Dependency Grammar

Grammar Formalisms for CL
SS 2010

Thanks to Markus Dickinson, Joakim Nivre and Sandra Kübler.

Dependency Grammar

- Not a coherent grammatical framework: wide range of different kinds of DG
  - just as there are wide ranges of "generative syntax"
- Different core ideas than phrase structure grammar
- We will base a lot of our discussion on [Mel'čuk(1988)]

Dependency grammar is important for those interested in CL:

- Increasing interest in dependency-based approaches to syntactic parsing in recent years (e.g., CoNLL-X shared task, 2006)

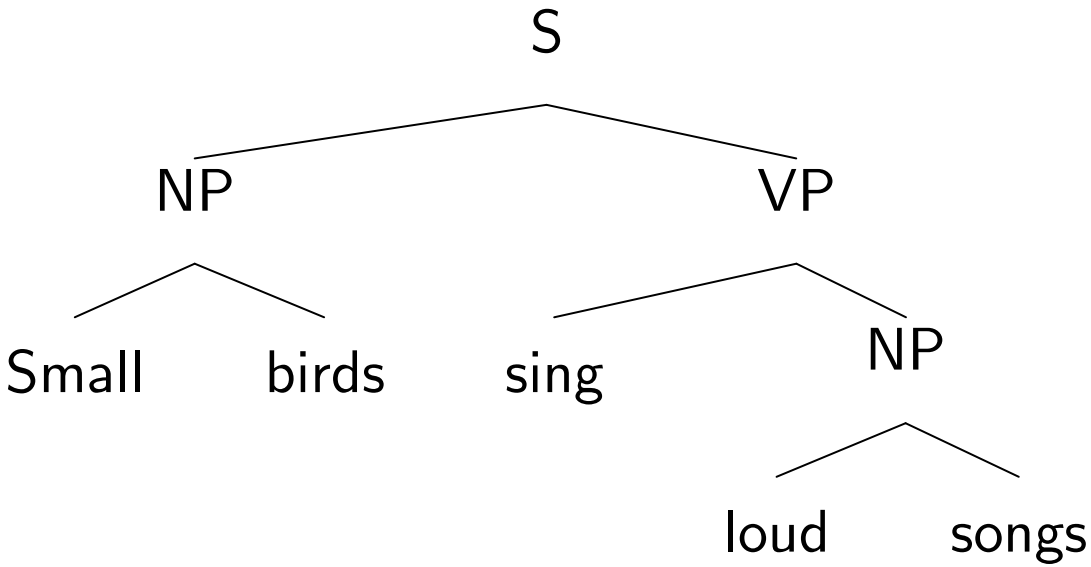
Dependency Syntax

- The basic idea:
  - Syntactic structure consists of lexical items, linked by binary asymmetric relations called dependencies.
- In the (translated) words of Lucien Tesnière [Tesnière(1959)]:
  - The sentence is an *organized whole*, the constituent elements of which are *words*. [1.2] Every word that belongs to a sentence ceases by itself to be isolated as in the dictionary. Between the word and its neighbors, the mind perceives *connections*, the totality of which forms the structure of the sentence. [1.3] The structural connections establish *dependency* relations between the words. Each connection in principle unites a *superior* term and an *inferior* term. [2.1] The superior term receives the name *governor*. The inferior term receives the name *subordinate*. Thus, in the sentence *Alfred parle* [. . .], *parle* is the governor and *Alfred* the subordinate. [2.2]

Overview: constituency

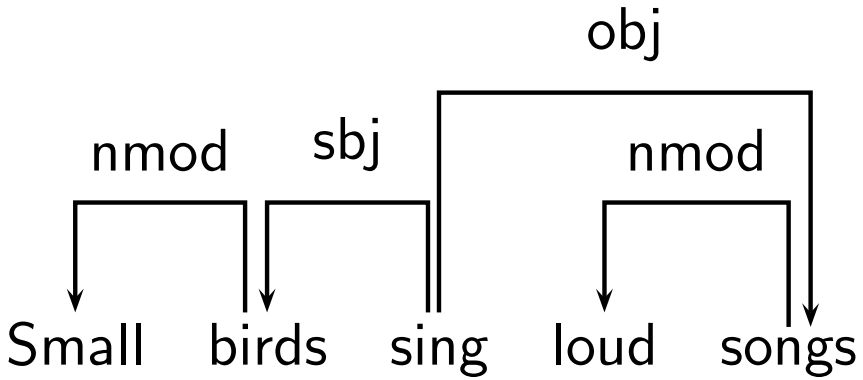
(1) Small birds sing loud songs

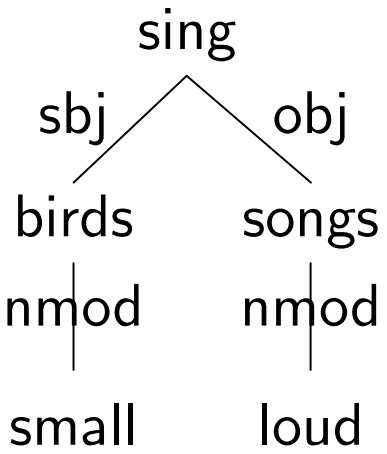
What you might be more used to seeing:



Overview: dependency

The corresponding dependency tree representations [Hudson(2000)]:





Constituency vs. Relations

- DG is based on relationships between words, i.e., **dependency relations**
  - A $\rightarrow$ B means *A governs B* or *B depends on A* ...
  - Dependency relations can refer to syntactic properties, semantic properties, or a combination of the two
    - $\rightarrow$ Some variants of DG separate syntactic and semantic relations by representing different layers of dependency structures
  - These relations are generally things like subject, object/complement, (pre-/post-)adjunct, etc.
    - Subject/Agent: *John* fished.
    - Object/Patient: Mary hit *John*.
- PSG is based on groupings (called *phrases* or *constituents*)
  - Grammatical relations are not usually seen as primitives, but as being derived from structure

Simple relation example

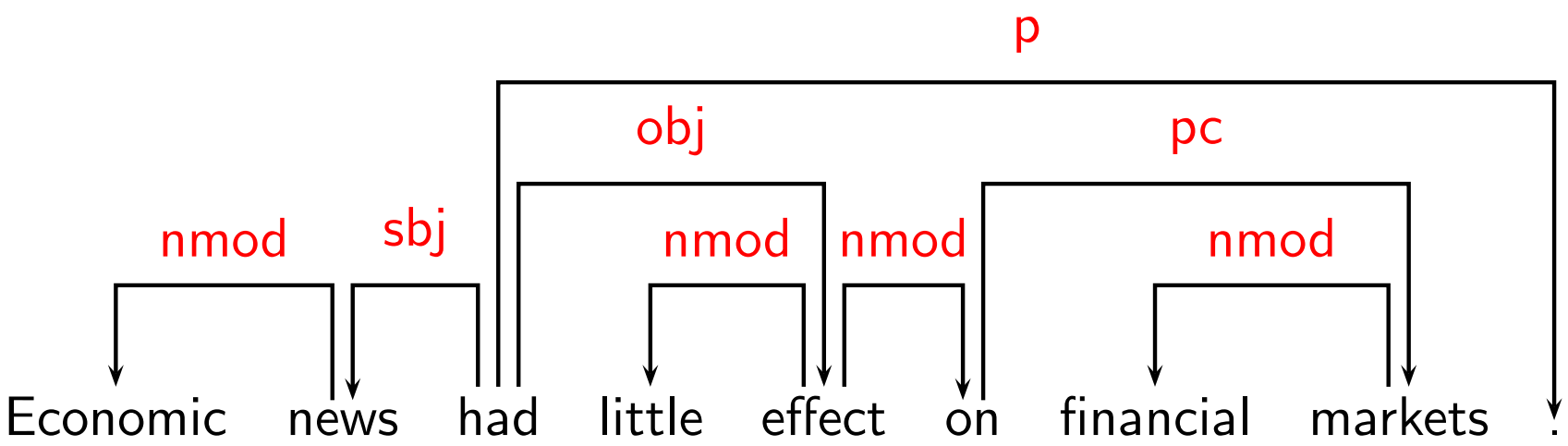
For the sentence *John loves Mary*, we have the relations:

- loves $\rightarrow_{\text{subj}}$ John
- loves $\rightarrow_{\text{obj}}$ Mary

Both *John* and *Mary* depend on *loves*, which makes *loves* the head, or **root**, of the sentence (i.e., there is no word that governs *loves*)

- The structure of a sentence, then, consists of the set of pairwise relations among words.

