

Lexical-Functional Grammar

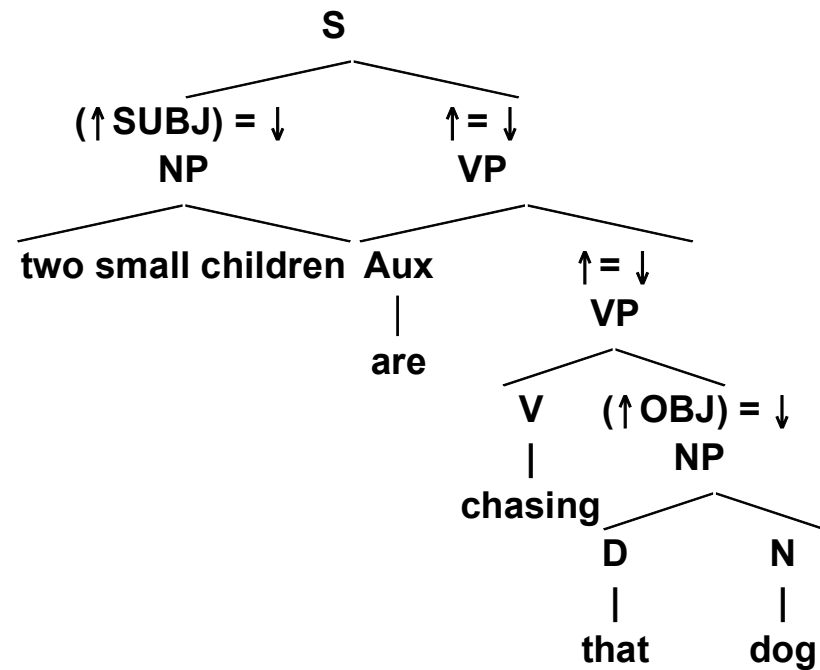
Lecture 3: Functional Structure

Stephen Wechsler

University of Texas at Austin

c-structure:

- composed of syntactic categories, terminal strings, and dominance and precedence relations between them
- models external structure of language
- generated by rules of grammar



f-structure:

- composed of grammatical function names (subject, etc.), semantic forms, and feature symbols
- models internal structure of language, which is largely invariant across languages
- derived from c-structure via algorithm of metavariable instantiation
- ungrammatical strings are often ruled out by principles applying at f-structure

SUBJ	[...'two small children...']
OBJ	[...'that dog...']
PRED	'chase <_,_>'
TENSE	present

What is an f-structure?

An *f-structure* is a set of features, normally depicted as a matrix with two columns. Each row of the matrix is a feature.

$$\begin{bmatrix} \text{PRED} & \text{'girl'} \\ \text{NUM} & \text{SG} \end{bmatrix}$$

A feature is an ordered pair: the first element of this pair is called the *attribute*, the second is called the *value*.

$$[\text{NUM} \quad \text{SG}]$$

An *f-structure* represents a *function* in the mathematical sense; namely, it is governed by the axiom of Consistency or Uniqueness:

Uniqueness: Every attribute has a unique value.

An **ill-formed** f-structure: $\left[\begin{array}{ll} \text{PRED} & \text{'girl'} \\ \text{NUM} & \text{SG} \\ \text{NUM} & \text{PL} \end{array} \right]$

attributes and *values*

Attributes range over the names of grammatical functions like SUBJ, OBJ, PRED, CASE, NUM, etc.

Values are of several types:

- simple atomic *symbols* (SG, PL, etc.)
- *semantic forms* like 'book', 'John', 'a', 'the', 'pro' (for pronouns), etc. These are normally found only as values for the attribute PRED ('PREDicate')
- *subsidiary f-structures* which are themselves sets of ordered pairs

(Also: *sets* of f-structures; these will be discussed later)

example:

$$f1 \left[\begin{array}{l} \text{PRED 'girl'} \\ \text{NUM SG} \\ \text{SPEC } f2[\text{PRED 'a'}] \end{array} \right]$$

types of value:

- *semantic form*
- *symbol*
- *f-structure*

f-descriptions are equations used to describe f-structures.

$(f1 \text{ NUM}) = \text{SG}$

$(f1 \text{ PRED}) = \text{'GIRL'}$

$(f1 \text{ SPEC}) = f2$

$(f2 \text{ PRED}) = \text{'a'}$

translations:

'Applying the function $f1$ to the attribute NUM yields the value SG.'

'Applying the function $f1$ to the attribute PRED yields the value 'girl'.'

'Applying the function $f1$ to the attribute SPEC yields the value the function $f2$.'

