Feature Graph Posets

Jean Mark Gawron
Mathematical Linguistics
Figure 1: A simple “partial description” feature graph A:
A(gen) = fem, A(case)=acc, A(num)=pl
A feature graph is a function from a Domain, a set of graph labels, to a set of feature graphs. We allow for a totally unspecified feature graph called \( \bot \) whose domain is the empty set \( \emptyset \) and we allow for terminal graphs, or terminals, such as acc, plu, and fem. The domain of the graph in Figure 1 is

\[
\{\text{case, gen, num}\}
\]

The values of the graph function are:

\[
\begin{align*}
A(\text{case}) &= \text{acc} \\
A(\text{gen}) &= \text{fem} \\
A(\text{num}) &= \text{plu}
\end{align*}
\]
Attribute-value matrices: a nicer notation

\[
\begin{bmatrix}
\text{gen} & \text{fem} \\
\text{case} & \text{acc} \\
\text{num} & \text{plu}
\end{bmatrix}
\]
More complex structures
Lists (ordered information)

1. A list
   \[ a, b, c \]

2. A different list
   \[ b, a, c \]
AVM representation of lists

Two attributes

Values

The empty list

\[ \langle \rangle \]

first always an element

rest always a list

\[ \begin{bmatrix}
[ \text{first} & \text{a} ] \\
[ \text{rest} & \text{b} ] \\
[ \text{rest} & \text{c} ] \\
[ \text{rest} & \langle \rangle ]
\end{bmatrix} \]
Verb arguments

John eats carrots

Beans Obj 1st argument

John Subj 2nd argument
Subcategorization

- Subcategorizes for a PP:
  (1) a. John doted on Mary.
     b. * John doted Mary.

- Subcategorizes for an NP (transitive):
  (2) a. John worshipped Mary,
     b. * John worshipped on Mary.

- Subcategorizes for nothing (intransitive):
  (3) a. John worshipped Mary,
     b. * John worshipped on Mary.
Verbs: Arguments as a list

- dotes
  - args
    - first pp
    - rest
      - first np
      - rest ⟨ ⟩

- worships
  - args
    - first np
    - rest
      - first np
      - rest ⟨ ⟩

- walks
  - args
    - first np
    - rest ⟨ ⟩
NPs

the
carrot

[cat
  case
  gen
  num      np
  sing]

the
carrots

[cat
  case
  gen
  num      np
  sing]

the
sheep

[cat
  case
  gen
  num      np]

him

[cat
  case
  gen
  num      np
  sing]

[cat
  case
  gen
  num      pl]

[cat
  case
  acc
  num]

[cat
  case
  acc
  num      masc
  sing]
Agreement, Case, Gender

- **Agreement:**
  (4) a. John eats beans. [sg.verb / sg.subj]
  b. * John eat beans. [pl.verb / sg.subj]

- **Case:**
  (5) a. John sees him. [Acc(ussative) Object]
  b. * John sees he. [Nom(inative) Object]

- **Gender:**
  (6) a. John shaved himself.
  b. * John shaved herself.
Verbs: arg specification

eats

\[
\begin{align*}
\text{args} & \quad \text{first} & \quad \text{rest} \\
\text{num} & & \text{sing} \\
\text{cat} & & \text{v}
\end{align*}
\]

\[
\begin{align*}
\text{cat} & \quad \text{np} \\
\text{case} & \quad \text{acc} \\
\text{num} & \quad \text{nom} \\
\text{sing}
\end{align*}
\]
Verbs: ate

Verbs: ate

ate

args

rest

num

cat

v

Feature Graph Posets – p. 13/44
When verb and arguments are combined, the information from both is *merged* or *unified*:

```
eats the carrot
```

Roman: from verb

Italics: from argument

Boldface: from both (agreement)
A sentence

The sheep eats the carrot

Feature Graph Posets – p. 15/44
Some number facts

(7) a. *sheep* (unspecified)
    b. *walked* (unspecified)
    c. The sheep walked. (unspecified)
    d. The sheep walked and are very tired. (plural)
    e. The sheep walked and is very tired. (singular)
Some agreement consequences

(8) a. The sheep that were in the barn walked. (plural)

b. The sheep that were in the barn walked and are very tired. (plural)

c. * The sheep that were in the barn walked and is very tired. (⊥)
An ordering on information

More information in the graph for *walks* than in the graph for *walked*
Another case of more information

The graph for *walked* has more information than one missing the agreement constraint:

\[
\begin{align*}
\text{subcat} & \quad \begin{cases}
\text{first} & \quad \begin{cases}
\text{cat np} & \quad \begin{cases}
\text{case acc} & \quad \begin{cases}
\text{num} & \quad \begin{cases}
\text{rest ⟨⟩} & \quad \begin{cases}
\text{num} & \quad [3] \\
\text{rest ⟨⟩} & \quad [3]
\end{cases}
\end{cases}
\end{cases}
\end{cases}
\end{cases}
\end{align*}
\]
Subsumption: A Partial Order on Information
Examples of Subsumption

\[
\begin{bmatrix}
\text{bar} & b \\
\end{bmatrix} \sqsubseteq 
\begin{bmatrix}
\text{foo} & a \\
\text{bar} & b \\
\end{bmatrix}
\]
Identified values

\[
\begin{bmatrix}
\text{foo} \\
\text{bar}
\end{bmatrix} \sqsubseteq 
\begin{bmatrix}
\text{foo} & \downarrow \\
\text{bar} & \downarrow
\end{bmatrix}
\]
\( \top \) and \( \bot \)