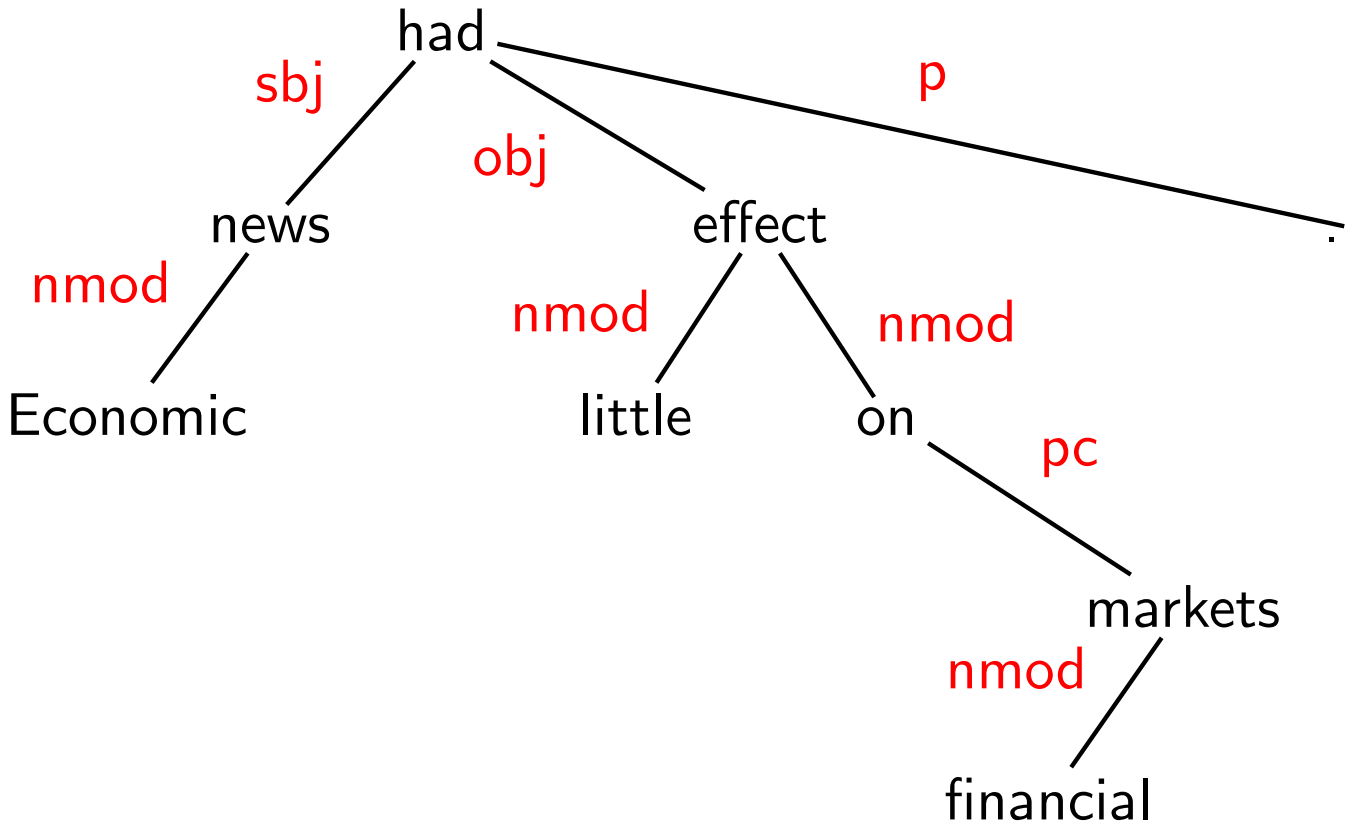
Dependency Structure



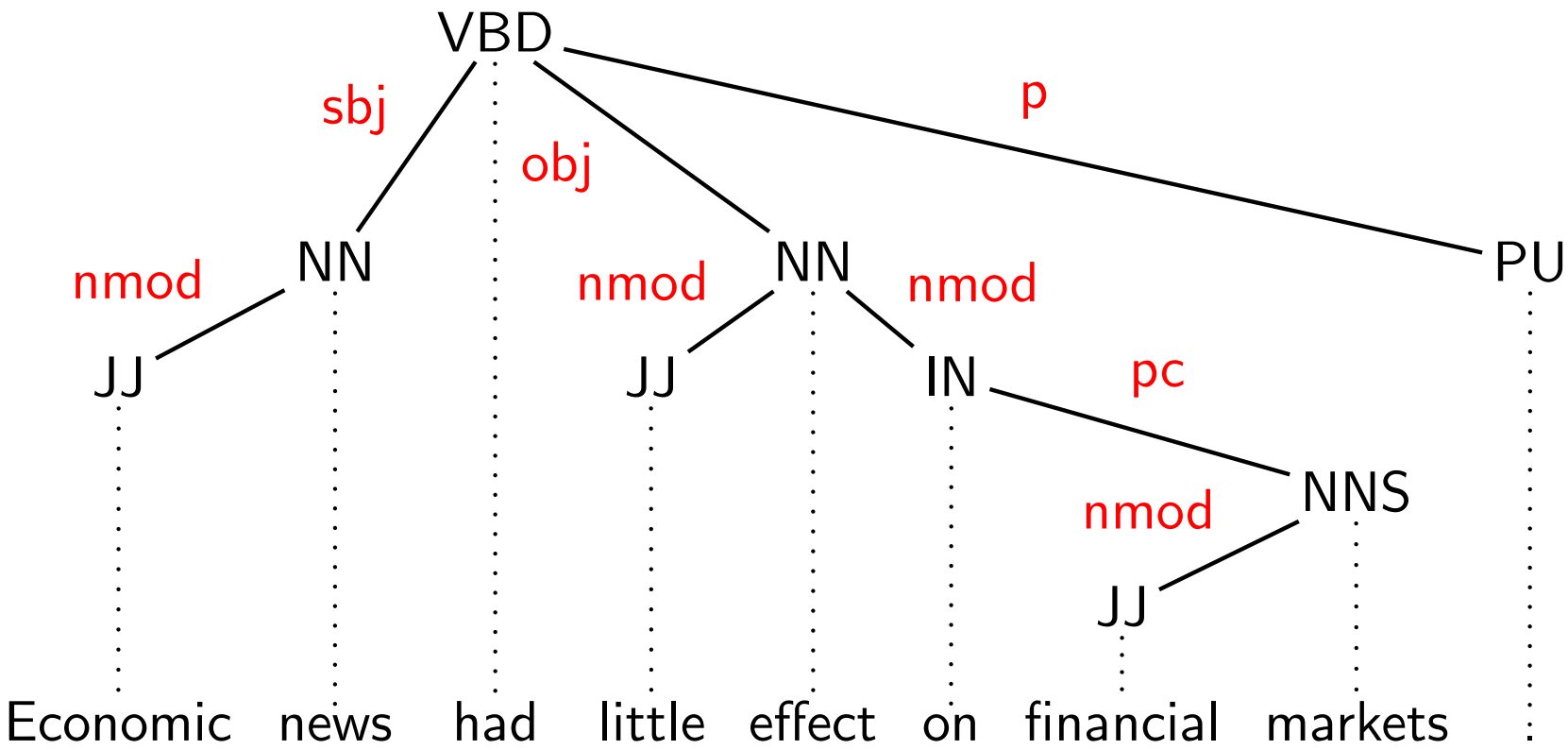
Terminology

Superior	**Inferior**
Head	Dependent
Governor	Modifier
Regent	Subordinate
⋮	⋮

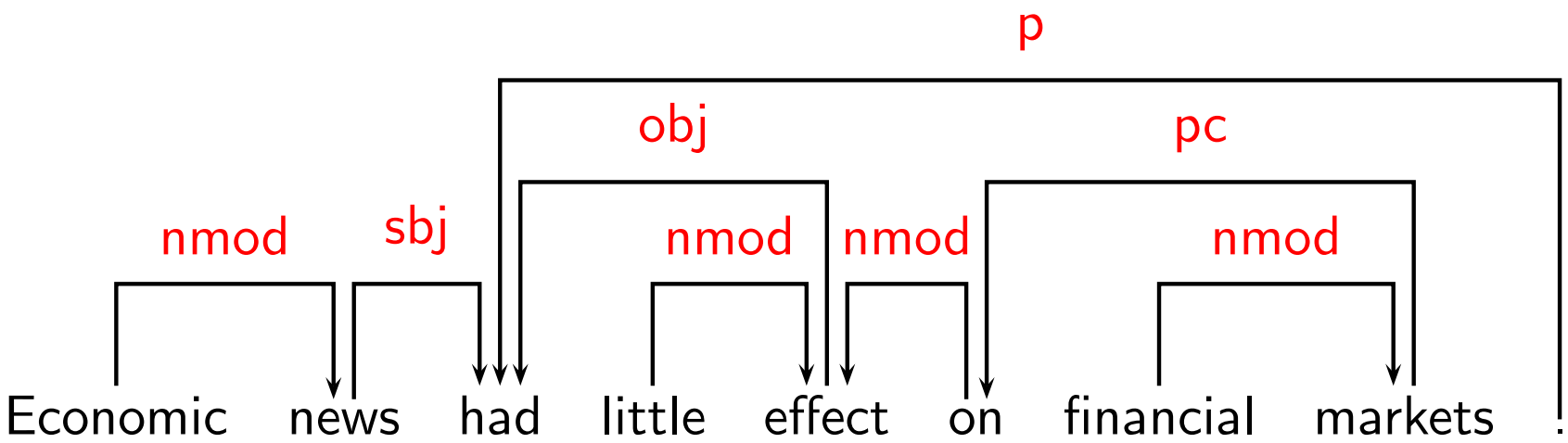
Notational Variants



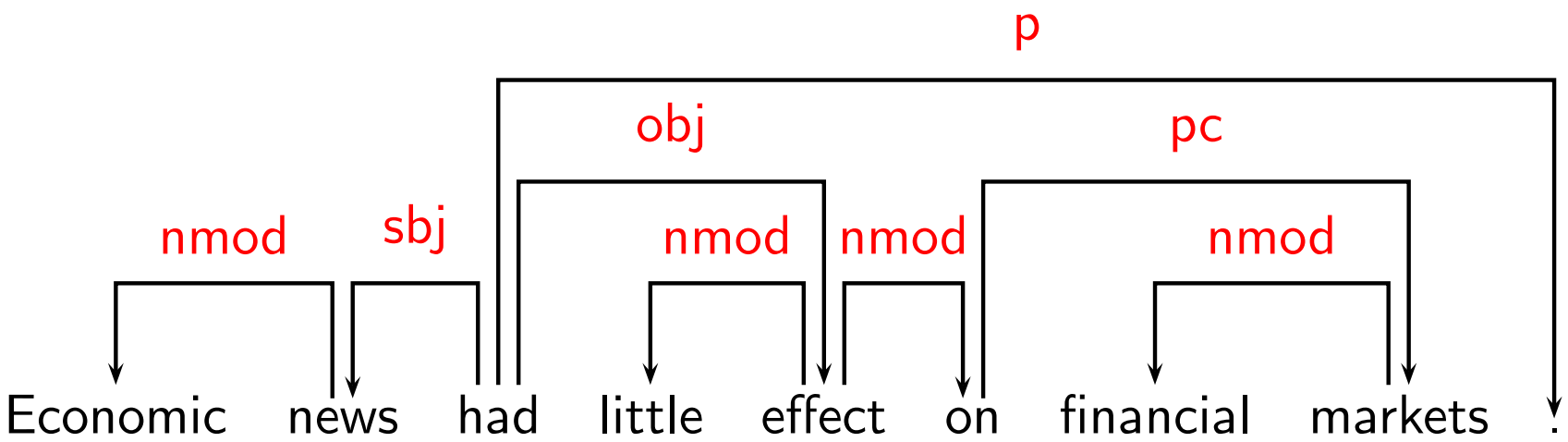
Notational Variants



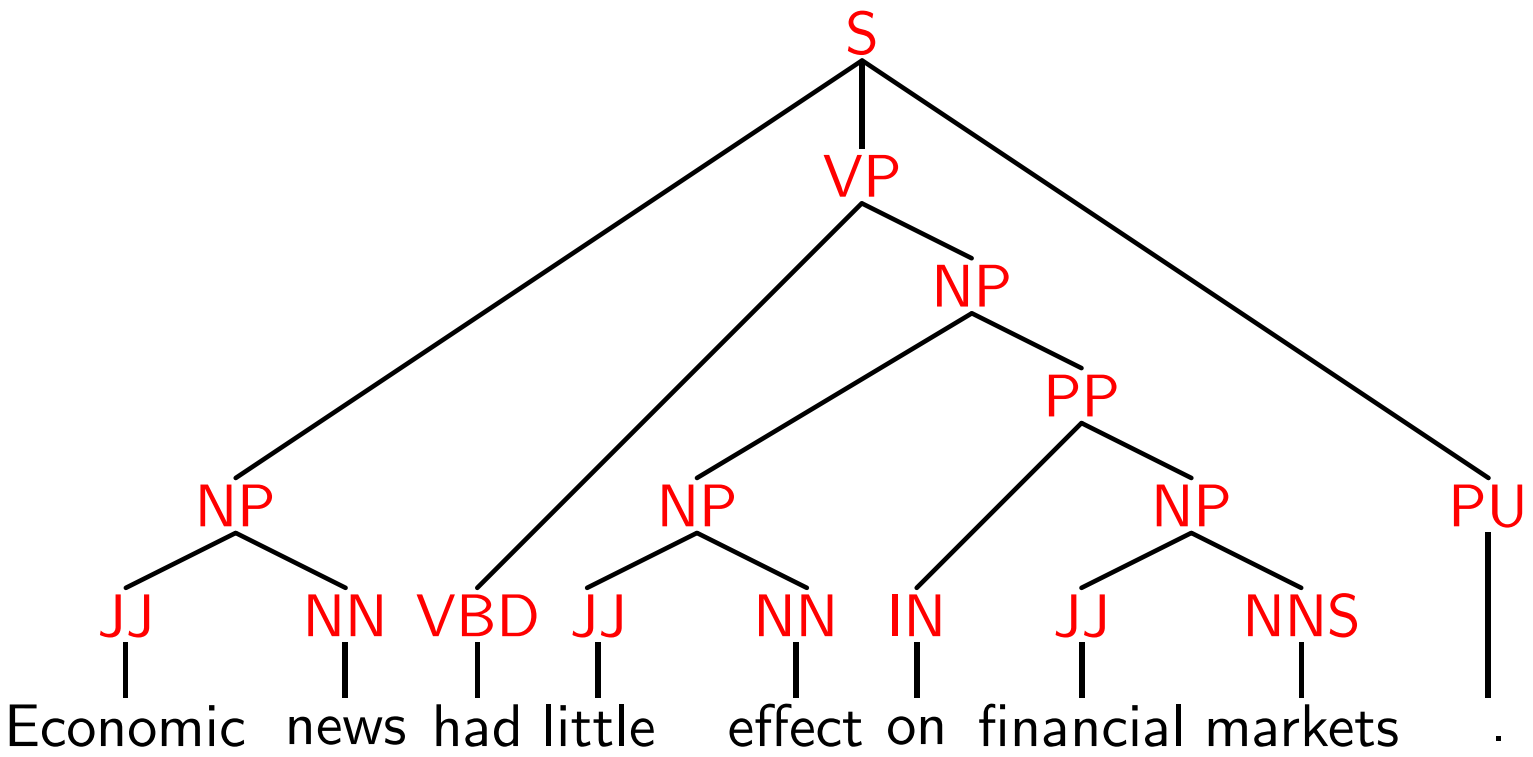
Notational Variants



Notational Variants



Phrase Structure



Comparison

- Dependency structures explicitly represent
  - head-dependent relations (directed arcs),
  - functional categories (arc labels),
  - possibly some structural categories (parts-of-speech).
- Phrase structures explicitly represent
  - phrases (nonterminal nodes),
  - structural categories (nonterminal labels),
  - possibly some functional categories (grammatical functions).
- Hybrid representations may combine all elements.

Some Theoretical Frameworks

- Word Grammar (WG) [Hudson(1984), Hudson(1990)]
- Functional Generative Description (FGD)
  [Sgall et al.(1986)Sgall, Hajičová and Panevová]
- Dependency Unification Grammar (DUG)
  [Hellwig(1986), Hellwig(2003)]
- Meaning-Text Theory (MTT) [Mel'čuk(1988)]
- (Weighted) Constraint Dependency Grammar ([W]CDG)
  [Maruyama(1990), Harper and Helzerman(1995),
