Parsing and syntactic analysis in NLP

Linguistic Resources for Natural Language Processing
LM Language Technologies and Digital Humanities
2024-25

Cristina Bosco

Ambiguity in syntax

Regardless of the format used for the annotation of syntax, in this level of linguistic knowledge, ambiguity is a very hard problem to be dealt with.

The notion of ambiguity referred to syntax is different from that referred to the part-of-speech tagging task: syntactic ambiguity is a form of **structural ambiguity**.

Structural ambiguity occurs when the grammar can assign more than one parse / structure to a sentence.

Ambiguity in syntax

Structural ambiguity comes in many forms but mostly as:

- attachment ambiguity

- coordination ambiguity

These two forms of structural ambiguities also combine in complex ways in real sentences.

Ambiguity in syntax

A sentence has an **attachment ambiguity** if a particular constituent can be attached to the parse tree structure at more than one place.

Various kinds of phrases are subject to the attachment ambiguity:

For example in:

We saw the Eiffel Tower flying to Paris.

the gerundive-**VP** *flying to Paris* can be:

- part of a gerundive sentence whose subject is *the Eiffel Tower*
- an adjunct modifying the VP headed by *saw*.

Ambiguity in syntax

For example in:

We saw the man with a telescope.

the **PP** *with a telescope* can be:

- part of a NP headed by *man*
- an adjunct modifying the VP headed by *saw*.

Ambiguity in syntax

The longer a sentence is, the more possibilities for attachment ambiguity arise.

For example, for the following sentences with increasing length, the number of syntactic structures that can be generated by a parser also increases:

7 words: *List the sales of products in 1973* > 3 structures
8 words: *List the sales of products produced in 1973* > 10 structures
13 words: *List the sales of products produced in 1973 with the products in 1972* > 28 structures
14 words: *List the sales of products produced in 1973 with the products produced in 1972* > 455 structures

Ambiguity in syntax

Coordination ambiguity depends on the presence of a conjunction that conjoin different phrases and can have different scope.

For example, *old men and women* can be bracketed as:

- [*old* [*men and women*]] referring *old* to *men* and *women* both, meaning that men and women are all old

- [*old men*] *and* [*women*] referring *old* to *men* only, meaning that men are old but women aren't .

Dependency structure

The syntactic structure of the sentence cannot be represented only through the constituents, there are many other completely different ways to highlight its complexity.

In particular, dependency grammars have imposed themselves in the NLP over the past 20 years as the alternative formalism to that of constituents that had dominated the NLP since the 1950s thanks to the work of Noham Chomsky.

Dependency structure

The first known dependency grammar consists of about 4,000 rules for Sanskrit provided by the Indian grammar Pāṇini (प ण न) (lived after the 7th and before the 3rd century b.C.).

The idea of formalizing the rules governing human language influenced not only the work of **Chomsky**, but also that of the Swiss linguist **Ferdinand de Saussure** (1857-1913).

But who formulated modern dependency grammar was the French linguist **Lucien Tesnière** (1893-1954).

Since then many scholars have worked on this type of grammar, including the Russian **Igor Mel'cuk** (Meaning Text Theory), the English **Richard Hudson** (Word Grammar) and **Peter Sgall** from the University of Prague.

Dependency structure

While the phrase structure is based on the idea of the hierarchical order of constituents and the possibility of organizing words into groups, the dependency structure is based on the **relationships that bind individual tokens** to each other within the sentence.

This means in particular that the notion of units and sub-units of a sentence does not exist in the dependency structure: in a dependency structure, **there is nothing corresponding to the hierarchical grouping words** typical of constituency structure.

Dependency structure

In a dependency structure, the tokens themselves are the smallest units to be considered (just as in the constituent structure we have phrases as well as words).

No non-terminal node is allowed in a dependency structure.

This means that the number of nodes in a dependency tree for a sentence S corresponds exactly to the number of tokens of S.

The dependency tree for S is therefore more compact than the constituent tree for the same sentence S, and this means that it is also computationally less expensive.

Dependency structure

The constituency structures and the dependency structures are however united by the fact of having the shape of trees.

So saying that a sentence is represented in the form of a syntactic tree is not enough to distinguish the two types of representation.

Other aspects differentiate them.

Dependency structure

Dependency structure

Dependency relationships are **oriented and asymmetric**, that is:

- they go in a well defined direction, and precisely from a word HEAD to a word DEPENDENT, and are therefore represented with arrows.

Example: in A —> B
the relationship goes from A (head) to B (employee)

Dependency structure

The words of the sentence inside a dependency tree are also called **nodes** while the dependency relations are also called **arcs** that bind the nodes together.

The relations are usually labeled to indicate their functions.

Example: in

A —*Rel-name*-> B

A and B are nodes and the arc from node A (head) to node B (dependent) is a relationship labeled "*Rel-name*".

Dependency structure

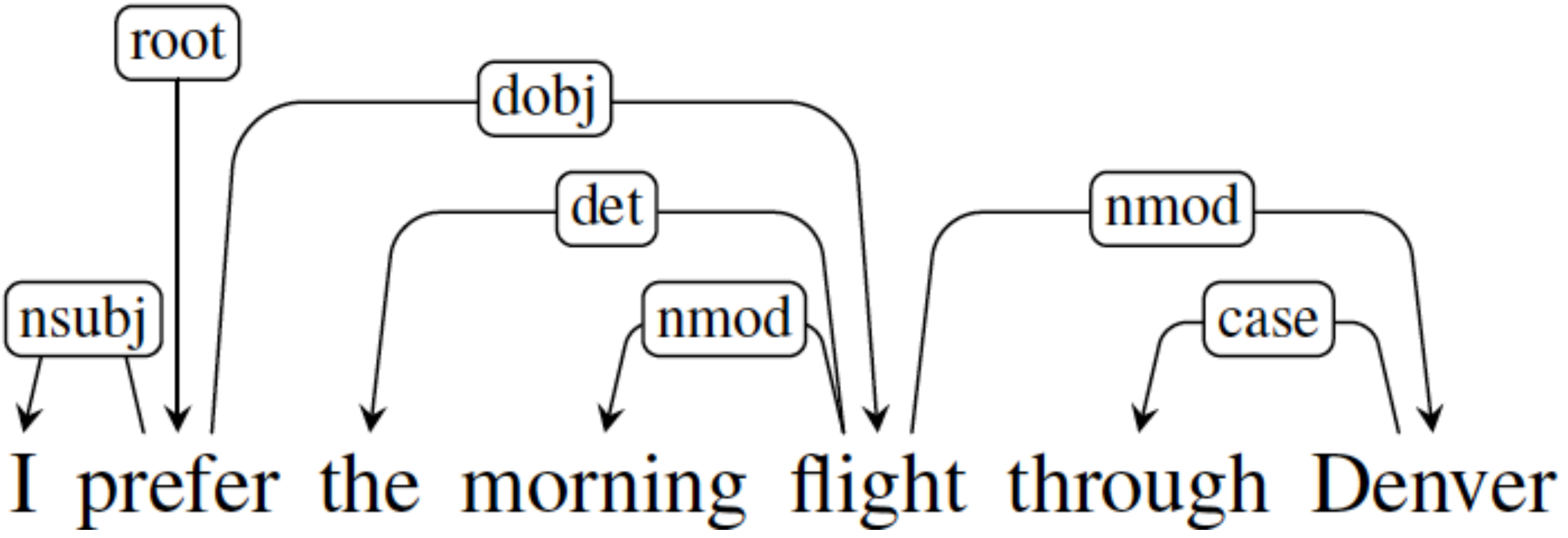
Labels applied to dependency relationships represent the type of dependency relationship. Dependency relationships can be of different types.