- \( \bot \) is the empty feature structure (a description everything satisfies). For the previous poset, the following are equivalent:
  - \( \bot \)
  - \[
    \begin{array}{c}
      \text{foo} \\
      \text{bar}
    \end{array}
  \]
  - \[
    \begin{array}{c}
      \text{foo} \\
      \text{bar}
    \end{array}
  \]
  - \[
    \begin{array}{c}
      \text{foo} \quad 1 \\
      \text{bar} \quad 2
    \end{array}
  \]

- \( \top \) is not a feature structure (not a satisfiable description)
Paths

\[
\begin{align*}
\langle \text{args} \rangle & \\
\langle \text{args} \text{ first} \rangle & \\
\langle \text{args} \text{ first cat} \rangle & \\
\langle \text{args} \text{ first cat np} \rangle & \\
\langle \text{args} \text{ first case} \rangle & \\
\langle \text{args} \text{ first case acc} \rangle & \\
\langle \text{args} \text{ rest} \rangle & \\
\langle \text{args} \text{ rest first} \rangle & \\
\langle \text{args} \text{ rest first cat} \rangle & \\
\langle \text{args} \text{ rest first cat np} \rangle & \\
\langle \text{args} \text{ rest first case} \rangle & \\
\langle \text{args} \text{ rest first case nom} \rangle & \\
\langle \text{args} \text{ rest first num} \rangle & \\
\langle \text{args} \text{ rest rest} \rangle & \\
\langle \text{args} \text{ rest rest} \langle \rangle \rangle & \\
\langle \text{num} \rangle &
\end{align*}
\]
Dom and Paths

We write

$$\text{Dom}(A)$$

$$\text{Paths}(A)$$

for the domain and paths of feature structure $A$. We write

$$\text{Paths}_1(A)$$

$$\text{Paths}_2(A)$$

$$\text{Paths}_{\leq 2}(A)$$

for the paths of length 1, length 2, and length less than 2. Note:

$$\text{Dom}(A) = \text{Paths}_1(A)$$
For a path $P$ and a feature graph $S$ we write $S(P)$ for the value of a path, or just $P$, when $S$ is understood.

$$
S(\langle \text{args} \rangle) =
\begin{bmatrix}
\text{first} & \text{cat} & \text{np} \\
\text{case} & \text{acc} \\
\text{rest} & \text{cat} & \text{np} \\
\text{first} & \text{case} & \text{nom} \\
\text{rest} & \text{num} & 1 \\
\text{rest} & \langle \rangle \\
\text{num} & 1
\end{bmatrix}
$$

$$
S(\langle \text{args first} \rangle) =
\begin{bmatrix}
\text{cat} & \text{np} \\
\text{case} & \text{acc} \\
\text{rest} & \text{cat} & \text{np} \\
\text{rest} & \text{case} & \text{nom} \\
\text{rest} & \text{num} & 1 \\
\text{rest} & \langle \rangle \\
\text{num} & 1
\end{bmatrix}
$$

$$
S(\langle \text{args rest first num} \rangle) = S(\langle \text{num} \rangle)
$$
More path values

\[
\begin{bmatrix}
\text{args} \\
\text{first} \\
\text{case} \\
\text{nom} \\
\text{num} \\
\text{rest} \\
\langle \rangle
\end{bmatrix}
\begin{bmatrix}
\text{cat} \\
\text{np} \\
\text{nom} \\
1
\end{bmatrix}
\]

\[S(\langle \text{args first num} \rangle) = 1\]

The value of the path \(\langle \text{args first num} \rangle\) is unspecified in this feature graph. We call such a value a variable.
Subsumption: An information ordering

We define subsumption, $\sqsubseteq$, as an information ordering on feature structures.

- The ordering respects the intuitions of our poset of feature structures.
- $A \sqsubseteq B$ (reads “$A$ subsumes $B$”) means $B$ has MORE information than $A$, $A$ is more general than $B$.
- Roughly: The information available along all the paths in $A$ is a subset of the information available along all the paths in $B$.
- This order is partial. There are many incomparable feature structures.
Formal definition of subsumption

For any two feature structures $D_1$ and $D_2$, $D_1 \sqsubseteq D_2$ if and only if

(a) For all attributes $a \in \text{Dom}(D_1)$, $D_1(a) \sqsubseteq D_2(a)$

(b) For all paths $p, q \in \text{Paths}(D_1)$,
   If $D_1(p) = D_1(q)$ then $D_2(p) = D_2(q)$
(B.1) For any two terminals a and b, if $a \neq b$, then $a \not\sqsubseteq b$, $b \not\sqsubseteq a$.

(B.2) For any two variables $1$ and $2$, $1 \sqsubseteq 2$, $2 \sqsubseteq 1$.