  Menzel and Schröder(1998), Schröder(2002)]
- Functional Dependency Grammar (FDG)
  [Tapanainen and Järvinen(1997), Järvinen and Tapanainen(1998)]
- Topological/Extensible Dependency Grammar ([T/X]DG)
  [Duchier and Debusmann(2001),
  Debusmann et al.(2004)Debusmann, Duchier and Kruijff]

Some Theoretical Issues

- Dependency structure sufficient as well as necessary?
- Mono-stratal or multi-stratal syntactic representations?
- What is the nature of lexical elements (nodes)?
  - Morphemes?
  - Word forms?
  - Multi-word units?
- What is the nature of dependency types (arc labels)?
  - Grammatical functions?
  - Semantic roles?
- What are the criteria for identifying heads and dependents?
- What are the formal properties of dependency structures?

Some Theoretical Issues

- Dependency structure sufficient as well as necessary?
- Mono-stratal or multi-stratal syntactic representations?
- What is the nature of lexical elements (nodes)?
  - Morphemes?
  - Word forms?
  - Multi-word units?
- What is the nature of dependency types (arc labels)?
  - Grammatical functions?
  - Semantic roles?
- What are the criteria for identifying heads and dependents?
- What are the formal properties of dependency structures?

Capturing Adjuncts and Complements

There are two main kinds of dependencies for A $\rightarrow$ B:

- Head-Complement: if A (the head) has a slot for B, then B is a complement
- Head-Adjunct: if B has a slot for A (the head), then B is an adjunct

B is dependent on A in either case, but the selector is different

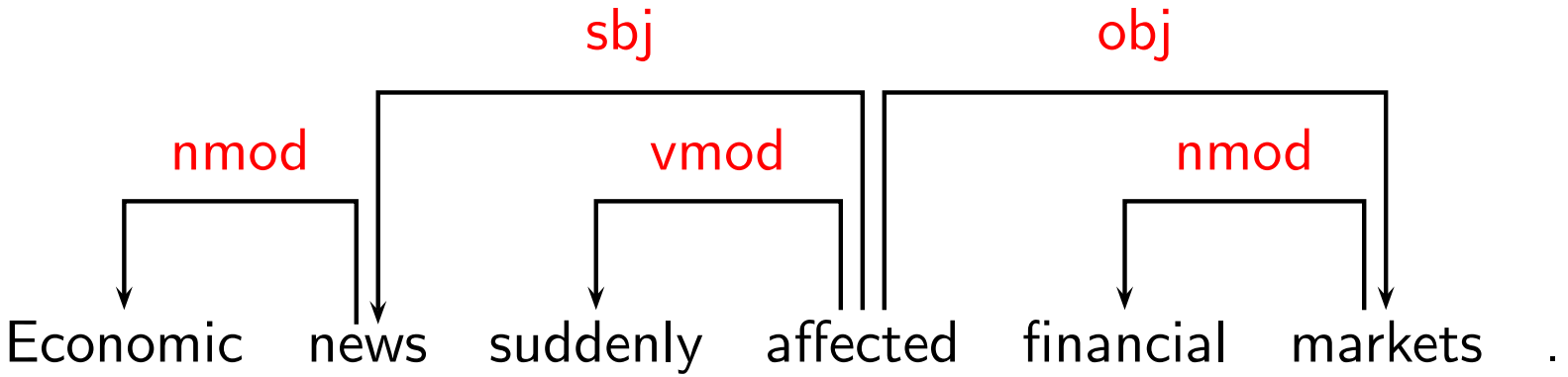
- The adjunct/complement distinction is captured in the type of dependency relation and/or in the lexicon

Criteria for Heads and Dependents

- Criteria for a syntactic relation between a head H and a dependent D in a construction C [Zwicky(1985), Hudson(1990)]:
  1. H determines the syntactic category of C; H can replace C.
  2. H determines the semantic category of C; C specifies H.
  3. H is obligatory; D may be optional.
  4. H selects D and determines whether D is obligatory.
  5. The form of D depends on H (agreement or government).
  6. The linear position of D is specified with reference to H.
- Issues:
  - Syntactic (and morphological) versus semantic criteria
  - Exocentric versus endocentric constructions

Some Clear Cases

Construction	Head	Dependent
Exocentric	Verb	Subject (sbj)
	Verb	Object (obj)
Endocentric	Verb	Adverbial (vmod)
	Noun	Attribute (nmod)