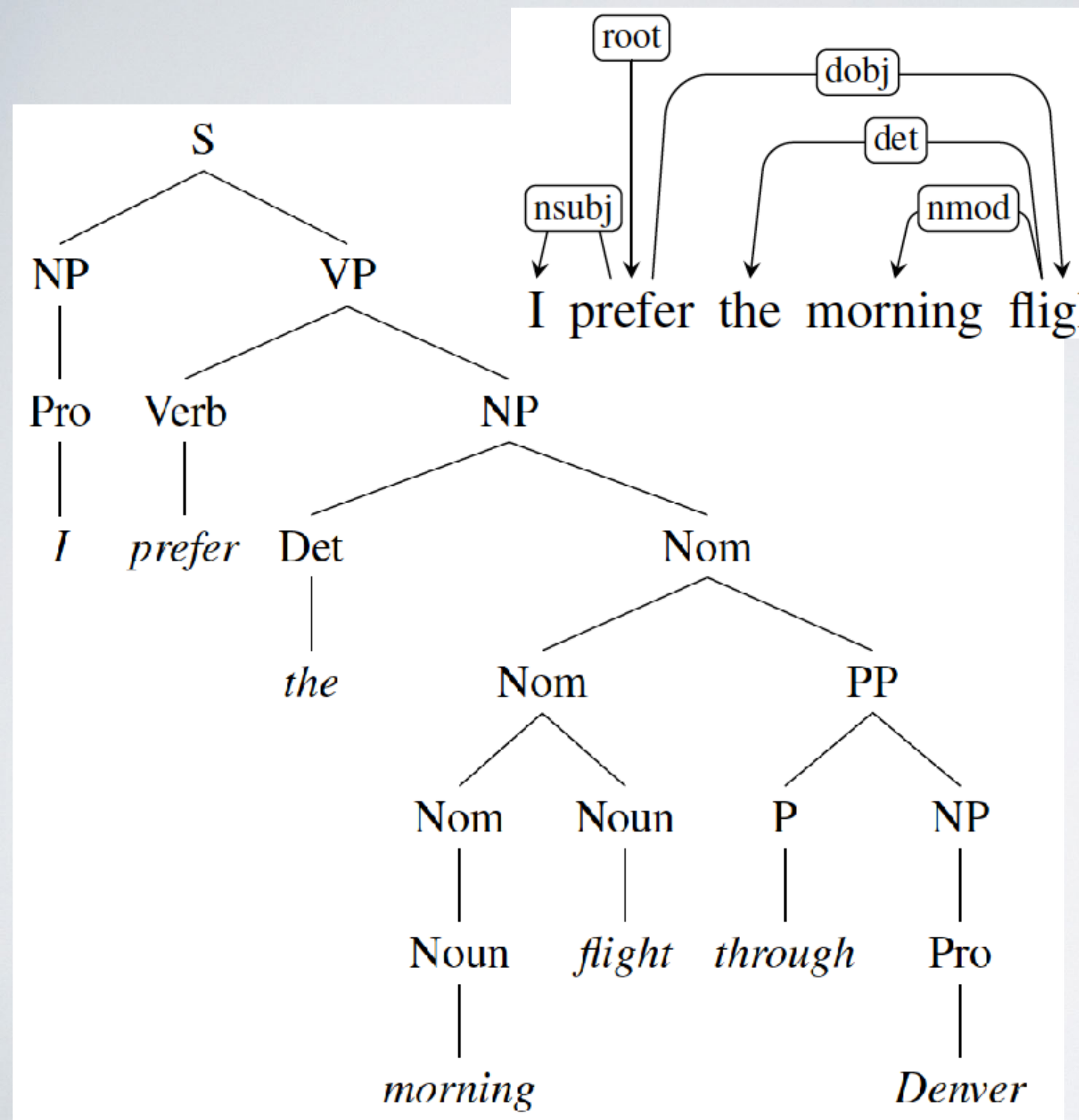
As a whole, the dependencies that bind the words of a sentence form a **tree** in which **all words must be connected**.

In every tree there is a **root**, that is a special node from which all the tree takes origin. The root is a node, so it's a word in the sentence.

Dependency structure

In a dependency tree there are no non-terminal nodes.
This means that each node corresponds to exactly one word of the sentence, and each node (except the root) depends on another node of the sentence.




S
NP
VP
Pro
Verb
NP
I
prefer
Det
Nom
the
Nom
PP
Nom
Noun
P
NP
Noun
flight
through
Pro
morning
Denver

root
dobj
det
nmod
nsubj
nmod
case
I prefer the morning flight through Denver

Dependency structure

Dependency relations encode **information that does not occur in constituency structures**.

They are independent of the order of the words they link and this makes dependency formalisms capable of representing languages with free order of constituents, which are problematic for CFGs.

For example:

the Adjective in Italian can precede or follow the Noun, in a CFG we are forced to insert a rule for each of the two cases, as

NP -> Article Adjective Noun

NP -> Article Noun Adjective

And this for every other element that can appear in different positions within the sentence.

Dependency structure

Every rule of a dependency grammar describes only the behaviour of two elements to be linked together and this makes irrelevant the order in which they occur in the sentence with respect to other elements.

For example:
For the Adjective in Italian the dependency structure can be built using the following independent rules:

- Adjective depends on Noun
- Article depends on Noun

which can be applied regardless of the relative order of Article, Adjective and Noun.

Dependency relations

Dependency **relations can be labeled** in various ways that relate to a set of relationships known in the literature that express functions that words perform in the sentence:

subject
direct object
indirect object
modifier
coordination
…

Dependency relations

In the various dependency formats the relations can have different names and can be more or less specialized.

For example in Universal Dependencies:

subject

> NSUBJ if the subject is headed by a noun

The cat *is sleeping in the garden*

> CCOMP if the subject is headed by a verb

Sleeping *in the garden in the sun is very pleasant*

Dependency relations

Direct object

> DOBJ if the object is headed by a noun

*Mary gave George **a book***

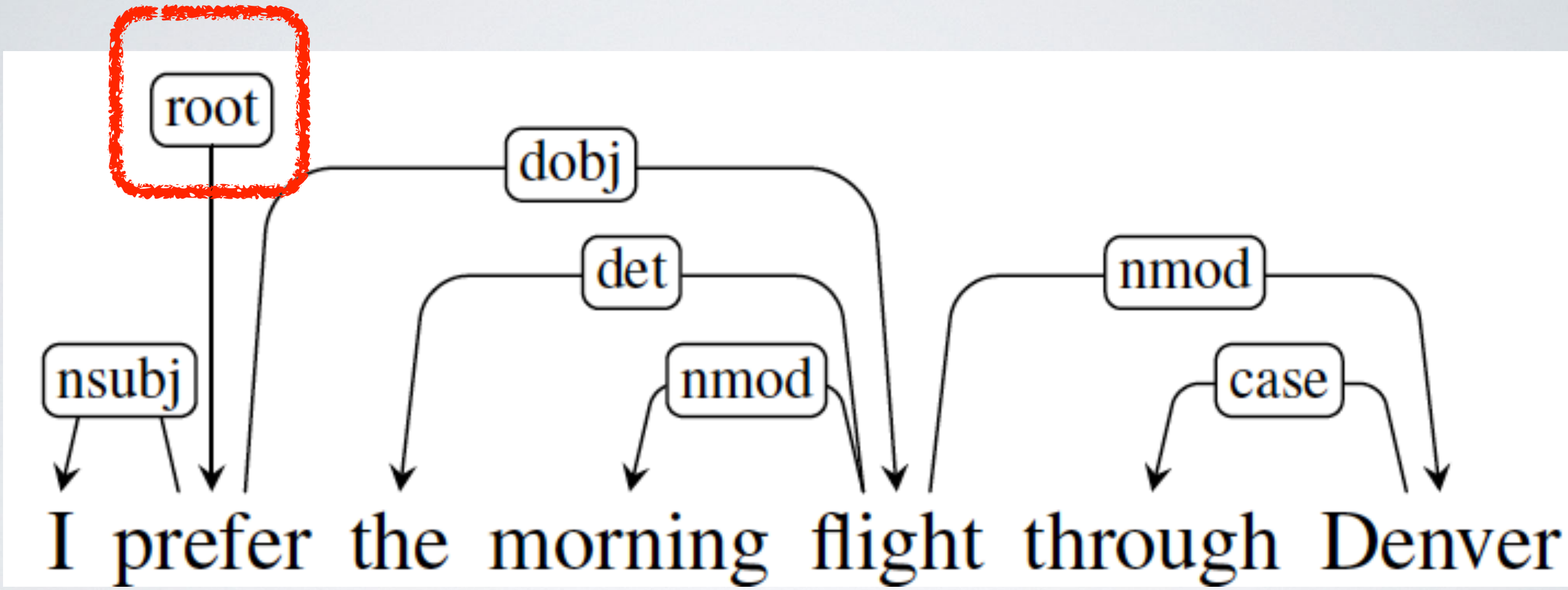
> CCOMP if the object is headed by a verb

*Mary said **to take** the book*

Dependency: properties

Formally a dependency structure is called a *dependency tree* and is defined as a **direct acyclic graph**, this entails compliance with the following **formal properties**:

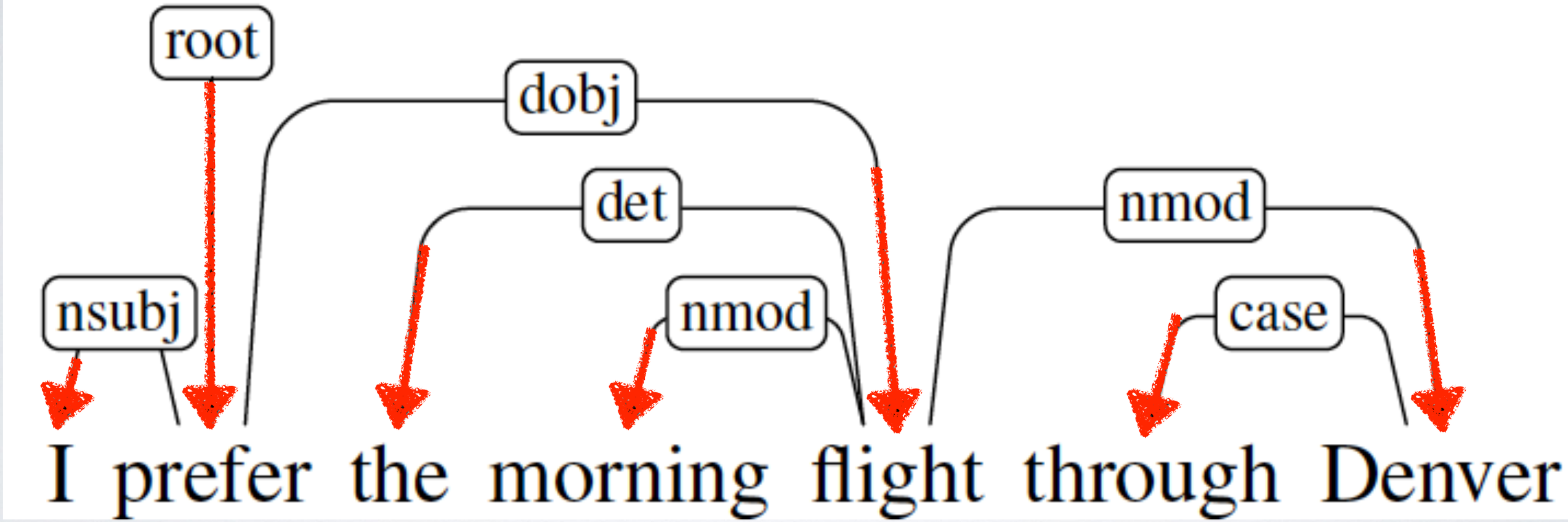
- in the tree there is only one node that has no input arcs (the **root** node)


