Grammatikformalismen

Vorlesung 2 (15.05.2012)

PD Dr.Valia Kordoni

Email: kordoni@coli.uni-sb.de

http://www.coli.uni-saarland.de/courses/grammatikformalismen/2012/

Feature Structures and Feature Structure Descriptions

Feature structure—

- Provides a description/representation of an object (linguistic or non-linguistic) by specifying information about its **attributes**
- Formally, it is a **directed acyclic graph** (DAG): a root node, directed edges corresponding to attributes, and end nodes corresponding to values. ("acyclic": no circular paths)

Feature Structures and Feature Structure Descriptions (cont.)

- More conveniently decribed using an **attribute-value matrix** (AVM):

(1)

$$\begin{bmatrix} \text{TITLE} & \textit{HPSG} \\ \text{AUTHOR1} & \textit{Pollard} \\ \text{AUTHOR2} & \textit{Sag} \\ \text{COVER} & \textit{blue} \\ \text{PAGES} & 440 \end{bmatrix}$$

Each row is an attribute-value pair, or **feature**. The order of the rows is unimportant.

Feature Structures and Feature Structure Descriptions (cont.)

- Within a single AVM, an attribute can only take one value. The following is an improper AVM; it does not describe any feature structure:

(2)

$$\begin{bmatrix} \text{NAME} & \textit{Sandy} \\ \text{SEX} & \textit{male} \\ \text{AGE} & 29 \\ \text{SEX} & \textit{female} \end{bmatrix}$$

It is common practice to refer to AVMs as "feature structures" although strictly speaking they are feature structure *descriptions* (FSDs).

Recursive Structure

The value of an attribute is not necessarily atomic; it can specify attributes of its own. In other words, in an AVM, the value of an attribute can be another AVM:

(3)

$$\begin{bmatrix} \text{NAME} & \textit{Sandy} \\ \text{AGE} & 29 \\ \text{ADDRESS} & \begin{bmatrix} \text{NUMBER} & 10 \\ \text{STREET} & \textit{Main} \\ \text{CITY} & \textit{Springfield} \end{bmatrix} \\ \text{FAVORITES} & \begin{bmatrix} \text{FOOD} & \textit{cheese} \\ \text{COLOR} & \textit{blue} \end{bmatrix} \end{bmatrix}$$

Recursive Structure (cont.)

In fact, the value of an attribute is *always* an AVM. Atomic values should be thought of as very simple AVMs that have no attributes.

In DAG terms: The end node of an attribute edge can itself be the root node of a feature structure, with attribute edges leaving from it. The edges leading from the root node of the entire DAG to an atomic value make up an "attribute path." E.g., the value of the path "ADDRESS | CITY" is *Springfield*.

An atomic value is a minimal DAG consisting of a single labelled node with no outward edges.

Structure sharing

Two attributes can share the same value. Two cases:

- **Type identity**
  The values are the same by accident, but in fact they are independently specified. There are really two values that happen to look the same. They could "just as easily" have been different.

- **Token identity**
  Some principle (physical law, fact about the world, grammatical rule) insists that the two attributes have identical values. There is only one value, and both attributes share it.

Structure sharing (cont.)

Example: Kim and Sandy are twins.

(4)

$$\begin{bmatrix} \text{TWIN1} & \begin{bmatrix} \text{NAME} & \textit{Kim} \\ \text{AGE} & \boxed{1}\ 29 \\ \text{FAVORITES} & \begin{bmatrix} \text{COLOR} & \textit{blue} \end{bmatrix} \end{bmatrix} \\ \text{TWIN2} & \begin{bmatrix} \text{NAME} & \textit{Sandy} \\ \text{AGE} & \boxed{1}\ 29 \\ \text{FAVORITES} & \begin{bmatrix} \text{COLOR} & \textit{blue} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Structure sharing (cont.)

It is a fact about the world that twins have the same age, so the values for AGE are token identical, or "structure-shared." This is notated with matching boxed indices. On the other hand, the fact that Kim and Sandy have the same favorite color is accidental; these values are not structure-shared.

In DAG terms—

- Type identity: The two attribute edges have distinct end nodes.
- Token identity: The two attribute edges point to the same end node (structure-sharing).

Structure sharing (cont.)