$(f1 \text{ NUM}) = \text{SG}$
 $(f1 \text{ PRED}) = \text{'GIRL'}$
 $(f1 \text{ SPEC}) = f2$
 $(f2 \text{ PRED}) = \text{'a'}$

Substituting $(f1 \text{ SPEC})$ for $f2$ in the last equation, we get:

$((f1 \text{ SPEC}) \text{ PRED}) = \text{'a'}$

Convention: for 'left-branching' equations like this, we can omit the inner parentheses and write:

$(f1 \text{ SPEC PRED}) = \text{'a'}$

$(f1 \text{ SPEC PRED}) = \text{'a'}$

The sequence of attributes **SPEC PRED** describes a *path* through the f-structure.

$$f1 \left[\begin{array}{l} \text{PRED 'girl'} \\ \text{NUM SG} \\ \text{SPEC } f2[\text{PRED 'a'}] \end{array} \right]$$

Deriving f-structure from c-structure

schemata: The up arrow \uparrow and down arrow \downarrow are *metavariables* over functions. They are pronounced 'up' and 'down'. Equations containing them are called *schemata*.

An f-structure is associated with each c-structure node. The projection from c- to f-structure occurs via schemata, as follows:

\uparrow = the f-structure associated with this node's mother
 \downarrow = the f-structure associated with this node

schemata are introduced into c-structures (but do not appear in f-structures) in two ways:

(i) annotations on phrase structure rules:

S → NP VP
 (↑SUBJ) = ↓ ↑ = ↓

NP → N
 ↑ = ↓

VP → V
 ↑ = ↓

(ii) lexical entries:

John: N, (\uparrow PRED) = 'John'
(\uparrow NUM) = SG
(\uparrow PERS) = 3

they: N, (\uparrow PRED) = 'pro'
(\uparrow NUM) = PL
(\uparrow PERS) = 3

smokes: V, (\uparrow PRED) = 'smoke<(\uparrow SUBJ)>'
(\uparrow TENSE) = PRES
(\uparrow SUBJ NUM) = SG
(\uparrow SUBJ PERS) = 3

Algorithm for deriving f-structure from c-structure

1. Assign an arbitrary f-structure name (f_1, f_2, \dots) to each node, including phrasal, terminal and preterminal nodes.

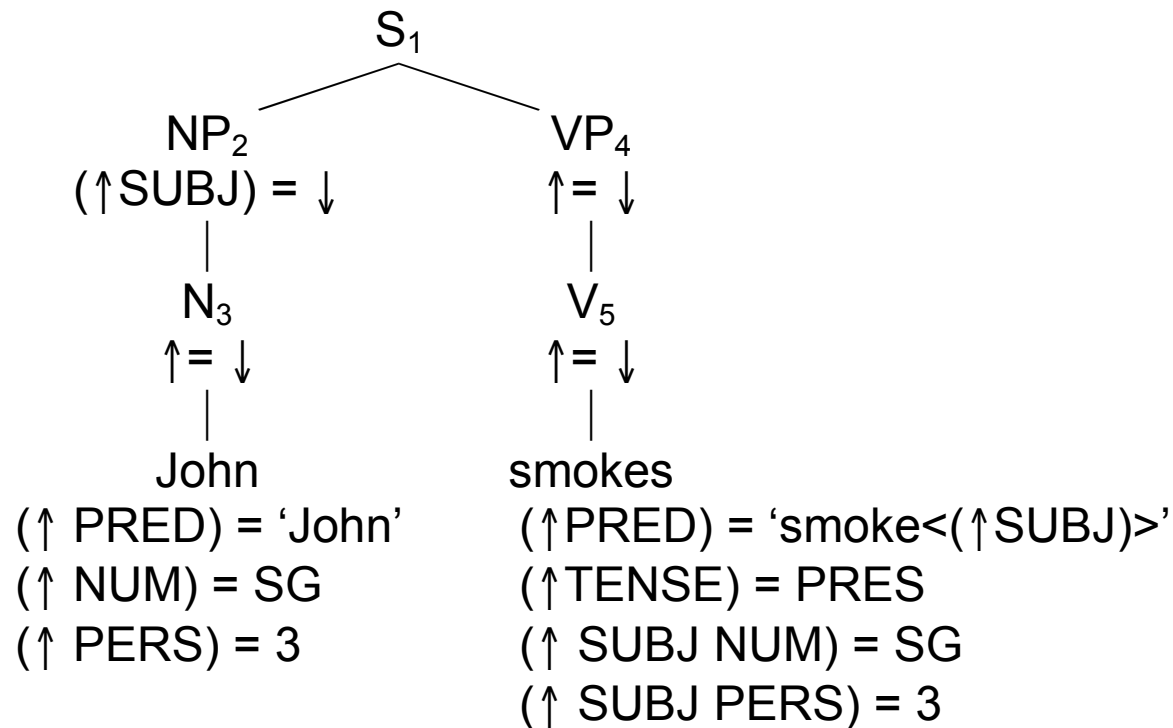
2. Collect equations from all nodes; substitute f-structure names (f_1, f_2, \dots) for the metavariables \uparrow and \downarrow as follows:

\uparrow = the f-structure associated with my mother

\downarrow = the f-structure associated with me

This set of equations is called the *f-description*.