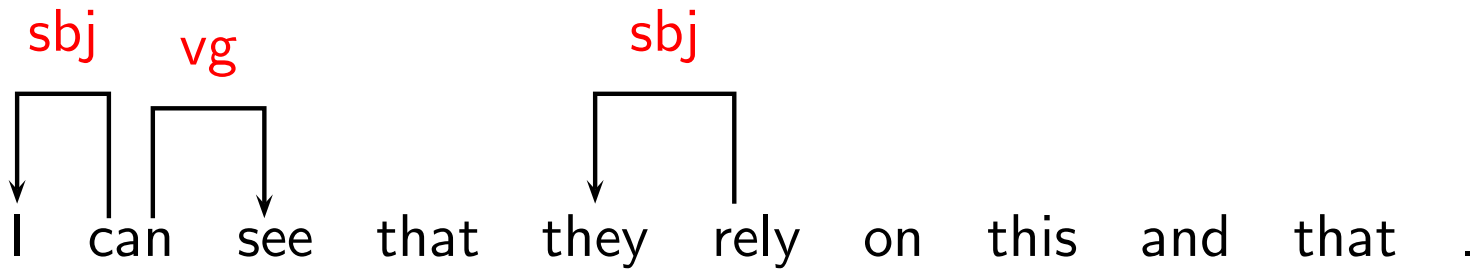


Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation

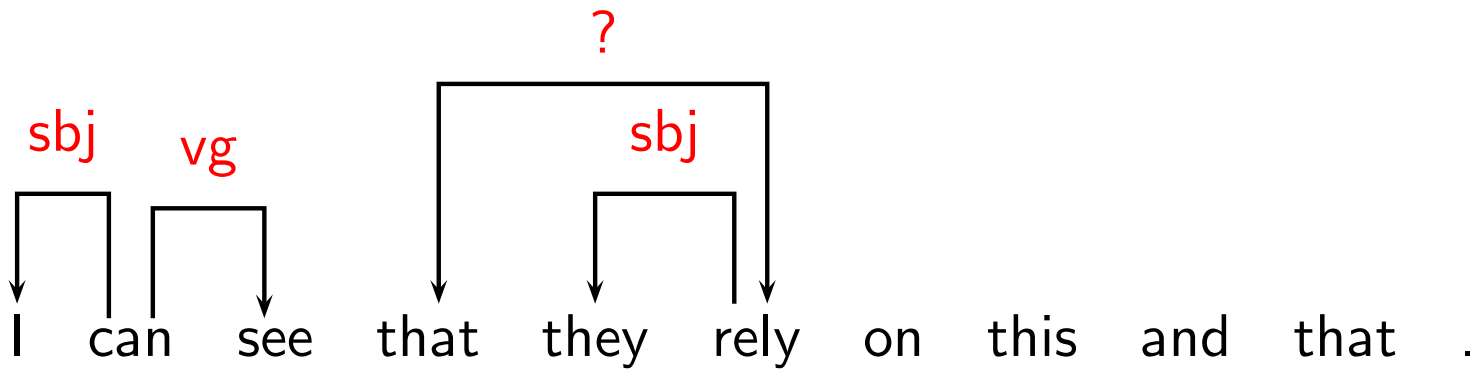
Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation



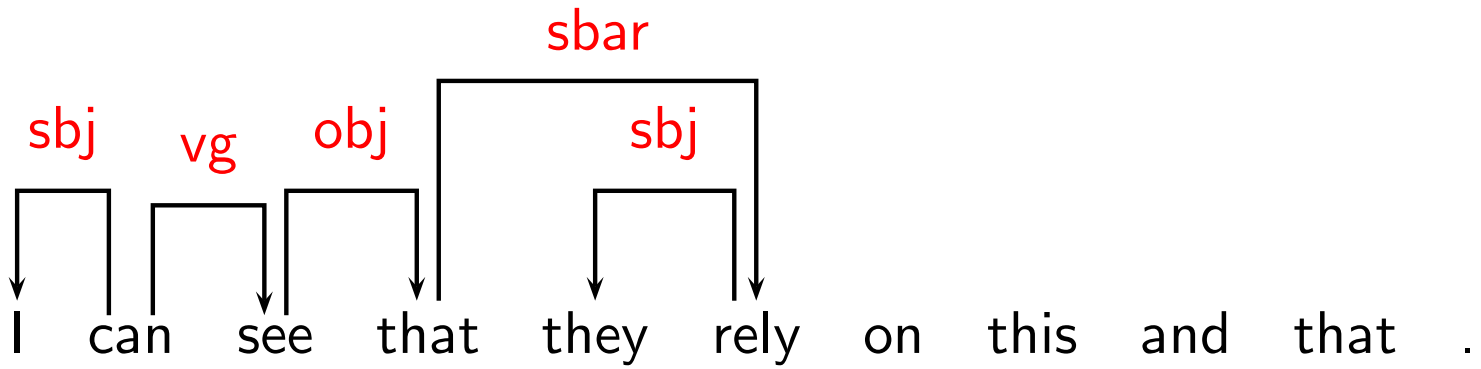
Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation



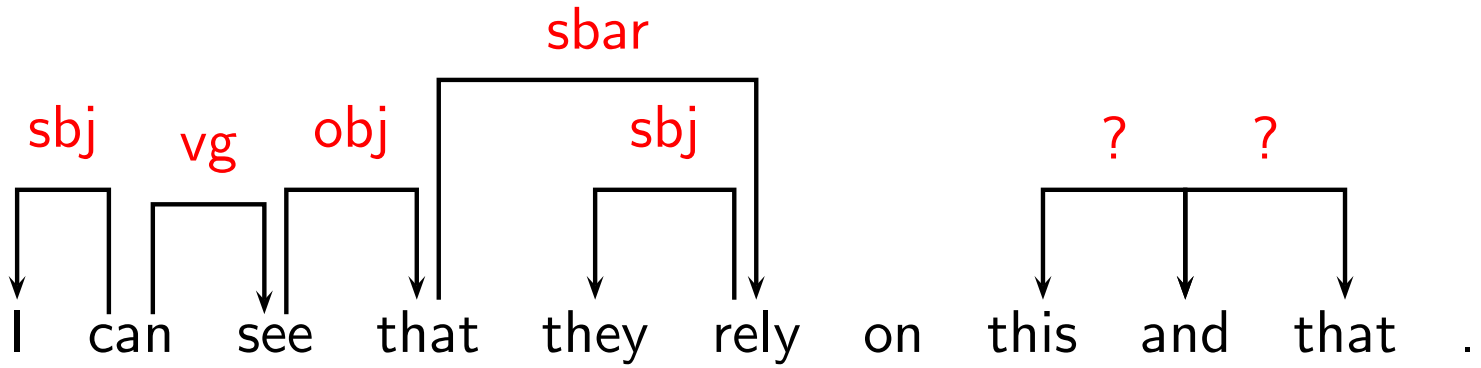
Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation



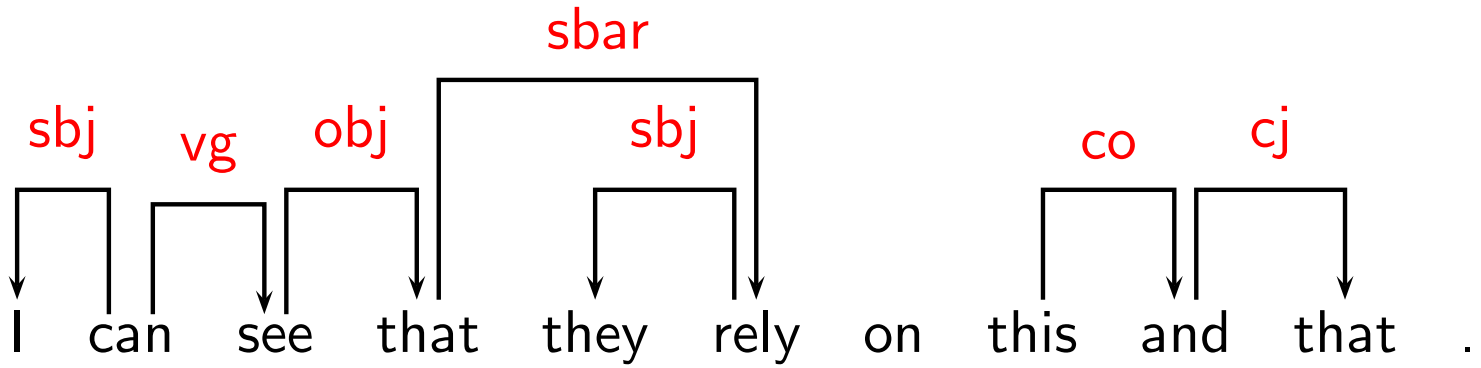
Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation



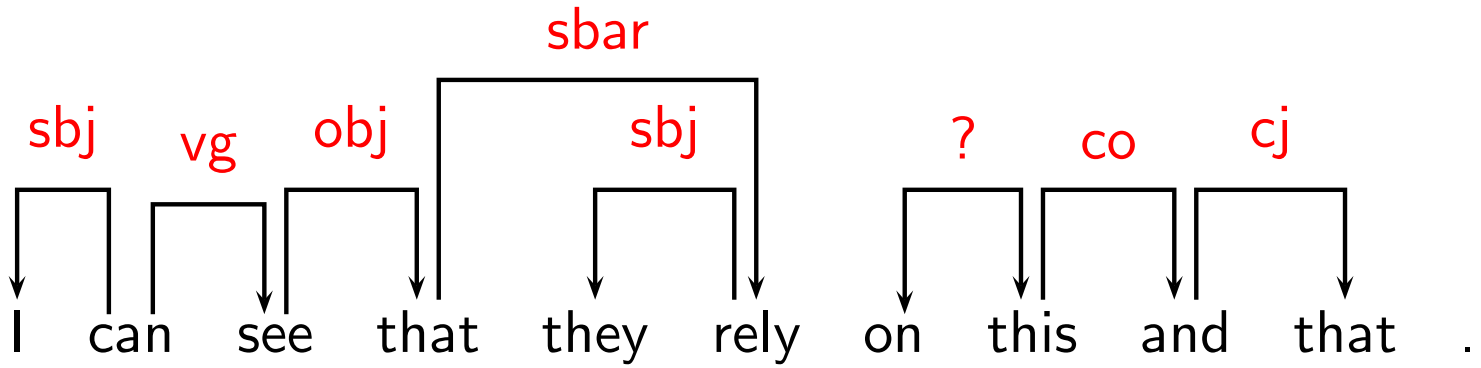
Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation



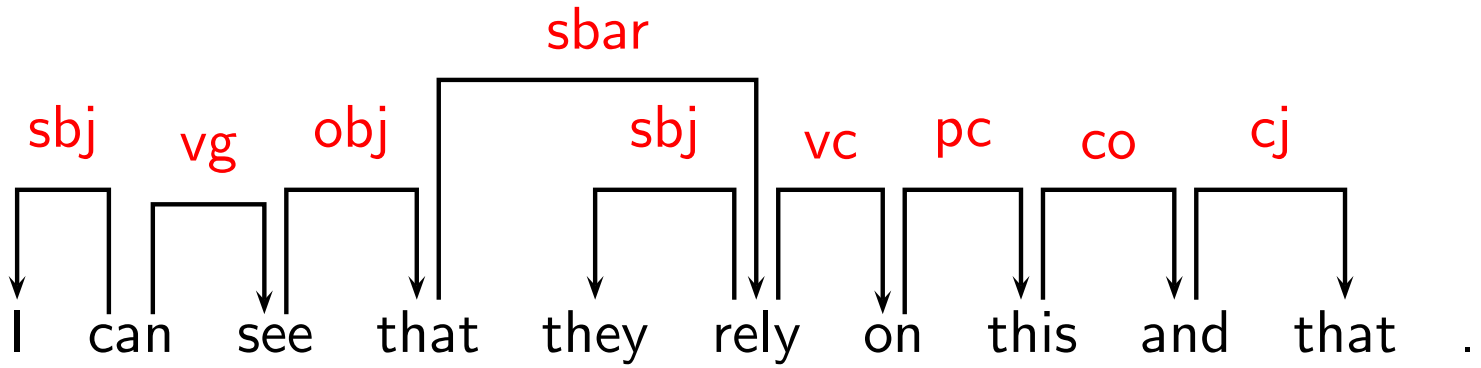
Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation



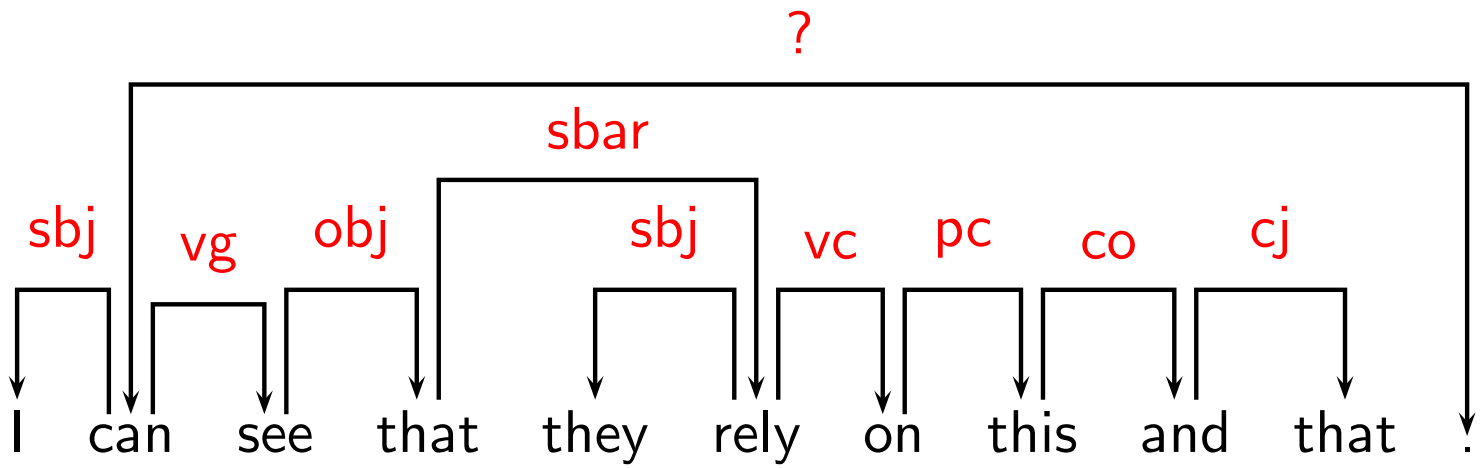
Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation



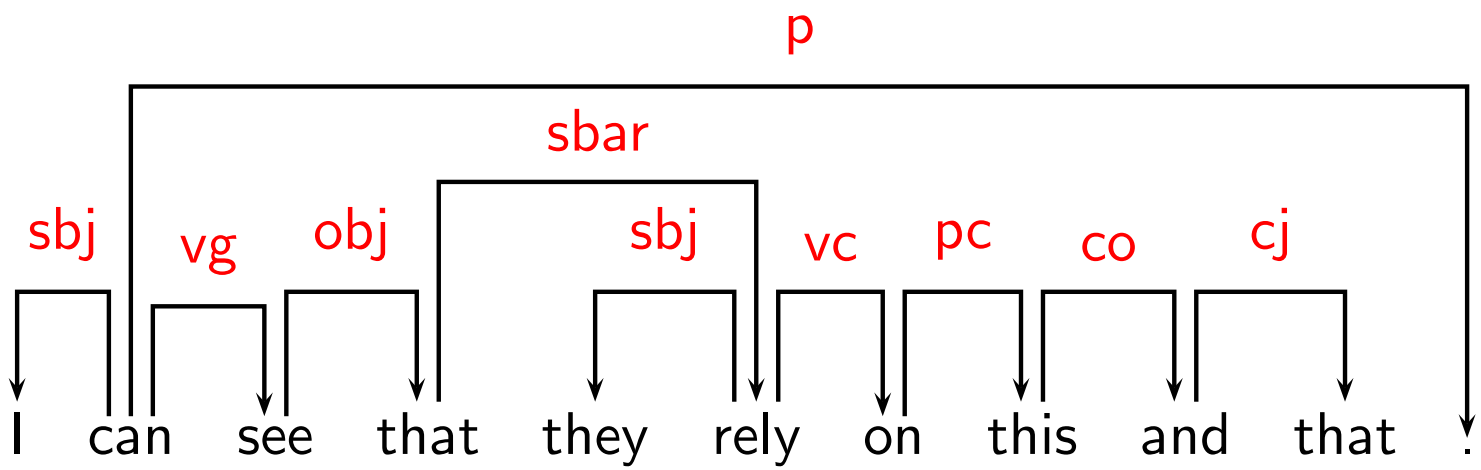
Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation



Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation



Dependency Graphs

- A dependency structure can be defined as a directed graph G, consisting of
  - a set V of nodes,
  - a set E of arcs (edges),
  - a linear precedence order $<$ on V
    (not in every theory)
- Labeled graphs:
  - Nodes in V are labeled with word forms (and annotation).
  - Arcs in E are labeled with dependency types.
- Notational conventions ($i, j \in V$):
  - $i \rightarrow j \;\equiv\; (i,j) \in E$
  - $i \rightarrow^* j \;\equiv\; i = j \vee \exists k : i \rightarrow k,\ k \rightarrow^* j$