root
dobj
det
nmod
nsubj
nmod
case
I prefer the morning flight through Denver

Dependency: properties

Formally a dependency structure is called a *dependency tree* and is defined as a **direct acyclic graph**, this entails compliance with the following **formal properties**:

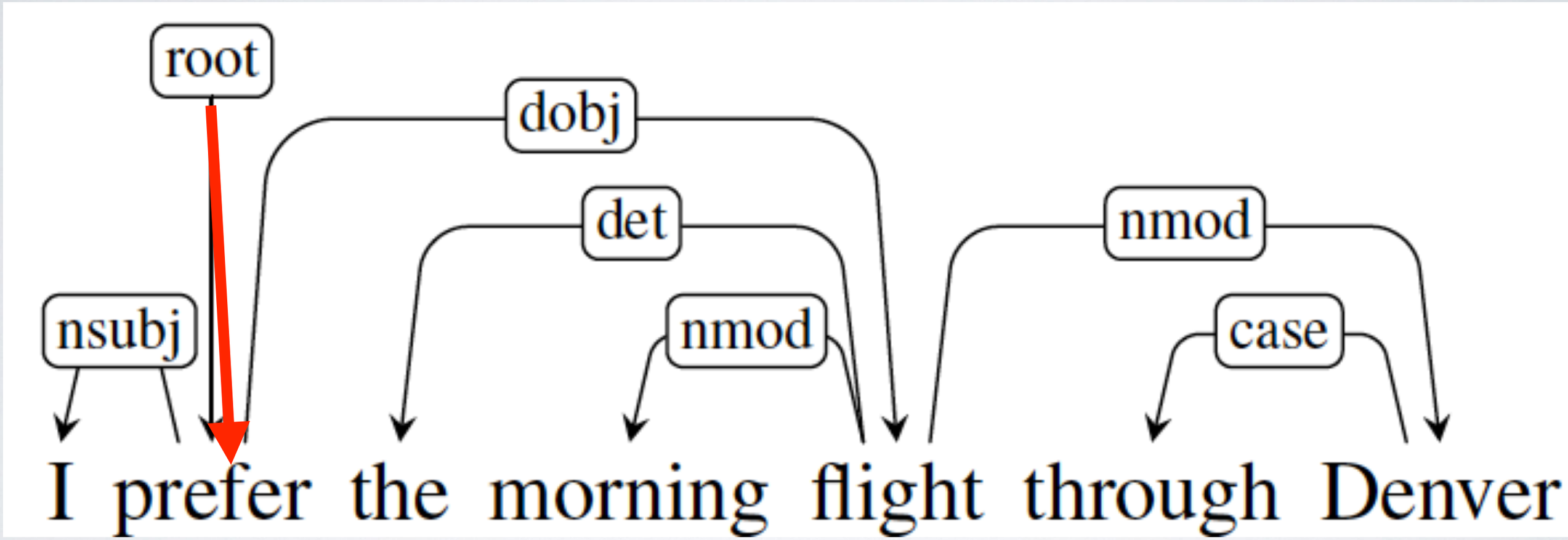
- in the tree there is only one node that has no input arcs (the **root** node)
- each node N of the tree has **one and only one arch in input**, representing its head node, from wich N depends

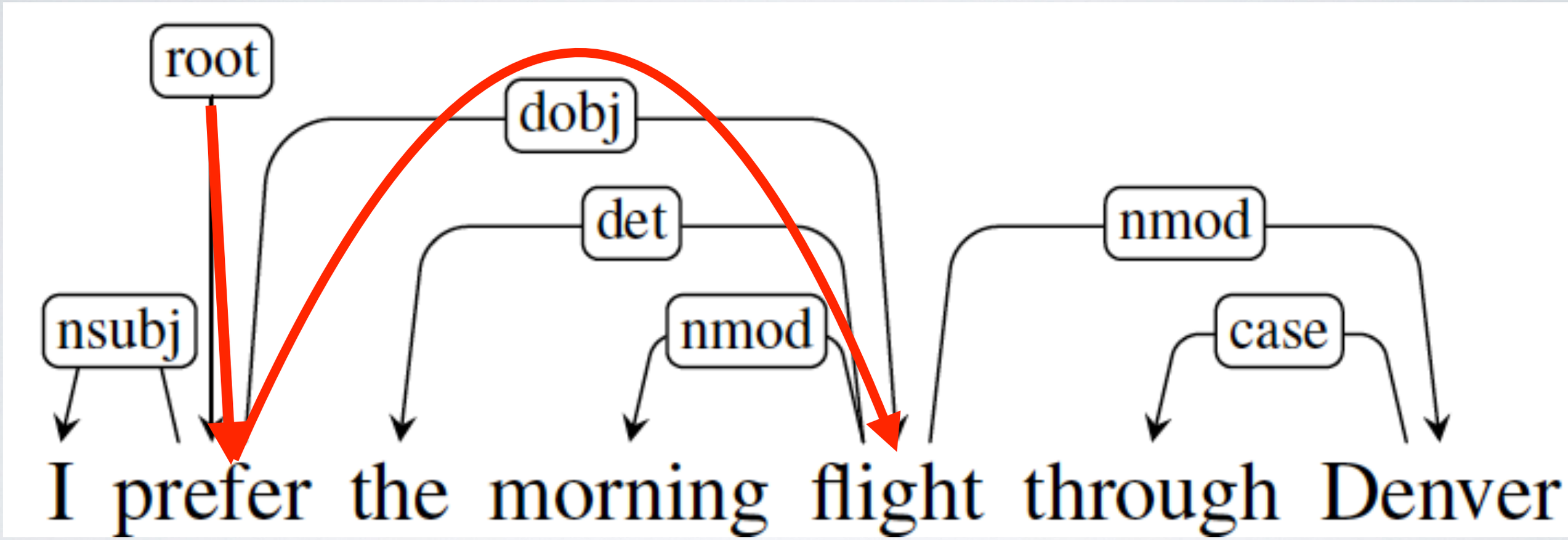

root
dobj
det
nmod
nsubj
nmod
case
I prefer the morning flight through Denver

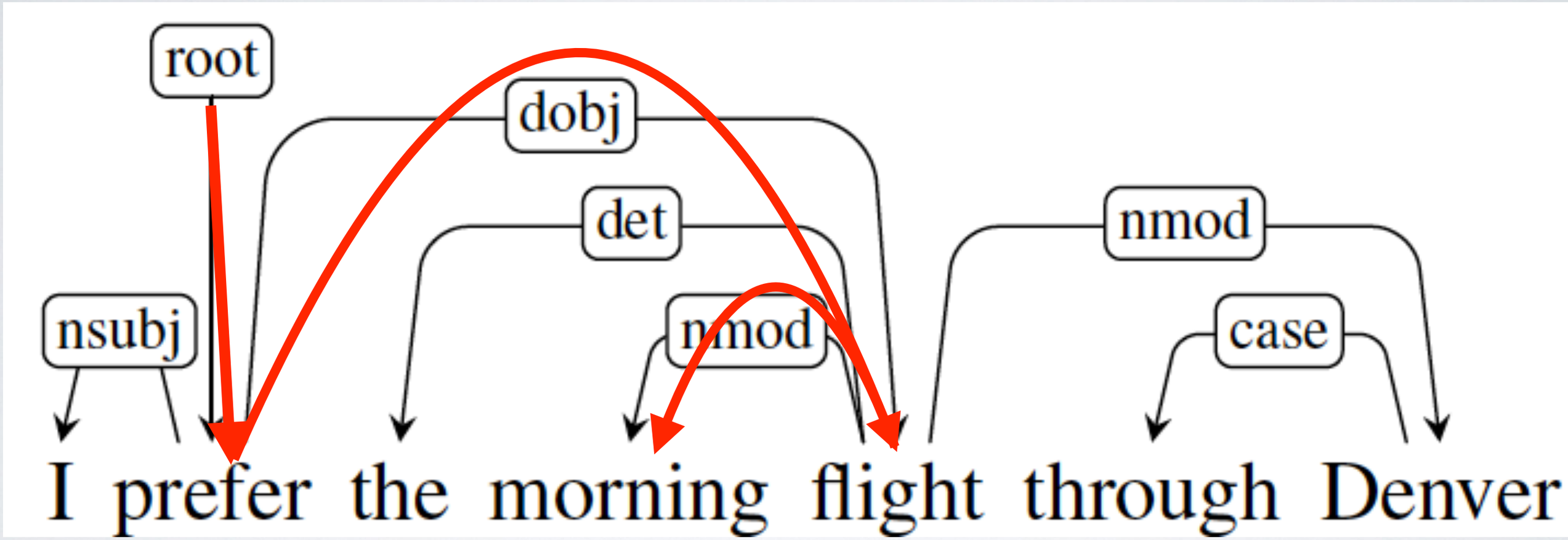
Dependency: properties

Formally a dependency structure is called a *dependency tree* and is defined as a **direct acyclic graph**, this entails compliance with the following **formal properties**:

- in the tree there is only one node that has no input arcs (the **root** node)
- each node N of the tree has **one and only one arch in input**, representing its head node, from wich N depends
- for each node of the tree there is **a single path** that goes from the root node to it


root
dobj
det
nmod
nsubj
nmod
case
I prefer the morning flight through Denver


root
dobj
det
nmod
nsubj
nmod
case
I prefer the morning flight through Denver

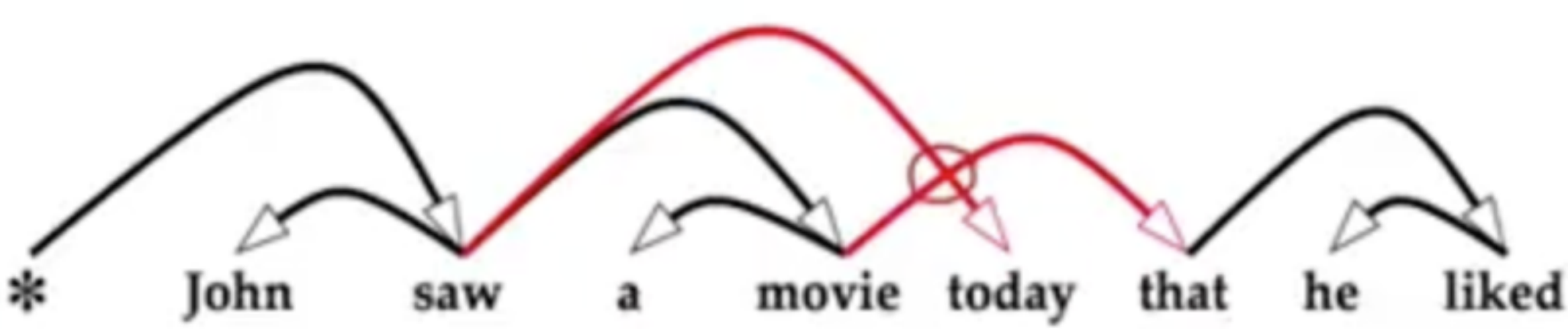

root
dobj
det
nmod
nsubj
nmod
case
I prefer the morning flight through Denver

Dependency: properties

Formally a dependency structure is called a *dependency tree* and is defined as a **direct acyclic graph**, this entails compliance with the following **formal properties**:

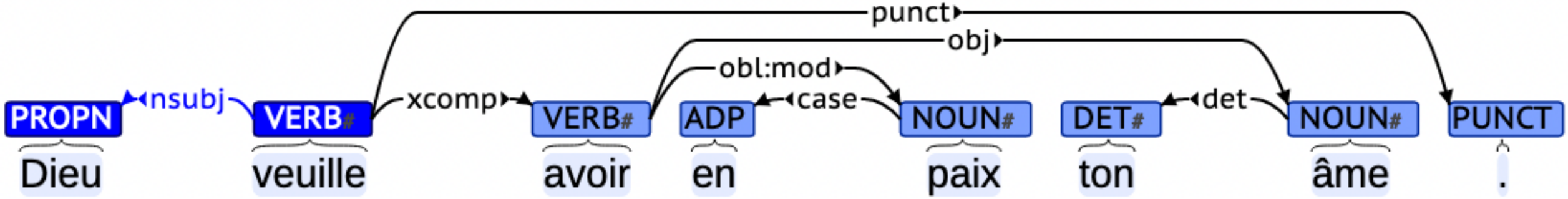
- in the tree there is only one node that has no input arcs (the **root** node)
- each node N of the tree has **one and only one arch in input**, representing its head node, from wich N depends
- for each node of the tree there is **a single path** that goes from the root node to it
- the **arcs** of a dependency tree **cannot cross**, i.e. the structure must be **projective**

Dependency: properties



Dependency structure

A dependency structure in Universal Dependencies



The tree for the French sentence:
Dieu veuille avoid en pain ton ame.
(God bless your soul. [God would have in peace your soul.])

Dependency structure

A dependency structure in Universal Dependencies **CONLLU** format

1	Il	il	DET	2	det
2	governo	governo	NOUN	6	nsubj
3	Monti	Monti	PROPN	2	nmod
4	sta	stare	AUX	6	aux
5	solo	solo	ADV	6	advmod
6	massacrando	massacrare	VERB	0	root
7	gli	il	DET	8	det
8	italiani	italiano	NOUN	6	obj
9	.	.	PUNCT	6	punct

RELATION

The tree for the sentence:
Il governo Monti sta solo massacrando gli italiani.
(The Monti's government is only slaughtering the Italians.)

Dependency structure

A dependency structure in Universal Dependencies **CONLLU** format

				HEAD	RELATION
1	Il	il	DET	2	det
2	governo	governo	NOUN	6	nsubj
3	Monti	Monti	PROPN	2	nmod
4	sta	stare	AUX	6	aux
5	solo	solo	ADV	6	advmod
6	massacrando	massacrare	VERB	0	root
7	gli	il	DET	8	det
8	italiani	italiano	NOUN	6	obj
9	.	.	PUNCT	6	punct

The tree for the sentence:
Il governo Monti sta solo massacrando gli italiani.
(The Monti's government is only slaughtering the Italians.)

Universal Dependencies

The nature of dependency grammar forces the linguist to choose in every dependency relation which element is to be regarded as head and which as dependent.

There is no *right* choice and different formalisms propose different solutions based on different criteria.

In UD, the choice is based on the criterion that the semantically more significant element of the token pair bound by a relation is chosen as the head of the respective relation.

Universal Dependencies

The application of this criterion has important consequences on the structures allowed in UD.

For example:

in “the dog” the determiner “the" is a dependent of the noun "dog" through the relation DET

in “the dog of John” the preposition “of" is a dependent of the proper noun “John" through the relation CASE

in “the man was running” the auxiliary verb “was" is a dependent of the main verb “running"through the relation AUX

Universal Dependencies

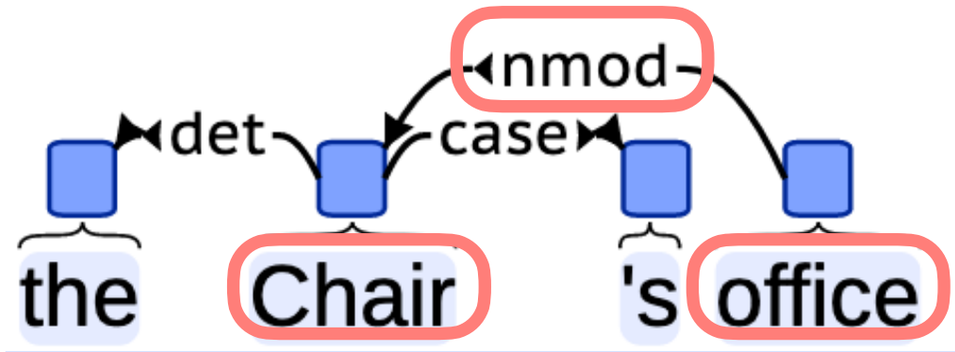
The criterion of considering the semantically more significant element as the head improves the consistency of the annotation and smoothes out the differences caused by different surface structures.

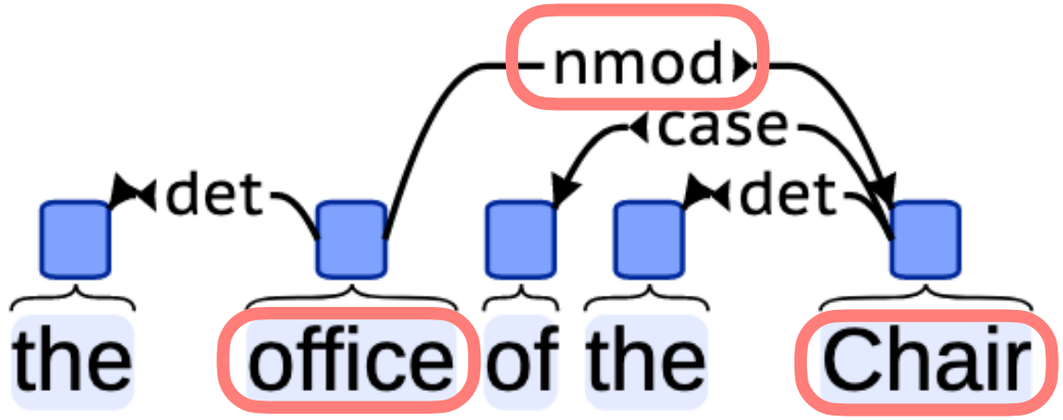
Considering that **different languages** often express the same content with **different surface forms**, the criterion applied in UD allows the syntactic structure to be described in a virtually **universal style**.