3. Solve for the f-structure of the root node. The solution is the minimal (i.e. least informative) f-structure that satisfies all the equations in the f-description.

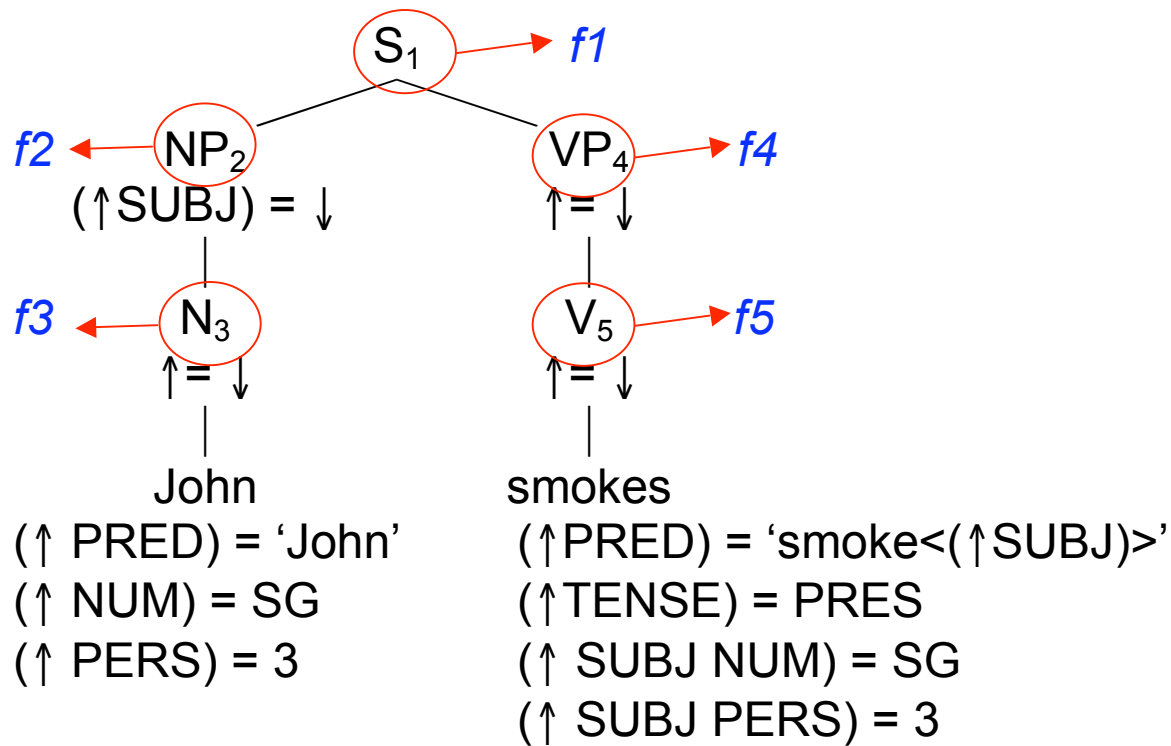


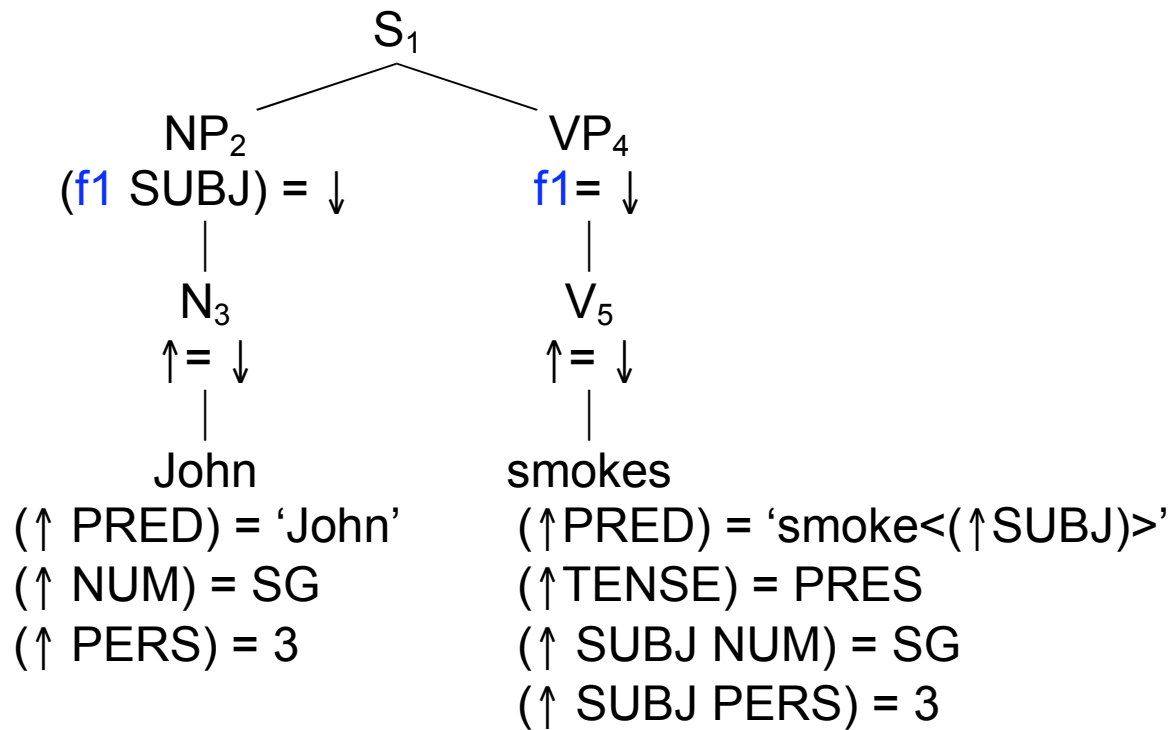
For convenience the f-structure for each node will use that node's numerical subscript. Hence, where ϕ is the projection function from c- to f-structure:

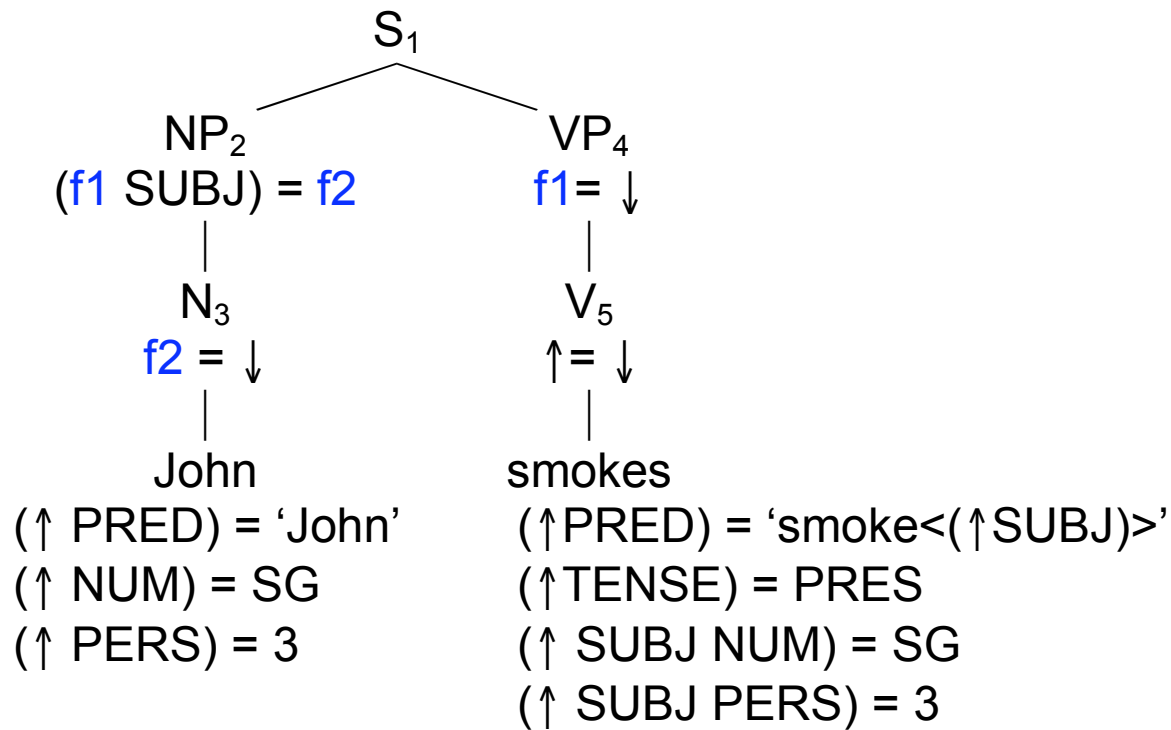
$$\phi(S_1) = f_1$$

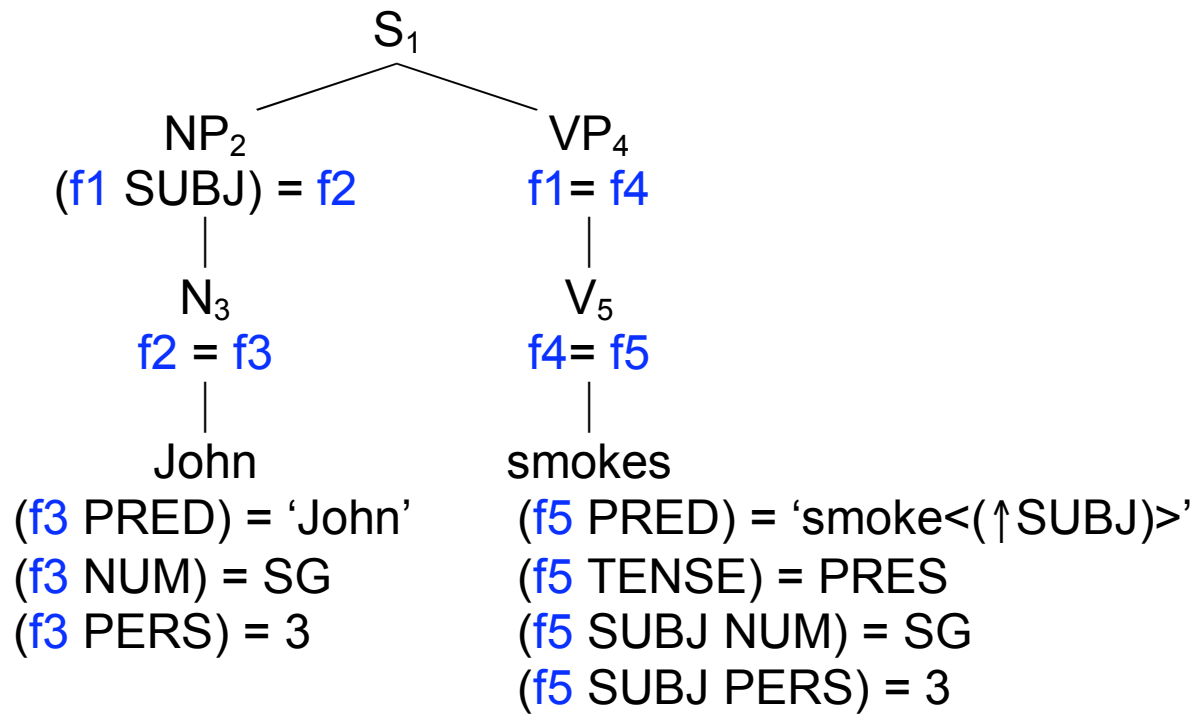
$$\phi(NP_2) = f_2$$

...etc.









Now collect the equations into the *f-description*:

(f3 PRED) = 'John'	(f5 PRED) = 'smoke<(↑SUBJ)>'
(f3 NUM) = SG	(f5 TENSE) = PRES
(f3 PERS) = 3	(f5 SUBJ NUM) = SG
(f1 SUBJ) = f2	(f5 SUBJ PERS) = 3
f2 = f3	f1 = f4
	f4 = f5

...and solve for f1— the f-structure for the root node S₁.

(f3 PRED) = 'John' (f5 PRED) = 'smoke<(↑SUBJ)>'
(f3 NUM) = SG (f5 TENSE) = PRES
(f3 PERS) = 3 (f5 SUBJ NUM) = SG
(f1 SUBJ) = f2 (f5 SUBJ PERS) = 3
f2 = f3 f1 = f4
 f4 = f5

Draw a set of brackets and label them f1:



(f3 PRED) = 'John'

(f3 NUM) = SG

(f3 PERS) = 3

(f1 SUBJ) = f2

f2 = f3

(f5 PRED) = 'smoke<(↑SUBJ)>'

(f5 TENSE) = PRES

(f5 SUBJ NUM) = SG

(f5 SUBJ PERS) = 3

f1 = f4

f4 = f5



(f3 PRED) = 'John'

(f3 NUM) = SG

(f3 PERS) = 3

(f1 SUBJ) = f2

f2 = f3

(f5 PRED) = 'smoke<(↑SUBJ)>'

(f5 TENSE) = PRES

(f5 SUBJ NUM) = SG

(f5 SUBJ PERS) = 3

f1 = f4

f4 = f5



(f3 PRED) = 'John'

(f3 NUM) = SG

(f3 PERS) = 3

(f1 SUBJ) = f3

f2 = f3

(f1 PRED) = 'smoke<(↑SUBJ)>'