Structure sharing is not allowed to result in circular or cyclical paths:

(5)

$$\boxed{1}\begin{bmatrix} \text{NAME} & \textit{Sandy} \\ \text{SIBLING} & \begin{bmatrix} \text{NAME} & \textit{Kim} \\ \text{SIBLING} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

This is not a well-formed feature structure description because is does not describe a DAG (directed acyclic graph).

Partial descriptions

Feature structures (DAGs) are assumed to represent an object as fully as possible, by specifying all values for all attributes.

Feature structure descriptions (AVMs), on the other hand, can be **partial** descriptions, specifying some features and omitting others. An underspecified AVM can correspond to a set of feature structures. As the AVM becomes more fully specified, the set of DAGs it describes becomes smaller and smaller, as fewer and fewer of them are compatible with the information given in the AVM.

Subsumption

The fact that feature structure descriptions can be more or less informative allows us to define a partial ordering on the set of FSDs.

For two AVMs A and B, A **subsumes** B ($A \preceq B$) iff

- B is at least as informative as A. That is, all the attributes and values specified in A are also found in B. B might specify other features as well, but it must include all of the ones in A.
- The set of feature structures described by A is a superset of the set of feature structures described by B.

Subsumption (cont.)

Formal definition of subsumption:

(6) a. For atomic values a and b (remember these are considered to be AVMs), $a \preceq b \Leftrightarrow b \preceq a \Leftrightarrow a = b$.[1]

b. For non-atomic AVMs A and B, A $\preceq$ B iff

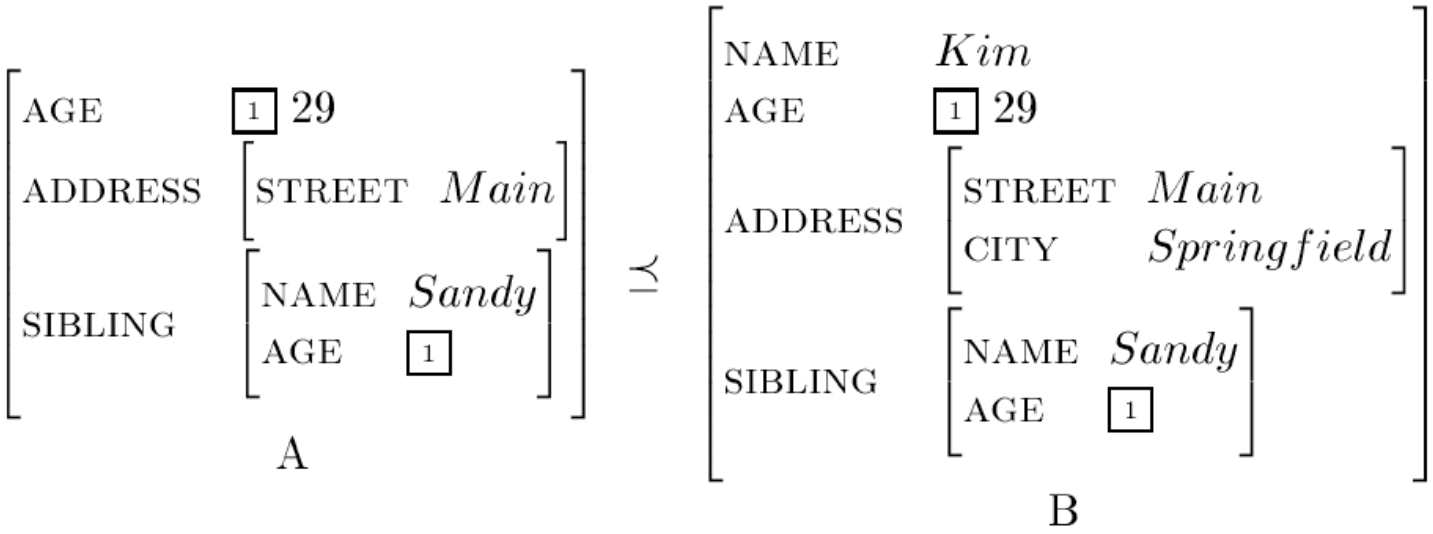
i. for every attribute path in A, the same path exists in B and its value in A subsumes its value in B

ii. for every pair of paths that is structure-sharing in A, the same pair of paths is structure-sharing in B.

[1] **NB**: This will change once we introduce *sorts*.

Subsumption (cont.)

(7)



Subsumption (cont.)