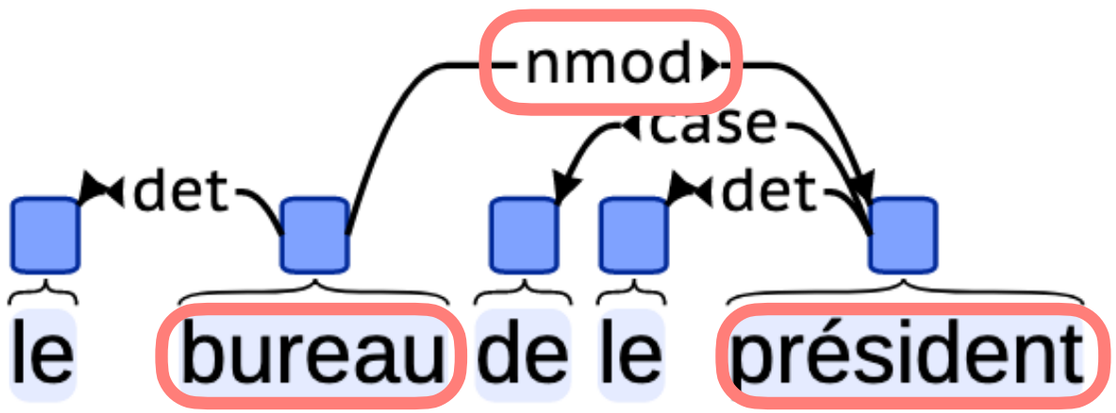
This annotation provides parallelism between different constructions across and within languages.

Universal Dependencies

In the **possessive alternation** the relations and the nodes are the same regardless of the surface form and word order

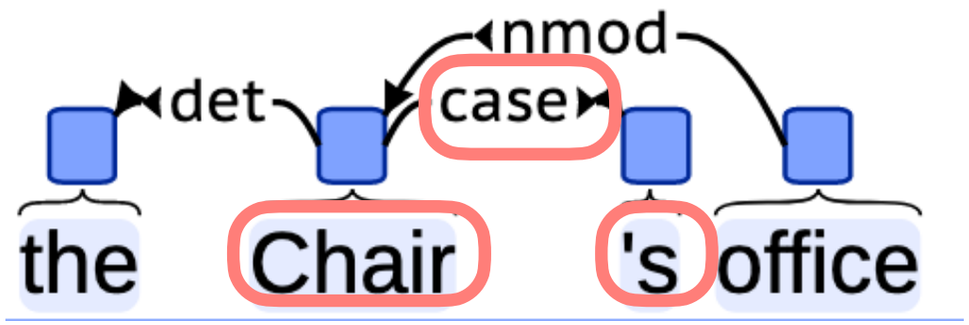


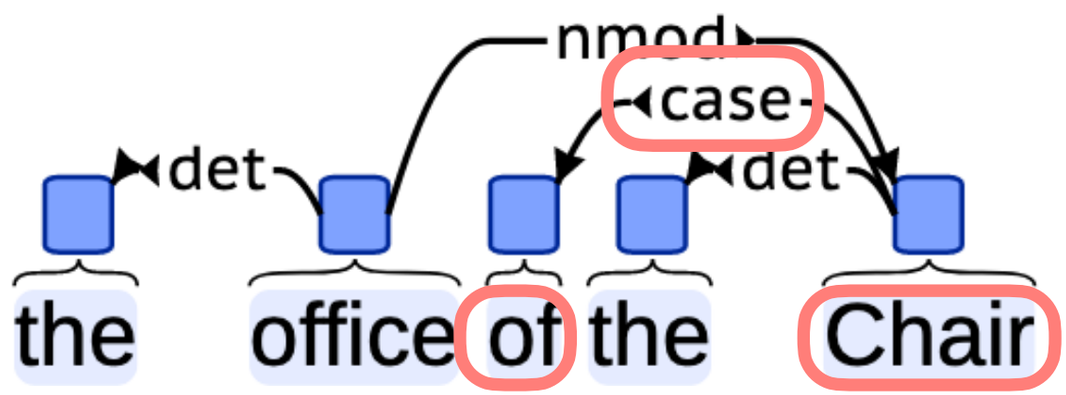


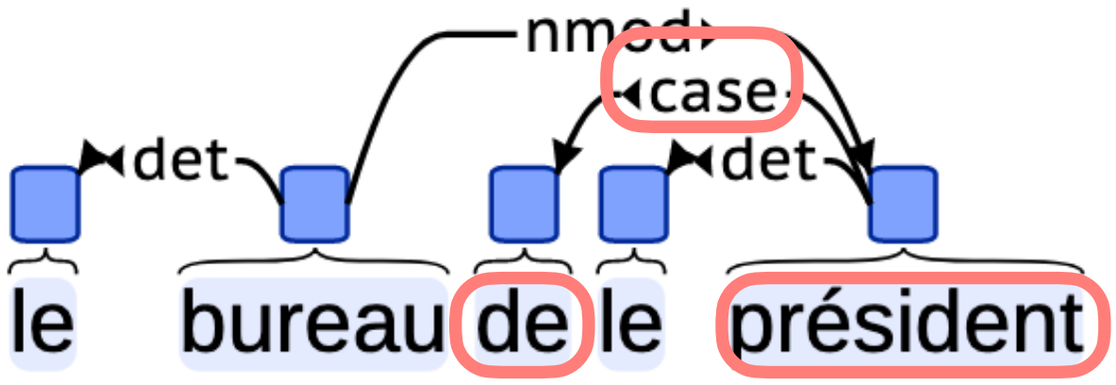


Universal Dependencies

In the **possessive alternation** the relations and the nodes are the same regardless of the surface form and word order

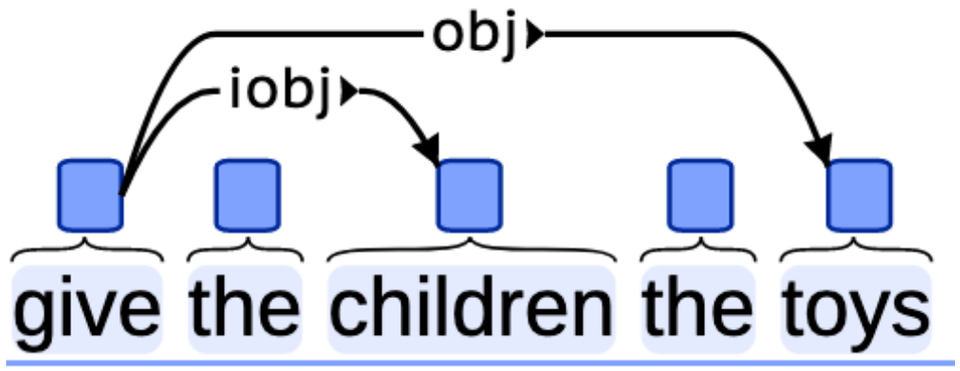


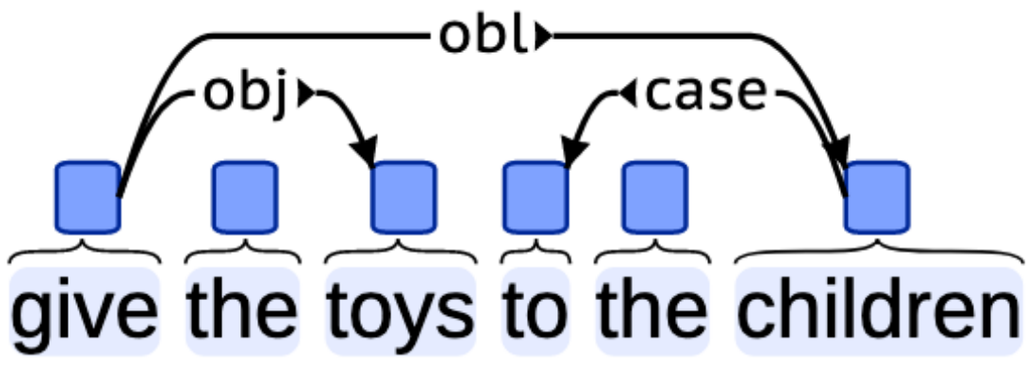


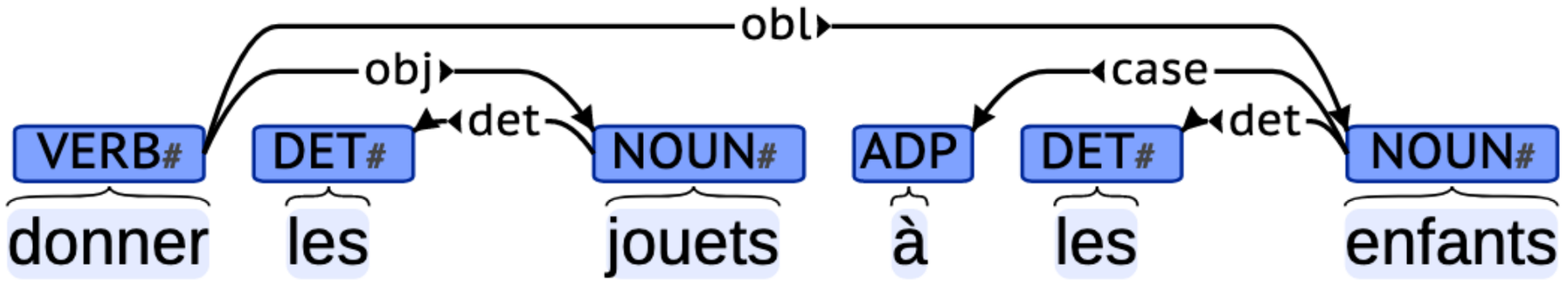


Universal Dependencies

In the **dative alternation** the relations and the nodes are the same regardless of the surface form in which the preposition is expressed or not