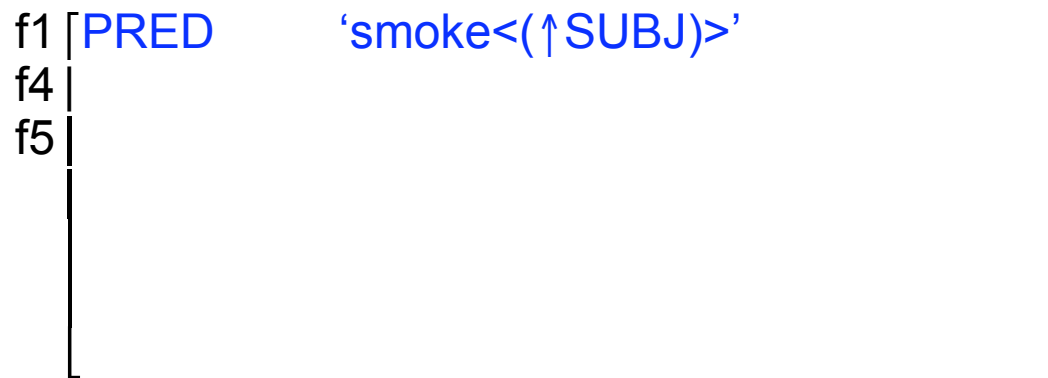
(f1 TENSE) = PRES

(f1 SUBJ NUM) = SG

(f1 SUBJ PERS) = 3

~~f1 = f4~~

~~f4 = f5~~



(f3 PRED) = 'John'

(f3 NUM) = SG

(f3 PERS) = 3

(f1 SUBJ) = f3

f2 = f3

~~(f1 PRED) = 'smoke<(↑SUBJ)>'~~

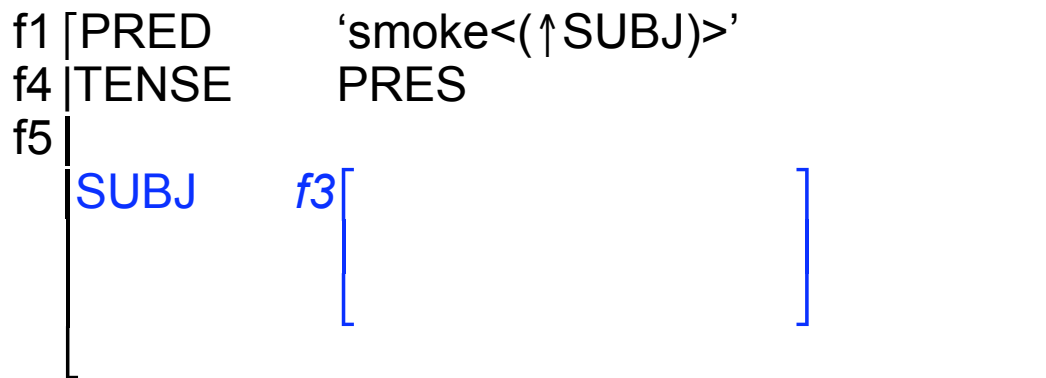
~~(f1 TENSE) = PRES~~

(f1 SUBJ NUM) = SG

(f1 SUBJ PERS) = 3

~~f1 = f4~~

~~f4 = f5~~



(f3 PRED) = 'John'

(f3 NUM) = SG

(f3 PERS) = 3

~~(f1 SUBJ) = f3~~

f2 = f3

~~(f1 PRED) = 'smoke<(↑SUBJ)>'~~

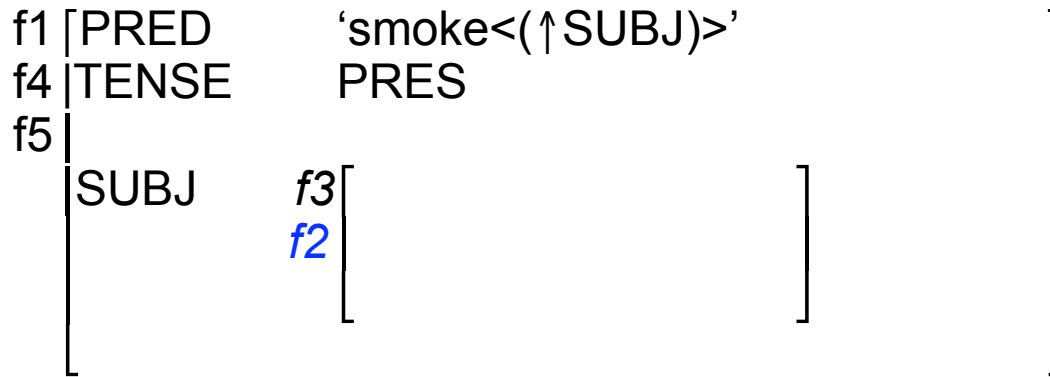
~~(f1 TENSE) = PRES~~

(f1 SUBJ NUM) = SG

(f1 SUBJ PERS) = 3

~~f1 = f4~~

~~f4 = f5~~



(f3 PRED) = 'John'