Notes:

- Subsumption is
  - reflexive: $A \preceq A$
  - transitive: if $A \preceq B$ and $B \preceq C$, then $A \preceq C$
  - anti-symmetric: if $A \preceq B$ and $B \preceq A$, then $A = B$.[2]

Subsumption (cont.)

- For two AVMs A and B, it is possible that neither one subsumes the other:

(8) a.

$$\underset{\text{A}}{\begin{bmatrix} \text{NAME} & \textit{Kim} \\ \text{AGE} & 29 \end{bmatrix}} \qquad \underset{\text{B}}{\begin{bmatrix} \text{AGE} & 30 \end{bmatrix}}$$

b.

$$\underset{\text{A}}{\begin{bmatrix} \text{NAME} & \textit{Sandy} \\ \text{JOB} & \textit{photographer} \end{bmatrix}} \qquad \underset{\text{B}}{\begin{bmatrix} \text{AGE} & 29 \\ \text{NATIONALITY} & \textit{British} \end{bmatrix}}$$

Subsumption (cont.)

- The minimum element with respect to the subsumption ordering is the feature structure that specifies no information at all (no attributes, no values). It is called "top" and is written $\top$ or [ ]. Top subsumes every other AVM, because every other AVM contains as least as much information as top: $\top \preceq A$ for all A.

Unification

Since AVMs can be partial descriptions, we might have several different descriptions of the same feature structure. It must be the case, though, that if two AVMs A and B are descriptions of the same object (or set of objects), then the information in A and B must be compatible. In particular, A and B cannot have different values for the same attributes or attribute paths.

[2] A and B are type identical, not necessarily token identical.

Unification (cont.)

(9) a.

$$\begin{bmatrix} \text{AGE} & 29 \\ \text{JOB} & \textit{photographer} \end{bmatrix} \quad \text{A} \qquad \begin{bmatrix} \text{NAME} & \textit{Sandy} \\ \text{AGE} & 29 \end{bmatrix} \quad \text{B}$$

b.

$$\begin{bmatrix} \text{AGE} & 29 \\ \text{JOB} & \textit{photographer} \end{bmatrix} \quad \text{A} \qquad \begin{bmatrix} \text{JOB} & \textit{photographer} \\ \text{AGE} & 30 \end{bmatrix} \quad \text{B}$$

Unification (cont.)

In the case where A and B contain compatible, consistent information, we say that the **unification** of A and B exists. The unification of A and B in (9a) is:

(10)

$$\begin{bmatrix} \text{NAME} & \textit{Sandy} \\ \text{AGE} & 29 \\ \text{JOB} & \textit{photographer} \end{bmatrix}$$

The unification of two feature structures contains all the information from both feature structures put together, but nothing more than that.

Unification (cont.)

Definition:

(11) For AVMs A and B, the unification C of A and B ($C = A \wedge B$) is the least informative AVM that is subsumed by both A and B.

(Even more formally: C is the least upper bound of A and B with respect to the subsumption ordering $\preceq$.)

Notes:

- The unification c of two atomic AVMs a and b is defined if and only if $a = b$ and in that case $c = a \wedge b = a = b$.[3]
- Unification works recursively on embedded AVMs.
- Unification preserves structure-sharing.

[3] **NB**: This will also change after we introduce sorts. (see fn. 1)

Unification (cont.)

(12)

$$\underbrace{\begin{bmatrix} \text{NAME} & Kim \\ \text{ADDRESS} & \begin{bmatrix} \text{CITY} & Springfield \end{bmatrix} \\ \text{SIBLING} & \begin{bmatrix} \text{NAME} & Sandy \\ \text{AGE} & 29 \end{bmatrix} \end{bmatrix}}_{\text{A}} \wedge \underbrace{\begin{bmatrix} \text{AGE} & \boxed{1} \\ \text{ADDRESS} & \begin{bmatrix} \text{STREET} & Main \end{bmatrix} \\ \text{SIBLING} & \begin{bmatrix} \text{NAME} & Sandy \\ \text{AGE} & \boxed{1} \end{bmatrix} \end{bmatrix}}_{\text{B}}$$

$$= \underbrace{\begin{bmatrix} \text{NAME} & Kim \\ \text{AGE} & \boxed{1}\ 29 \\ \text{ADDRESS} & \begin{bmatrix} \text{STREET} & Main \\ \text{CITY} & Springfield \end{bmatrix} \\ \text{SIBLING} & \begin{bmatrix} \text{NAME} & Sandy \\ \text{AGE} & \boxed{1} \end{bmatrix} \end{bmatrix}}_{\text{C}}$$

Unification (cont.)