Formal Properties of Dependency Graphs

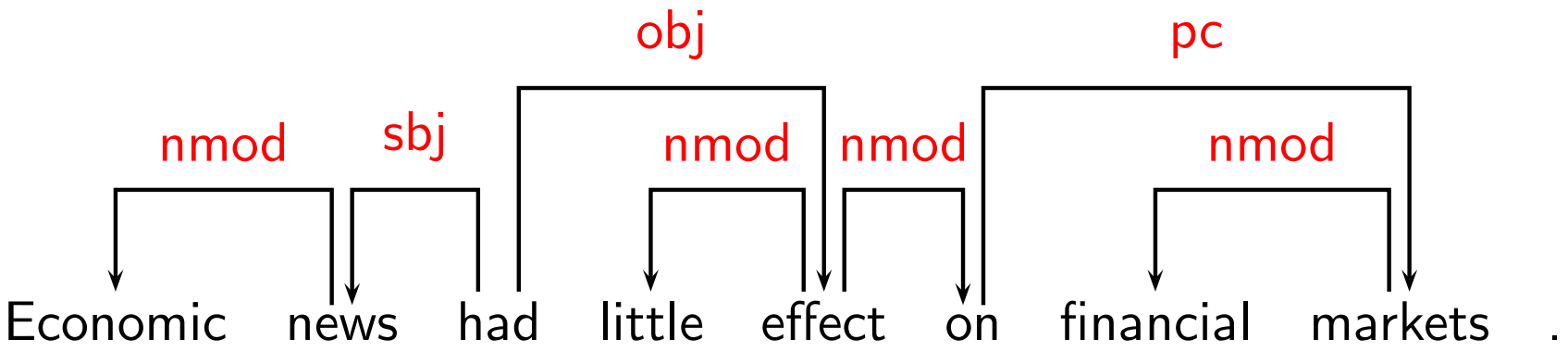
- **antisymmetric**: if A $\rightarrow$ B, then B $\nrightarrow$ A
  - cf. *box lunch* (lunch $\rightarrow$ box) vs. *lunch box* (box $\rightarrow$ lunch)
- **antireflexive**: if A $\rightarrow$ B, then B $\neq$ A
- **antitransitive**: if A $\rightarrow$ B and B $\rightarrow$ C, then A $\nrightarrow$ C
  - These are *direct* dependency relations
  - cf. *a usually reliable source*: source $\rightarrow$ reliable & reliable $\rightarrow$ usually, but source $\nrightarrow$ usually
- **labeled**: $\forall \rightarrow$, $\rightarrow$ has a label (r)

Formal Conditions on Dependency Graphs

- G is (weakly) connected:
  - For every node i there is a node j such that $i \rightarrow j$ or $j \rightarrow i$.
- G is acyclic:
  - If $i \rightarrow j$ then not $j \rightarrow^* i$.
- G obeys the single-head constraint:
  - If $i \rightarrow j$, then not $k \rightarrow j$, for any $k \neq i$.
- G is projective:
  - If $i \rightarrow j$ then $i \rightarrow^* k$, for any k such that $i < k < j$ or $j < k < i$.

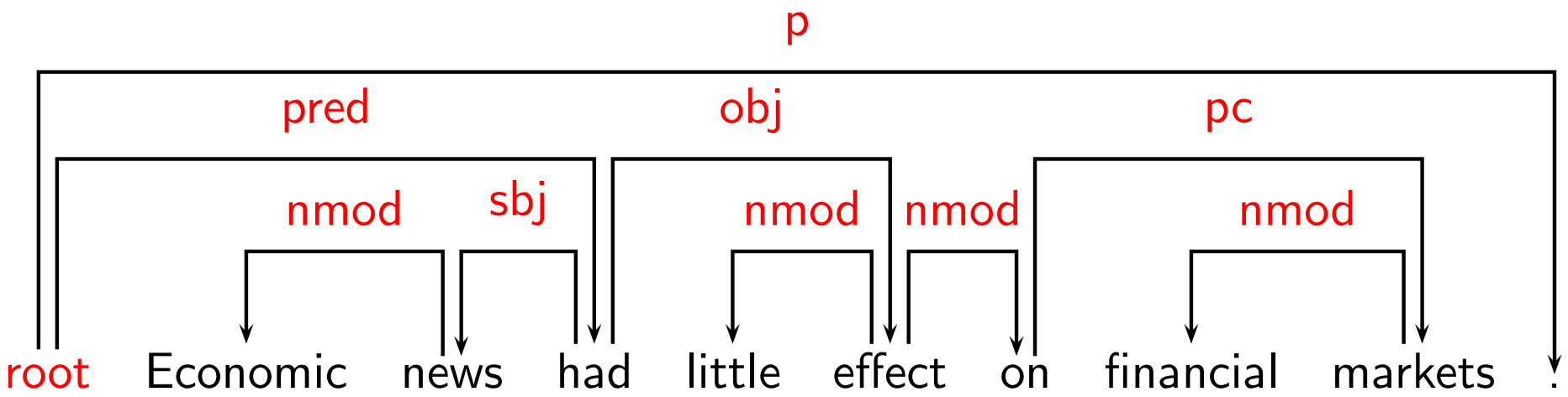
Connectedness, Acyclicity and Single-Head

- Intuitions:
  - Syntactic structure is complete (Connectedness).
  - Syntactic structure is hierarchical (Acyclicity).
  - Every word has at most one syntactic head (Single-Head).
- Connectedness can be enforced by adding a special root node.



Connectedness, Acyclicity and Single-Head

- Intuitions:
  - Syntactic structure is complete (Connectedness).
  - Syntactic structure is hierarchical (Acyclicity).
  - Every word has at most one syntactic head (Single-Head).
- Connectedness can be enforced by adding a special root node.



Projectivity

Projectivity (or, less commonly, **adjacency** [Hudson(1990)])

- A head (A) and a dependent (B) must be adjacent: A is adjacent to B provided that every word between A and B is a subordinate of A.