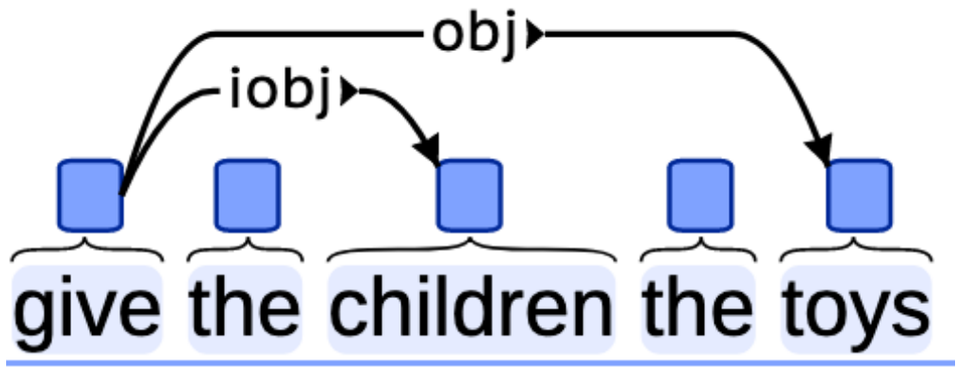


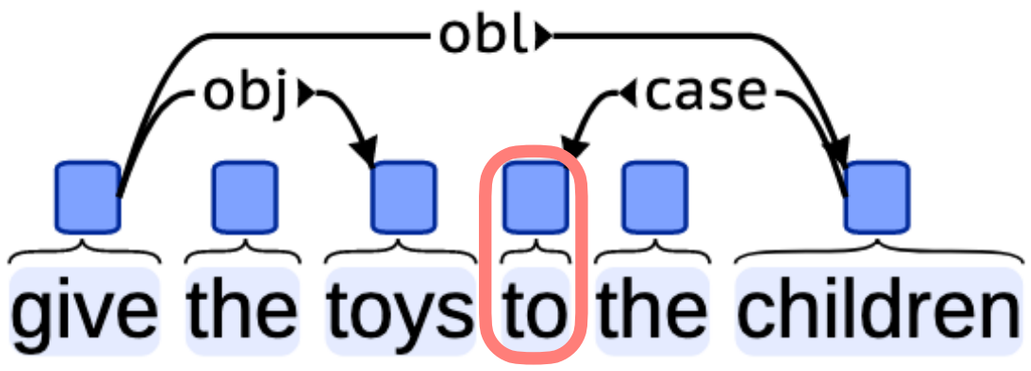


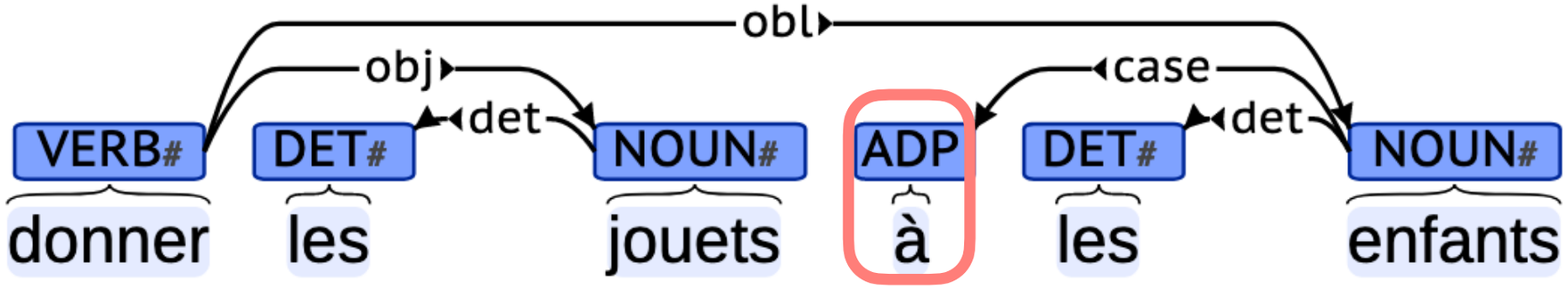


Universal Dependencies

In the **dative alternation** the relations and the nodes are the same regardless of the surface form in which the preposition is expressed or not



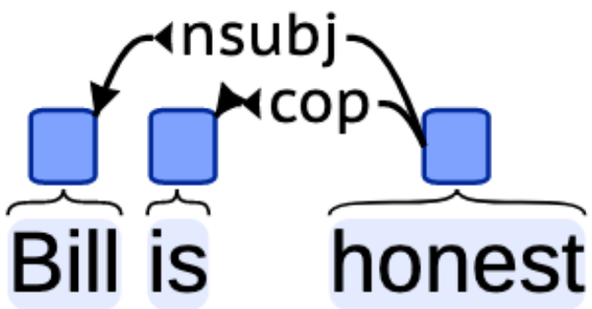


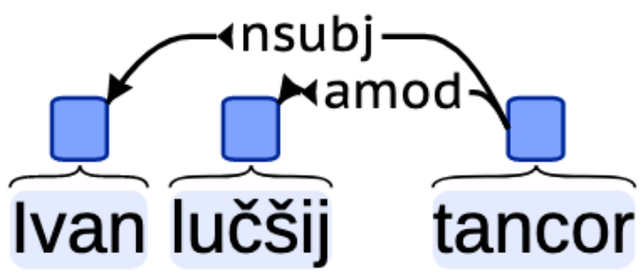


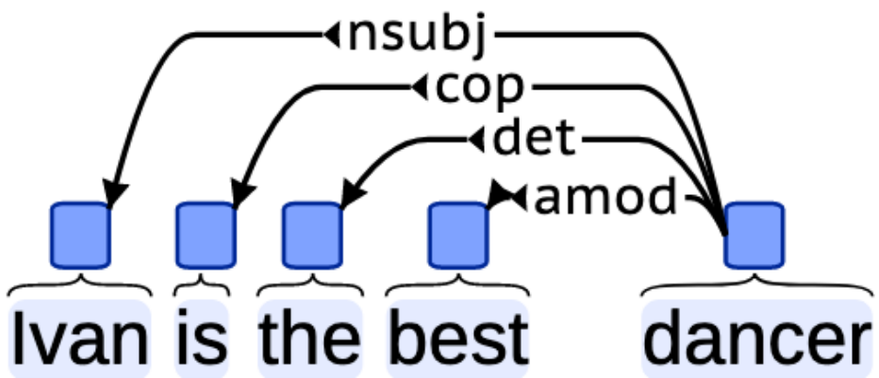
Universal Dependencies

In **copular sentences**, the (auxiliary) verb is not treated as the head of the sentence. The head is the non-verbal predicate.

This enables parallel constructions for languages that have an overt copula and those that do not (e.g. Russian)