(f3 NUM) = SG

(f3 PERS) = 3

~~(f1 SUBJ) = f3~~

~~f2 = f3~~

~~(f1 PRED) = 'smoke<(↑SUBJ)>'~~

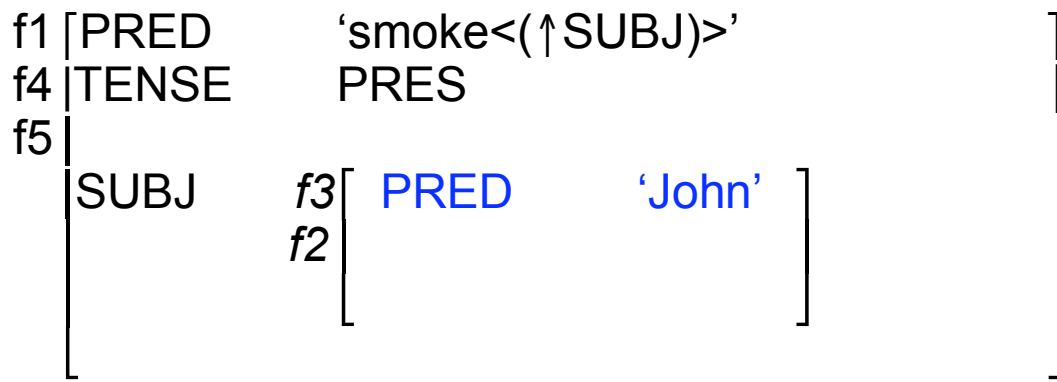
~~(f1 TENSE) = PRES~~

(f1 SUBJ NUM) = SG

(f1 SUBJ PERS) = 3

~~f1 = f4~~

~~f4 = f5~~



~~(f3 PRED) = 'John'~~

(f3 NUM) = SG

(f3 PERS) = 3

~~(f1 SUBJ) = f3~~

~~f2 = f3~~

~~(f1 PRED) = 'smoke<(↑SUBJ)>'~~

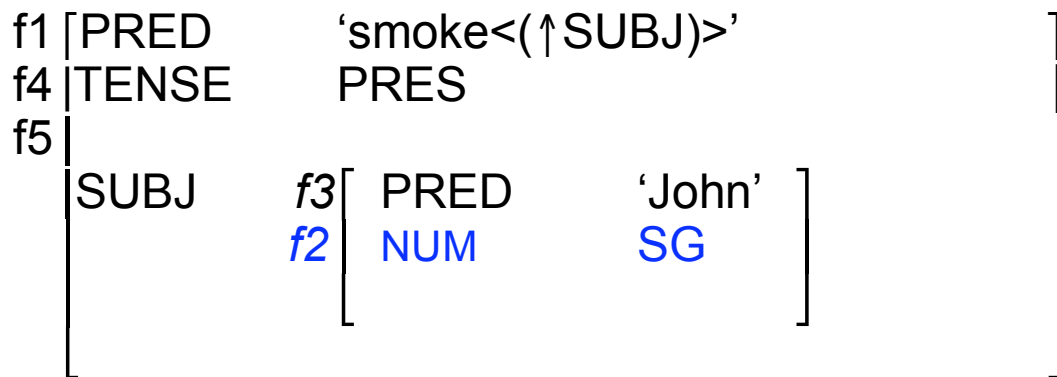
~~(f1 TENSE) = PRES~~

(f1 SUBJ NUM) = SG

(f1 SUBJ PERS) = 3

~~f1 = f4~~

~~f4 = f5~~



~~(f3 PRED) = 'John'~~

~~(f3 NUM) = SG~~

(f3 PERS) = 3

~~(f1 SUBJ) = f3~~

~~f2 = f3~~

~~(f1 PRED) = 'smoke<(↑SUBJ)>'~~

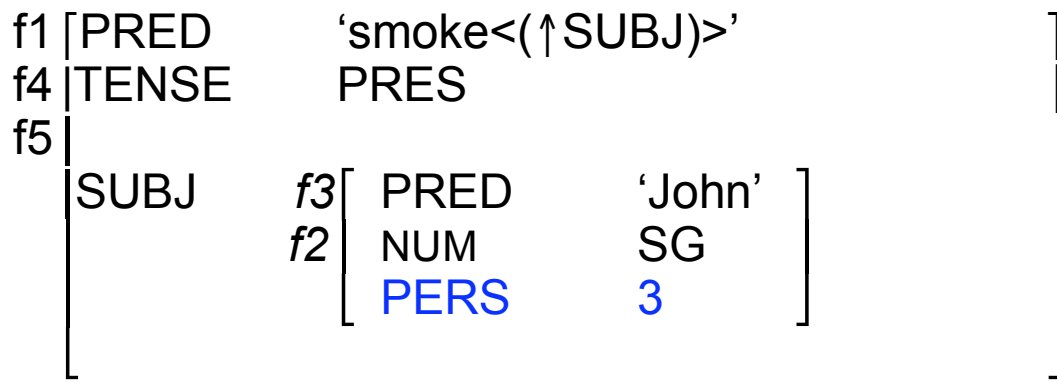
~~(f1 TENSE) = PRES~~

(f1 SUBJ NUM) = SG

(f1 SUBJ PERS) = 3

~~f1 = f4~~

~~f4 = f5~~



(f3 PRED) = 'John'	(f1 PRED) = 'smoke<(↑SUBJ)>'
