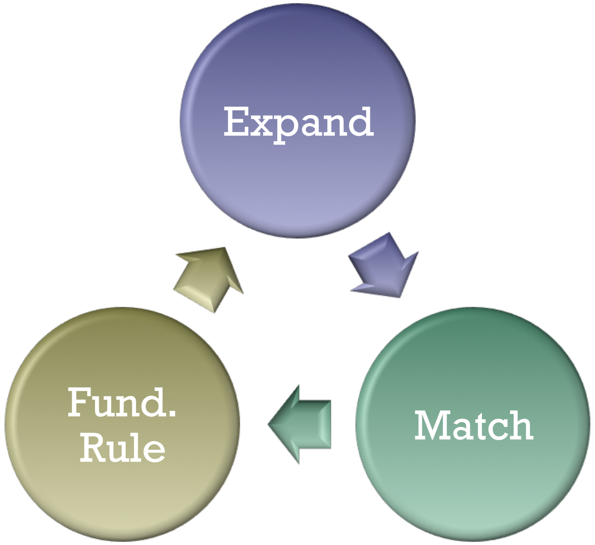
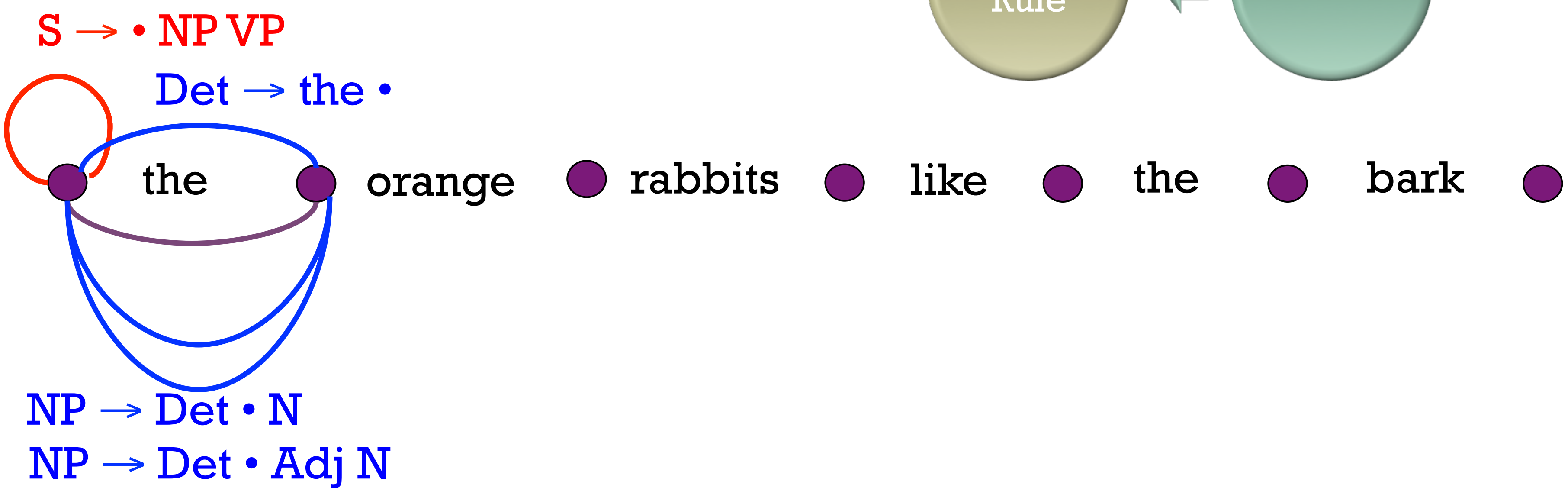