(B.3) For any terminal $a$ and variable $1$, $1 \sqsubseteq a$, $a \not\sqsubseteq 1$. 
The lattice connection

We have argued that $\sqsubseteq$ is a weak order. It is possible to show that the subsumption poset is a lattice.
Unification: Merging information
Examples
Example 1

\[
\begin{bmatrix}
\text{foo} \\
\text{bar}
\end{bmatrix} \sqcup \begin{bmatrix}
\text{foo} \\
\text{bar}
\end{bmatrix} = \begin{bmatrix}
\text{foo} \\
\text{bar}
\end{bmatrix}
\]

Compare the fundamental Fundamental Theorem for lattices.
Equivalently

\[
\begin{bmatrix}
\text{foo} & \bot \\
\text{bar} & \bot \\
\end{bmatrix} \sqcup \begin{bmatrix}
\text{foo} & 1 \\
\text{bar} & 1 \\
\end{bmatrix} = \begin{bmatrix}
\text{foo} & 1 \\
\text{bar} & 1 \\
\end{bmatrix}
\]
Example 2: Merging information

\[
\begin{bmatrix}
\text{foo} & a \\
\text{bar} & \\
\end{bmatrix} \sqcup \begin{bmatrix}
\text{foo} & \\
\text{bar} & 1 \\
\end{bmatrix} = \begin{bmatrix}
\text{foo} & a \\
\text{bar} & a \\
\end{bmatrix} = A
\]

(a) \(A(\langle \text{foo} \rangle) = a \sqcup [1] = a\)

(b) \(A(\langle \text{foo} \rangle) = A(\langle \text{bar} \rangle)\)

(c) \(A(\langle \text{bar} \rangle) = a\)  \((a), (b)\)
Some further examples

\[
\begin{align*}
&\text{to} \quad \begin{bmatrix}
\text{mo} & a \\
\text{larry} & b \\
\text{curly} & c \\
\end{bmatrix} \\
&\text{fro} \quad \begin{bmatrix}
3 \\
3 \\
\end{bmatrix} \\
\end{align*}
\]

\[
\begin{align*}
&\text{to} \quad \begin{bmatrix}
\text{mo} & a \\
\text{larry} & b \\
\text{curly} & c \\
\end{bmatrix} \\
&\text{fro} \quad \begin{bmatrix}
3 \\
3 \\
\end{bmatrix} \\
\end{align*}
\]

\[
\begin{align*}
&\text{to} \quad \begin{bmatrix}
2 \\
\text{larry} & 2 \\
\end{bmatrix} \\
&\text{fro} \quad \begin{bmatrix}
3 \\
3 \\
\end{bmatrix} \\
\end{align*}
\]

\[
\begin{align*}
&\text{to} \quad \begin{bmatrix}
2 \\
\text{larry} & 2 \\
\end{bmatrix} \\
&\text{fro} \quad \begin{bmatrix}
\text{larry} & 2 \\
\text{larry} & 2 \\
\end{bmatrix} \\
\end{align*}
\]
Unification Failure

We use ($\top$) to represent the case where information merger results in **contradictory** information.

\[
\begin{bmatrix}
\text{mo} & a \\
\text{to} & \text{larry} & b \\
\text{curly} & c \\
\end{bmatrix} \sqcup \begin{bmatrix}
\text{mo} & a \\
\text{to} & \text{larry} & b \\
\text{curly} & a \\
\end{bmatrix} = \top
\]

\[a \sqcup b = \top\]

Formally we will say: $\top$ is a member of the feature structure poset but **NOT** a feature structure (not a consistent description of a linguistic object).
Unification as Least Upper Bound
A linguistic example
**Unification**

**Definition 1.** \( D_1 \sqcup D_2 \) Let

\[
S = D_1 \sqcup D_2
\]

Then \( S \) is the following feature structure, if it exists:

(a) \( \text{Dom}(S) = \text{Dom}(D_1) \cup \text{Dom}(D_2) \)

(b) \( \text{Paths}(S) = \text{Paths}(D_1) \cup \text{Paths}(D_2) \)

(c) For all attributes \( a \in \text{Dom}(S) \)

\[
S(a) = D_1(a) \sqcup D_2(a)
\]

(d) For all paths \( p, q \in \text{Paths}(S) \)

\[
S(p) = S(q) \text{ if and only if } D_1(p) = D_1(q) \text{ or } D_2(p) = D_2(q)
\]

If \( S \) does not exist, then

\[
\top = D_1 \sqcup D_2
\]
Background for definition

(a) \( a \sqcup a = a \)

(b) \( a \sqcup 1 = a 1, \) a variable

(c) \( 1 \sqcup 2 = 1 = 2 1, 2 \) variables

(d) \( a \sqcup b = \top \) if \( a \neq b, \) a, b terminals
Summary

- The unification of two feature structures is their least upper bound with respect to the subsumption ($\sqsubseteq$) order.
- Unification always returns a value, although sometimes that value is $\top$, which is not a feature structure.
- This is also referred to as unification failure.
- Intuitively, unification failure means failure to merge the information: There is no consistent way to combine the two descriptions.