If two AVMs A and B contain inconsistent feature specifications, then they "fail to unify," or their unification does not exist. Alternatively, we can write

(13) $\mathrm{A} \wedge \mathrm{B} = \bot$

where $\bot$ ("bottom") is an "improper" AVM that cannot describe any object (the opposite of $\top$, which can describe any object).

Unification (cont.)

As in the case of structure sharing, unification does not succeed if it results in cyclical structure:

(14)

$$\underbrace{\begin{bmatrix} \text{ATTRIBUTE1} & \begin{bmatrix} \text{ATTRIBUTE3} & \boxed{1} \end{bmatrix} \\ \text{ATTRIBUTE2} & \boxed{1} \end{bmatrix}}_{\text{A}} \wedge \underbrace{\begin{bmatrix} \text{ATTRIBUTE1} & \boxed{2} \\ \text{ATTRIBUTE2} & \begin{bmatrix} \text{ATTRIBUTE3} & \boxed{2} \end{bmatrix} \end{bmatrix}}_{\text{B}}$$

$$= \underbrace{\begin{bmatrix} \text{ATTRIBUTE1} & \boxed{2}\begin{bmatrix} \text{ATTRIBUTE3} & \boxed{1} \end{bmatrix} \\ \text{ATTRIBUTE2} & \boxed{1}\begin{bmatrix} \text{ATTRIBUTE3} & \boxed{2} \end{bmatrix} \end{bmatrix}}_{\text{C}} = \bot$$

Unification (cont.)

Properties of unification (from Pollard and Sag (1987), p. 38):

- idempotency: $A \wedge A = A$
- commutativity: $A \wedge B = B \wedge A$
- associativity: $(A \wedge B) \wedge C = A \wedge (B \wedge C)$
- $\top \wedge A = A$
- $\bot \wedge A = \bot$
- link between subsumption and unification: $A \preceq B \Leftrightarrow A \wedge B = B$

References

Barwise, J., & Perry, J. (1983). *Situations and attitudes.* Cambridge, MA: MIT Press.

Bouma, G., Malouf, R., & Sag, I. (1998). A unified theory of complement, adjunct, and subject extraction. In G. Bouma, G.-J. M. Kruijff, & R. T. Oehrle (Eds.), *Proceedings of the joint conference on Formal Grammar, Head-Driven Phrase Structure Grammar, and Categorial Grammar* (pp. 83–97). Saarbrücken: .

Copestake, A., Flickinger, D., & Sag, I. A. (1997). *Minimal Recursion Semantics: An introduction.*

Davis, A. (1997). *Lexical semantics and linking in the hierarchical lexicon.* Unpublished doctoral dissertation, Stanford University.

Gazdar, G., Klein, E., Pullum, G. K., & Sag, I. A. (1985). *Generalized Phrase Structure Grammar.* Cambridge, MA: Harvard University Press.

References (cont.)

Goldberg, A. E. (1995). *Constructions: A Construction Grammar approach to argument structure.* Chicago: University of Chicago Press.

Haegeman, L. (1994). *Introduction to Government and Binding Theory* (2nd ed.). Oxford: Blackwell.

Kamp, H., & Reyle, U. (1993). *From discourse to logic.* Dordrecht: Kluwer.

Kaplan, R., & Bresnan, J. (1982). Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan (Ed.), *The mental representation of grammatical relations* (p.). Cambridge, MA: MIT Press.

McGee Wood, M. (1993). *Categorial grammars.* London: Routledge.

Pollard, C., & Sag, I. A. (1987). *Information-based syntax and semantics.* Stanford: CSLI.

References (cont.)

Sag, I. A. (1997). English relative clause constructions. *Journal of Linguistics*, *33*, 431–484.

Sag, I. A., & Miller, P. H. (To appear). French clitic movement without clitics or movement. *Natural Language and Linguistic Theory*.

Scalise, S. (1984). *Generative morphology*. Dordrecht: Foris.

Wechsler, S. (1995). *The semantic basis of argument structure*. Stanford: CSLI.