Chart Parsing

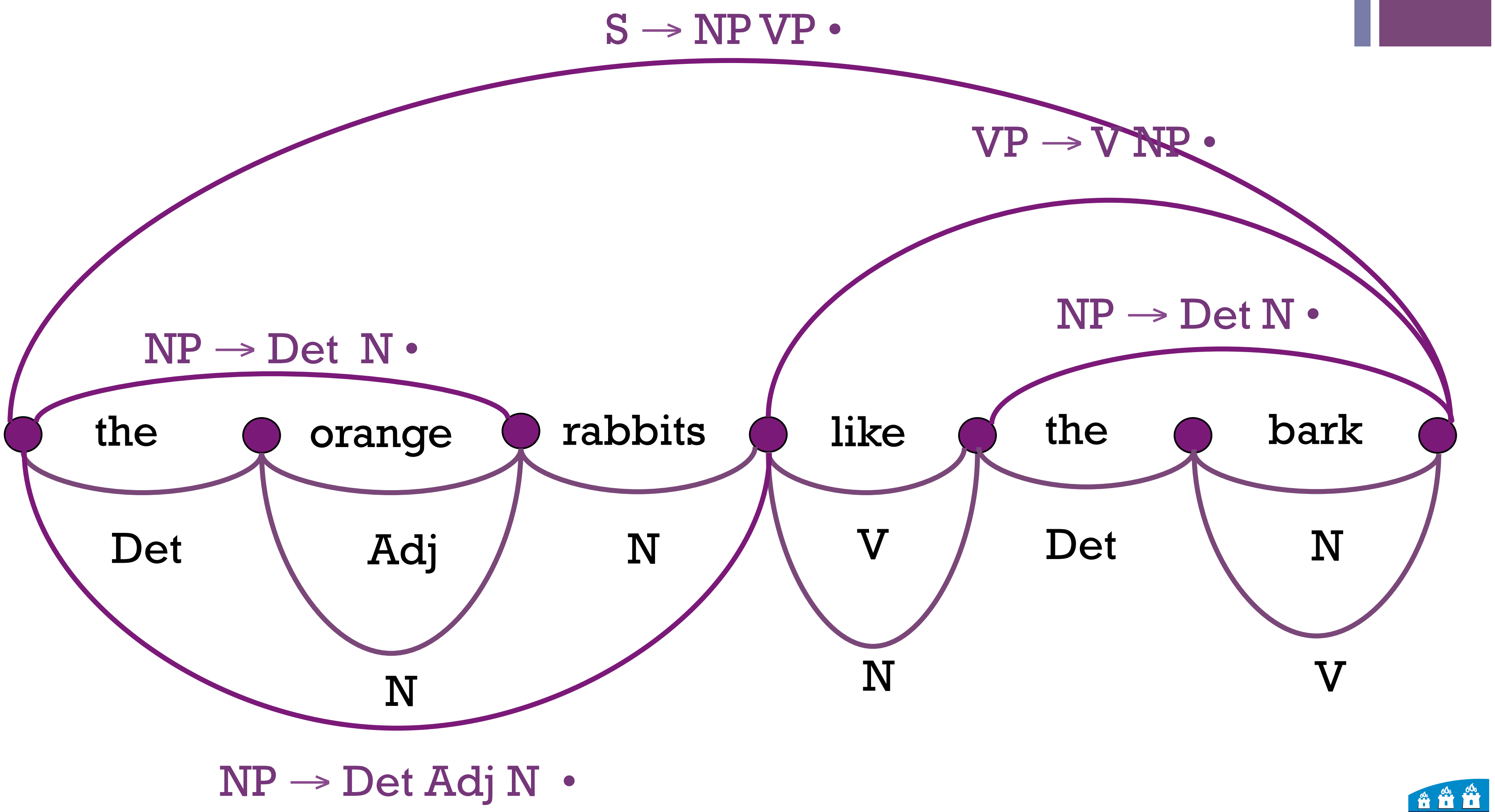
- Apply Fundamental Rule





+ Chart Parsing

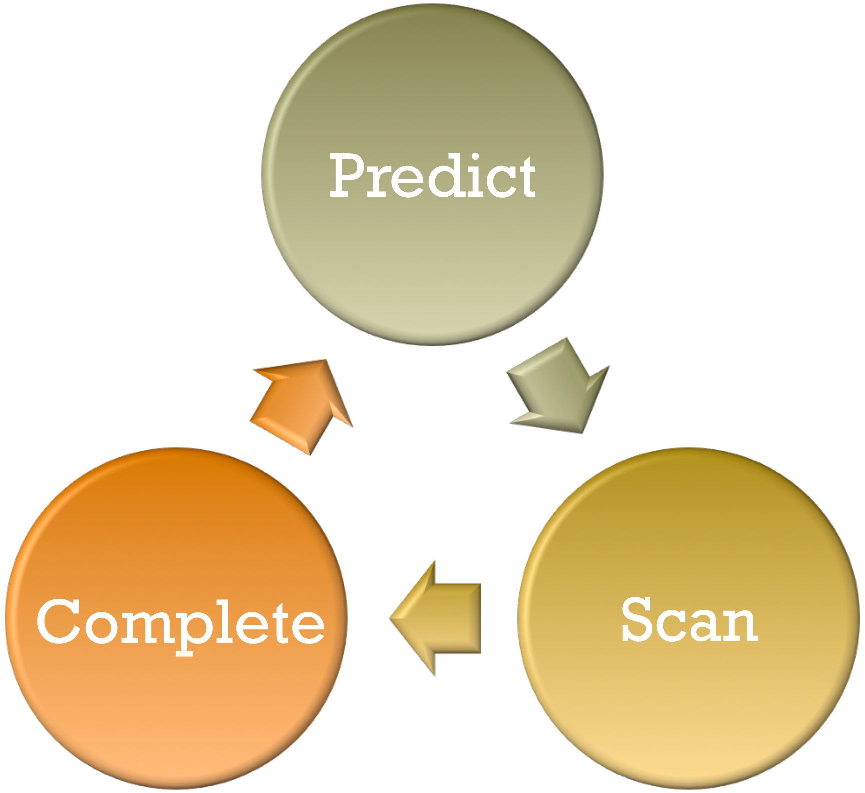
- Finished



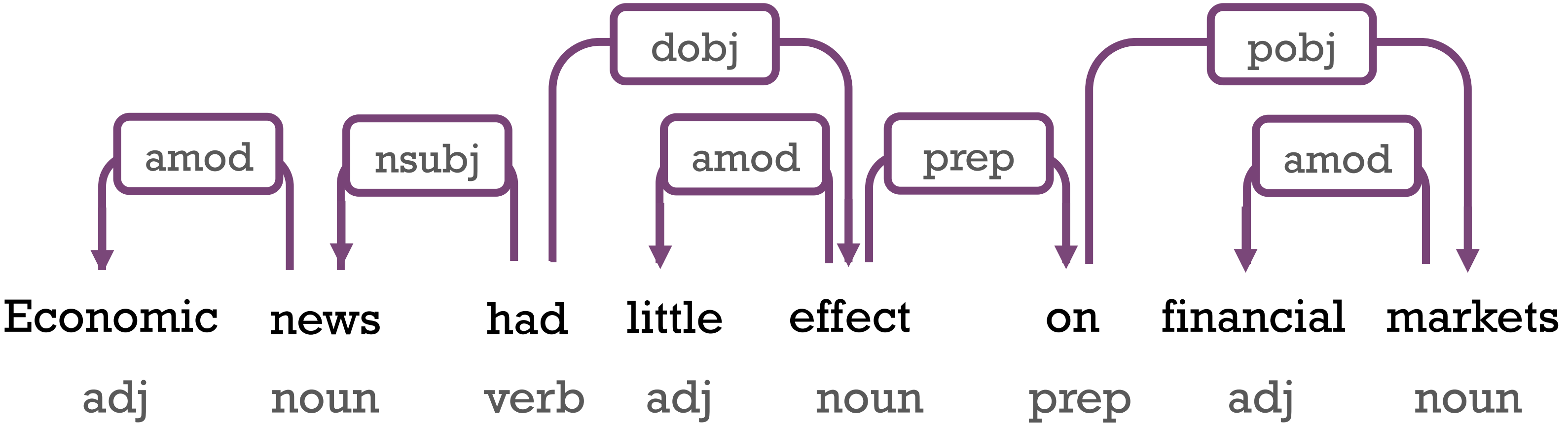
+

Chart Parsing