(2) with great difficulty

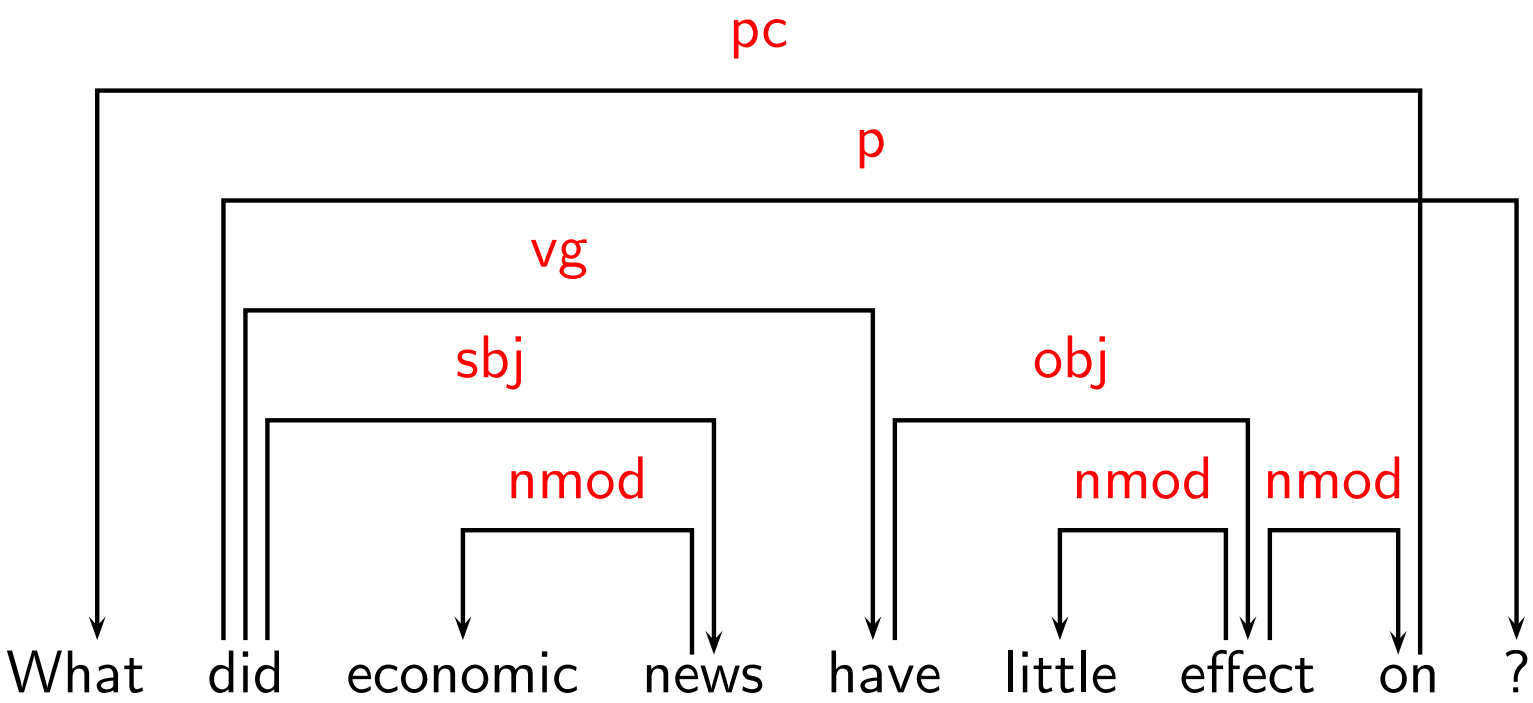
(3) *great with difficulty

- with $\rightarrow$ difficulty
- difficulty $\rightarrow$ great

**great with difficulty* is ruled out because branches would have to cross in that case

Projectivity

- Most theoretical frameworks do not assume projectivity.
- Non-projective structures are needed to account for
  - long-distance dependencies,
  - free word order.



Valency and Grammaticality

An important concept in many variants of DG is that of **valency** = the ability of a word to take arguments
A lexicon might look like the following
[Hajič et al.(2003)Hajič, Panevová, Urešová, Bémová, Kolářová and Pajas]:

	Slot_1	Slot_2	Slot_3
$sink_1$	ACT(nom)	PAT(acc)	
$sink_2$	PAT(nom)		
give	ACT(nom)	PAT(acc)	ADDR(dat)

To determine grammaticality (roughly) ...

1. Words have valency requirements that must be satisfied
2. Apply general rules to the valencies to see if a sentence is valid

Layers of dependencies

[Mel'čuk(1988)] allows for different dependency layers

It looks like a subject depends on the verb, but the form of the verb depends on the subject (mutual dependence):

(4) a. The child is playing.
    b. The children are playing.

Solution:

- Dependence of *child*/*children* on the verb is syntactic
- Dependence of the verb(form) on the subject is morphological

Double dependencies

Likewise, here it seems that *clean* depends both on the verb *wash* and on the noun *dish*

(5) Wash the dish *clean*.

Solution:

- Dependence of *clean* on *wash* is syntactic (cf. case)
- Dependence of *clean* on *dish* is semantic (cf. gender)

(6) My našli zal pust-ym
We found the hall$_{masc}$ empty$_{masc.sg.inst}$

Double dependencies (2)

Hudson's Word Grammar [Hudson(2004)] explicitly allows for **structure-sharing**, explicitly violating the single-head constraint:

- wash $\rightarrow$ clean
- dish $\rightarrow$ clean

NB: Hudson also uses this to account for non-projectivity, but we'll ignore the details.

Relation to phrase structure

After all this discussion, what is the relation between DG and PSG?

- If a PS tree has heads marked, then you can derive the dependencies
- Likewise, a DG tree can be converted into a PS tree by grouping a word with its dependents
  - But what the constituents are is still open (binary-branching, flat)
  - And phrases are not categorized

Advantages and Disadvantages of DG

Advantages:

- Close connection to semantic representation
- More flexible structure for, e.g., non-constituent coordination
- Easier to capture some typological regularities
- Vast & expanding body of computational work on dependency parsing

Disadvantages:

- No constituents makes analyzing coordination difficult
- No distinction between modifying a constituent vs. an individual word
- Harder to capture things like, e.g., subject-object asymmetries

- Debusmann, Ralph, Denys Duchier and Geert-Jan M. Kruijff (2004). Extensible Dependency Grammar: A New Methodology.
  In *Proceedings of the Workshop on Recent Advances in Dependency Grammar*. pp. 78–85.
- Duchier, Denys and Ralph Debusmann (2001). Topological Dependency Trees: A Constraint-based Account of Linear Precedence.
  In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*. pp. 180–187.
- Hajič, Jan, Jarmila Panevová, Zdeňka Urešová, Alevtina Bémová, Veronika Kolářová and Petr Pajas (2003). PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation.
  In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003)*. Växjö, Sweden, pp. 57–68.
  `http://w3.msi.vxu.se/~rics/TLT2003/doc/hajic_et_al.pdf`.
- Harper, Mary P. and R. A. Helzerman (1995). Extensions to constraint dependency parsing for spoken language processing.
  Computer Speech and Language 9, 187–234.
- Hellwig, Peter (1986). Dependency Unification Grammar.
  In *Proceedings of the 11th International Conference on Computational Linguistics (COLING)*. pp. 195–198.
- Hellwig, Peter (2003). Dependency Unification Grammar.

In Vilmos Agel, Ludwig M. Eichinger, Hans-Werner Eroms, Peter Hellwig, Hans Jürgen Heringer and Hening Lobin (eds.), *Dependency and Valency*, Walter de Gruyter, pp. 593–635.

- Hudson, Richard A. (1984). *Word Grammar*. Blackwell.
- Hudson, Richard A. (1990). *English Word Grammar*. Blackwell.
- Hudson, Richard A. (2000). Dependency Grammar Course Notes. `http://www.cs.bham.ac.uk/research/conferences/esslli/notes/hudson.\html`.
- Hudson, Richard A. (2004). Word Grammar. `http://www.phon.ucl.ac.uk/home/dick/intro.htm`.
- Järvinen, Timo and Pasi Tapanainen (1998). Towards an Implementable Dependency Grammar.
  In Sylvain Kahane and Alain Polguère (eds.), *Proceedings of the Workshop on Processing of Dependency-Based Grammars*. pp. 1–10.
- Maruyama, Hiroshi (1990). Structural Disambiguation with Constraint Propagation.
  In *Proceedings of the 28th Meeting of the Association for Computational Linguistics (ACL)*. pp. 31–38.
- Mel'čuk, Igor (1988). *Dependency Syntax: Theory and Practice*.

State University of New York Press.

- Menzel, Wolfgang and Ingo Schröder (1998). Decision Procedures for Dependency Parsing Using Graded Constraints.
  In Sylvain Kahane and Alain Polguère (eds.), *Proceedings of the Workshop on Processing of Dependency-Based Grammars*. pp. 78–87.
- Schröder, Ingo (2002). Natural Language Parsing with Graded Constraints.
  Ph.D. thesis, Hamburg University.
- Sgall, Petr, Eva Hajičová and Jarmila Panevová (1986). *The Meaning of the Sentence in Its Pragmatic Aspects*.
  Reidel.
- Tapanainen, Pasi and Timo Järvinen (1997). A non-projective dependency parser.
  In *Proceedings of the 5th Conference on Applied Natural Language Processing*. pp. 64–71.
- Tesnière, Lucien (1959). *Éléments de syntaxe structurale*.
  Editions Klincksieck.
- Zwicky, A. M. (1985). Heads.
  Journal of Linguistics 21, 1–29.