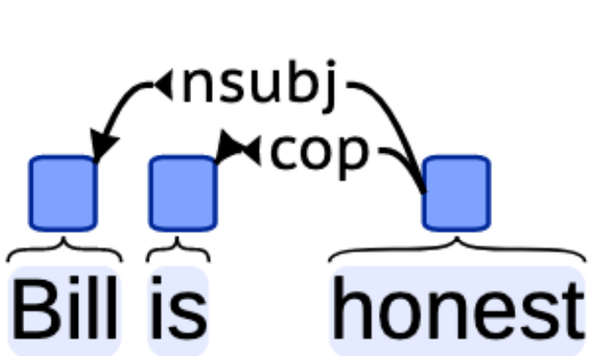


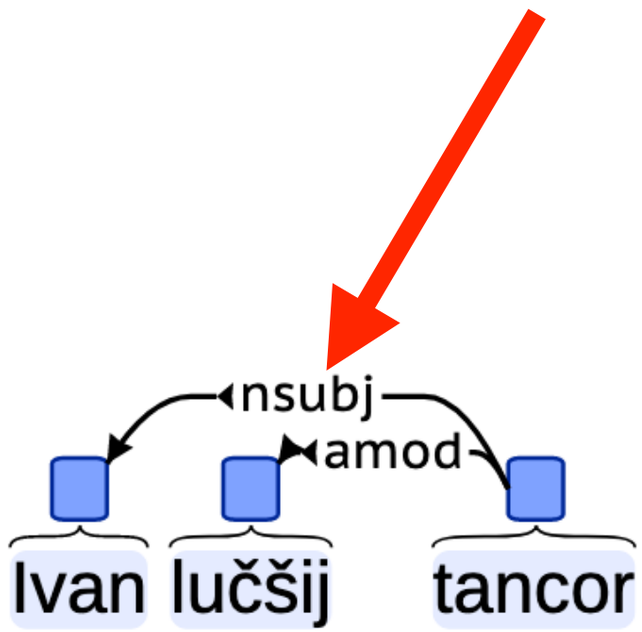


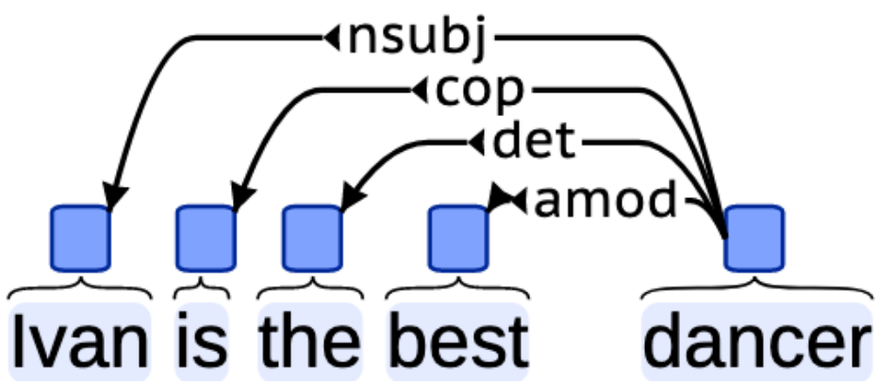
Universal Dependencies

In **copular sentences**, the (auxiliary) verb is not treated as the head of the sentence. The head is the non-verbal predicate.

This enables parallel constructions for languages that have an overt copula and those that do not (e.g. Russian)



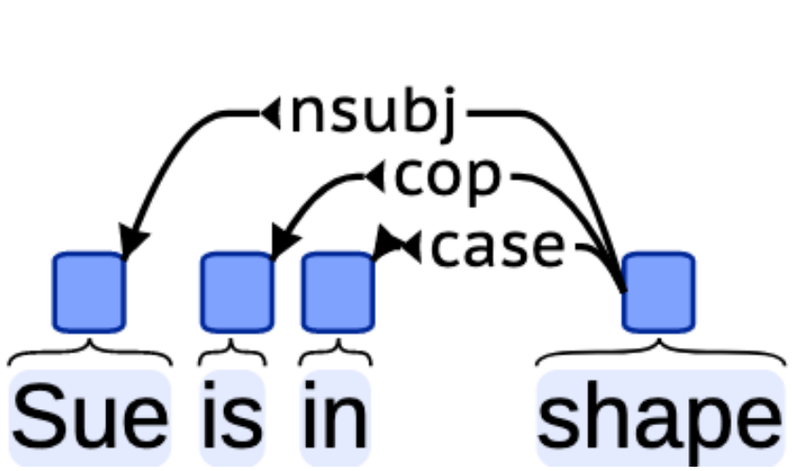


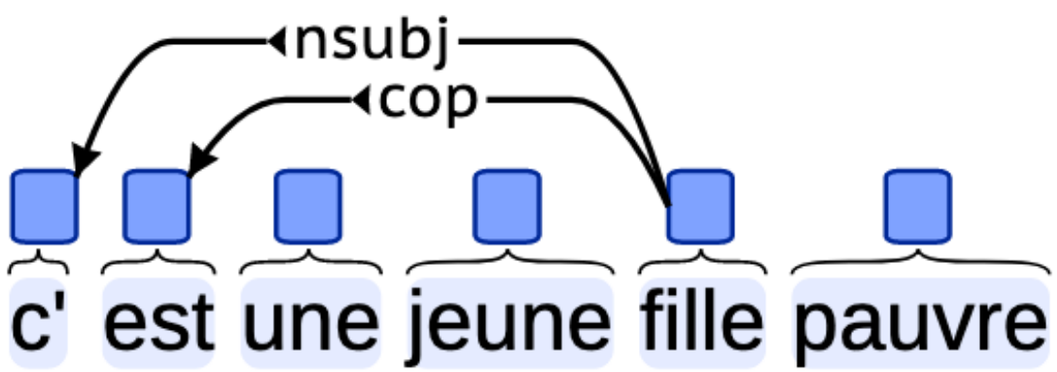


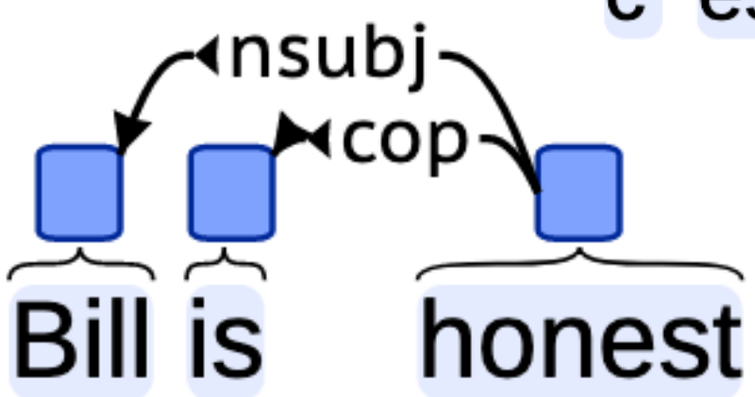
Universal Dependencies

In **copular sentences**, the (auxiliary) verb is not treated as the head of the sentence. The head is the non-verbal predicate. This enables parallel constructions for sentences in which the predicate is

a prepositional phrase or and adjective or a noun



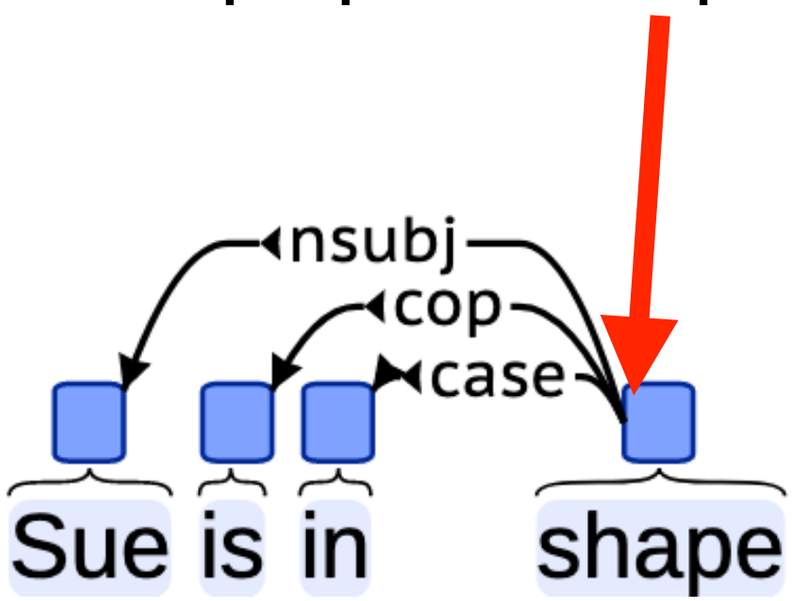


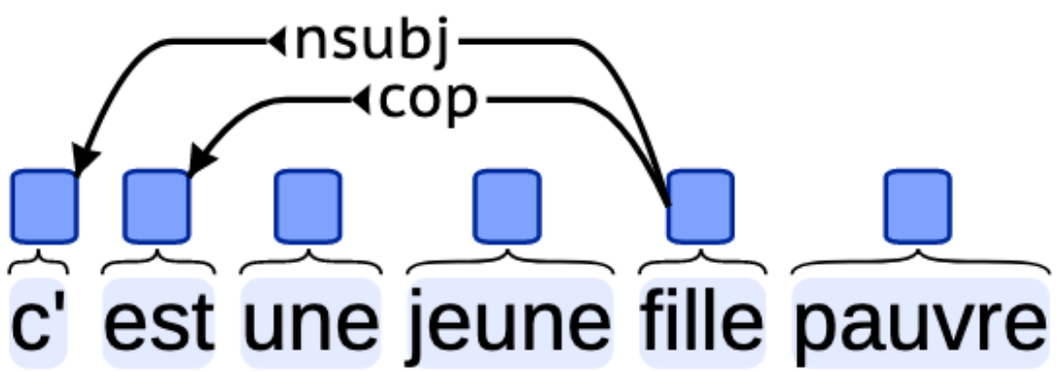


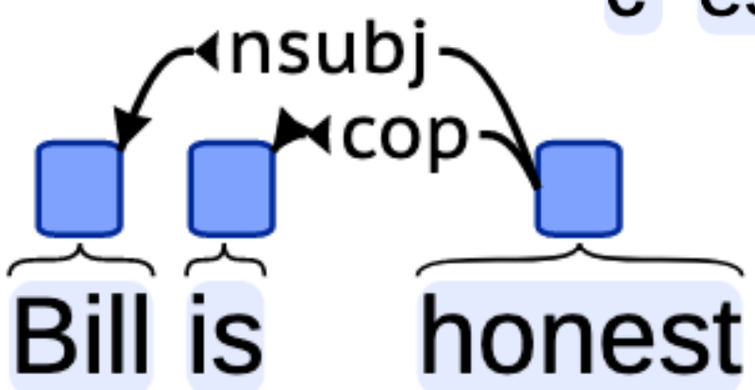
Universal Dependencies

In **copular sentences**, the (auxiliary) verb is not treated as the head of the sentence. The head is the non-verbal predicate. This enables parallel constructions for sentences in which the predicate is