- Earley Algorithm
  - Predict
  - Scan
  - Complete

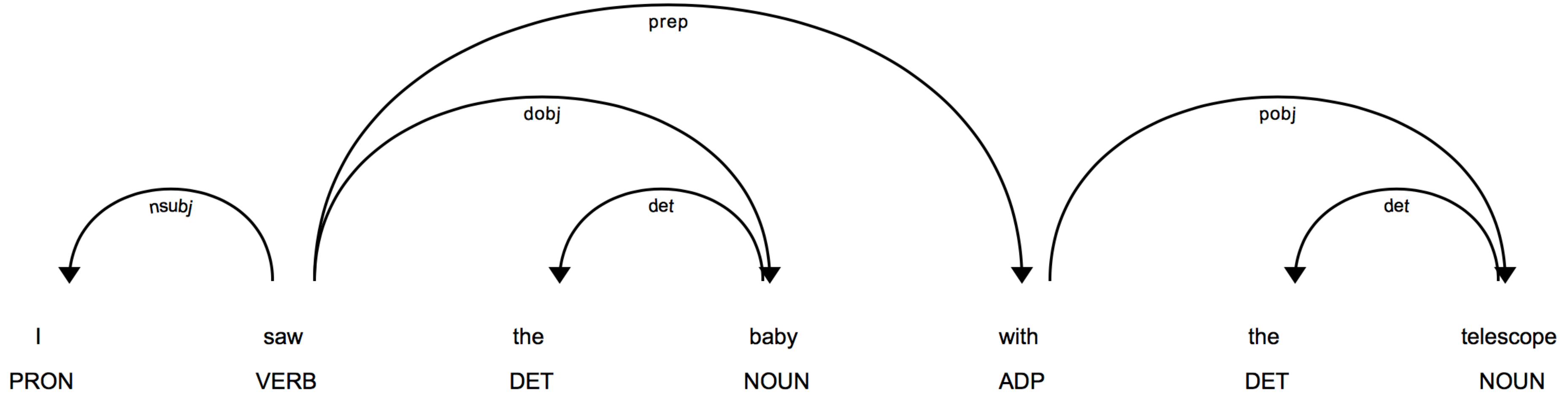


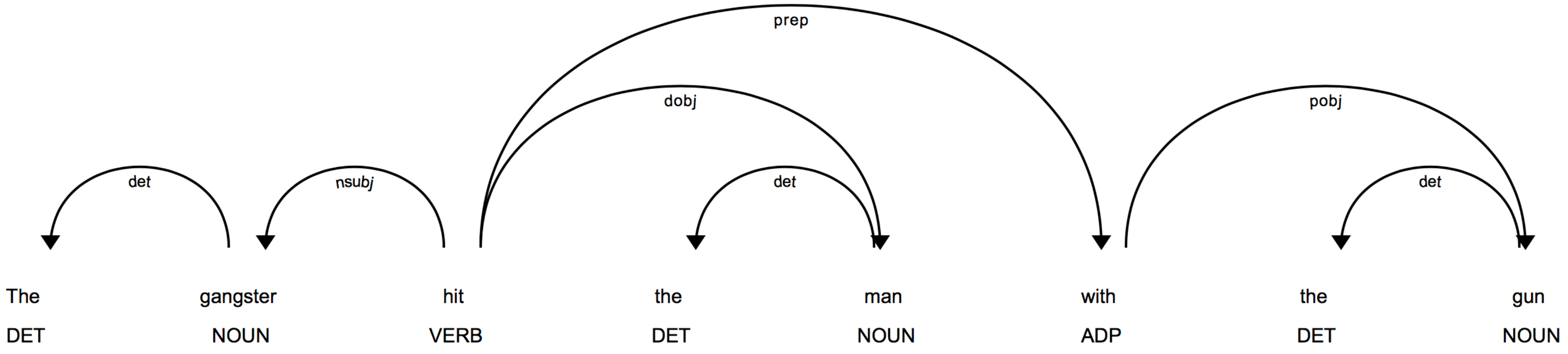
Dependency Parsing



From: http://stp.lingfil.uu.se/~nivre/docs/05133.pdf

+ Dependency Parsing


prep
dobj
pobj
nsubj
det
det
I
PRON
saw
VERB
the
DET
baby
NOUN
with
ADP
the
DET
telescope
NOUN


prep
dobj
pobj
det
nsubj
det
det
The
DET
gangster
NOUN
hit
VERB
the
DET
man
NOUN
with
ADP
the
DET
gun
NOUN