a prepositional phrase or and adjective or a noun



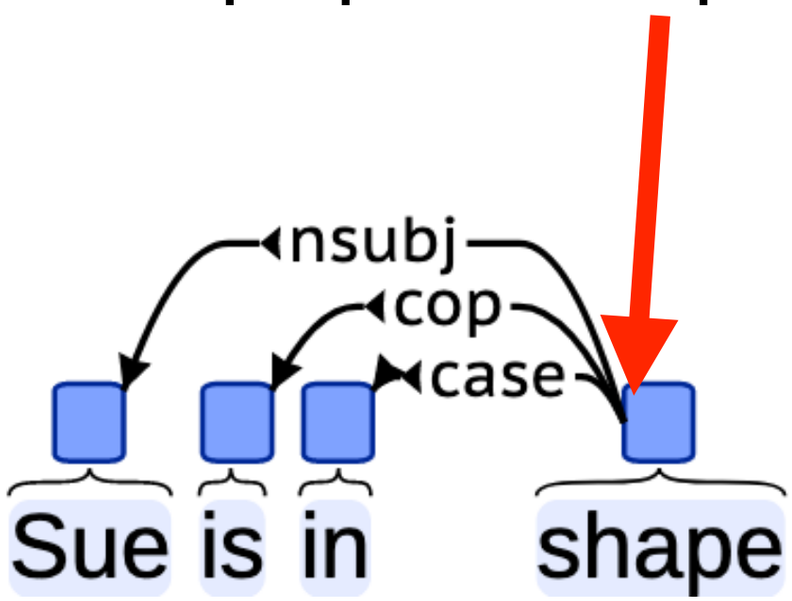


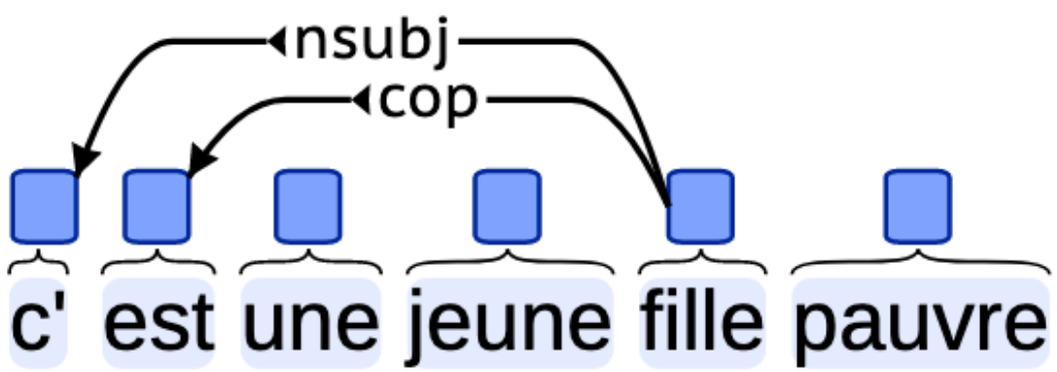


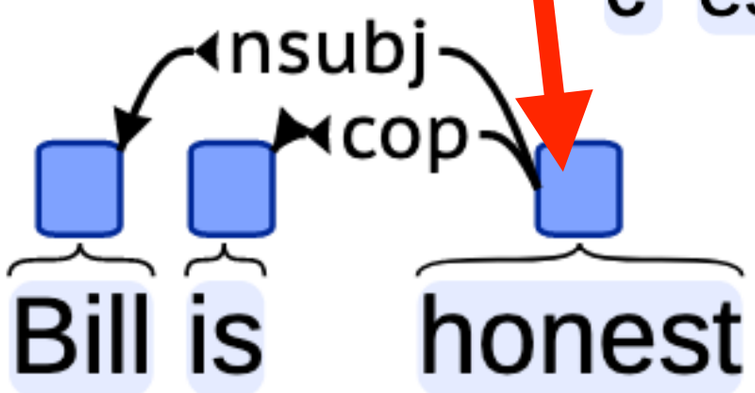
Universal Dependencies

In **copular sentences**, the (auxiliary) verb is not treated as the head of the sentence. The head is the non-verbal predicate. This enables parallel constructions for sentences in which the predicate is

a prepositional phrase or and adjective or a noun



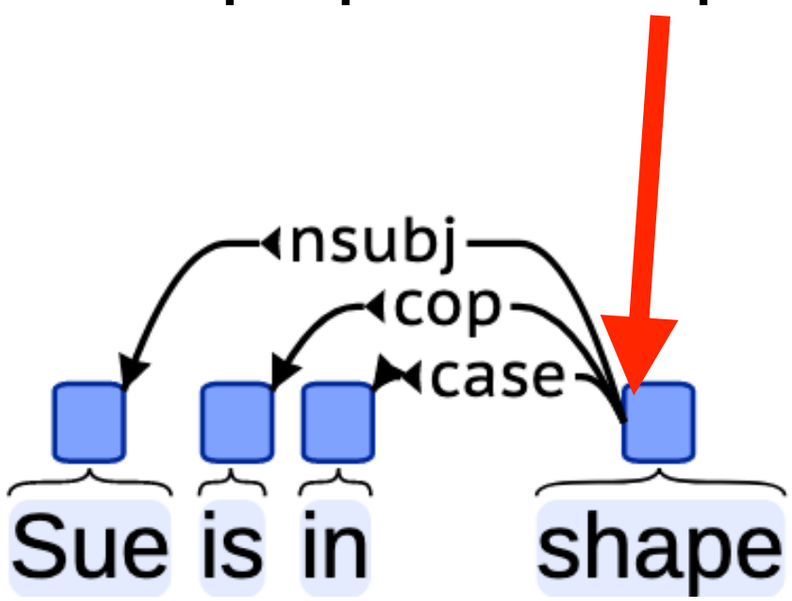


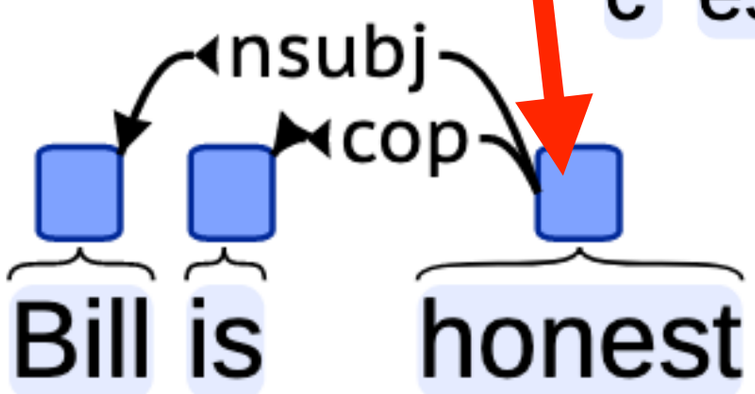


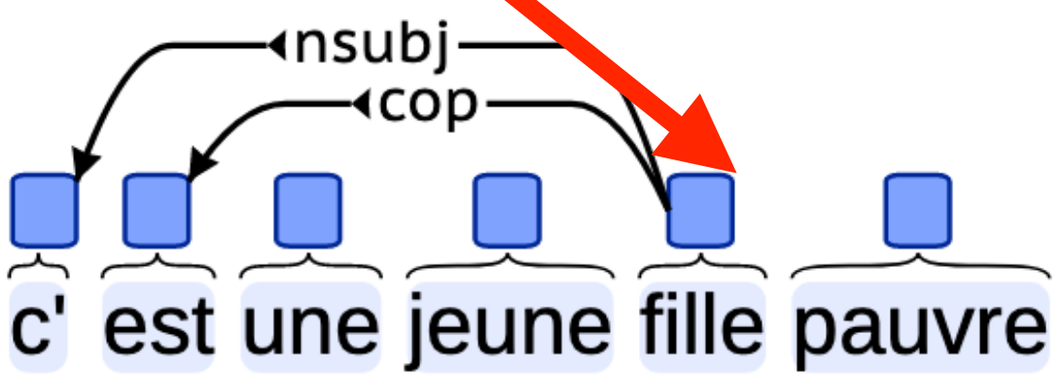
Universal Dependencies

In **copular sentences**, the (auxiliary) verb is not treated as the head of the sentence. The head is the non-verbal predicate. This enables parallel constructions for sentences in which the predicate is

a prepositional phrase or and adjective or a noun







Universal Dependencies

In **relative clauses**, the head is the main verb and the relativizer (the relative pronoun that, who, which, whose …) its dependent. This enables parallel constructions for sentences in which the relativizer is lexically realised or not.