(f3 NUM) = SG	(f1 TENSE) = PRES
(f3 PERS) = 3	(f1 SUBJ NUM) = SG
(f1 SUBJ) = f3	(f1 SUBJ PERS) = 3
f2 = f3	f1 = f4
	f4 = f5

f1	[PRED		'smoke<(↑SUBJ)>']		
f4		TENSE		PRES				
f5								
	[SUBJ	f3	[PRED		'John']
			f2	[NUM		SG]
				[PERS		3]
]

This equation is redundant.

What if it had been **(f1 SUBJ NUM) = PL** instead?

~~(f3 PRED) = 'John'~~

~~(f3 NUM) = SG~~

~~(f3 PERS) = 3~~

~~(f1 SUBJ) = f3~~

~~f2 = f3~~

~~(f1 PRED) = 'smoke<(↑SUBJ)>'~~

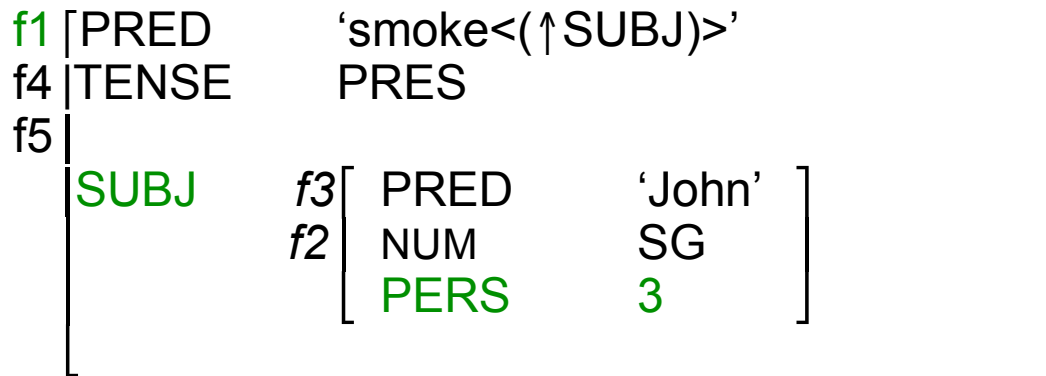
~~(f1 TENSE) = PRES~~

~~(f1 SUBJ NUM) = SG~~

(f1 SUBJ PERS) = 3

~~f1 = f4~~

~~f4 = f5~~



~~(f3 PRED) = 'John'~~

~~(f3 NUM) = SG~~

~~(f3 PERS) = 3~~

~~(f1 SUBJ) = f3~~

~~f2 = f3~~

~~(f1 PRED) = 'smoke<(↑SUBJ)>'~~

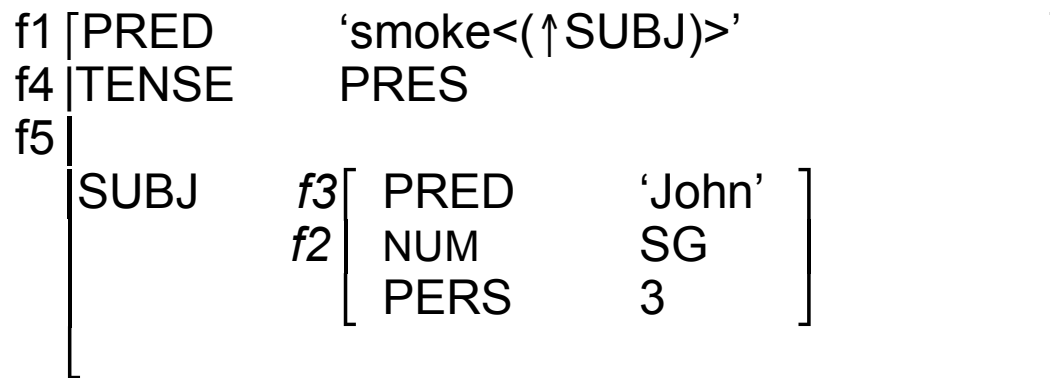
~~(f1 TENSE) = PRES~~

~~(f1 SUBJ NUM) = SG~~

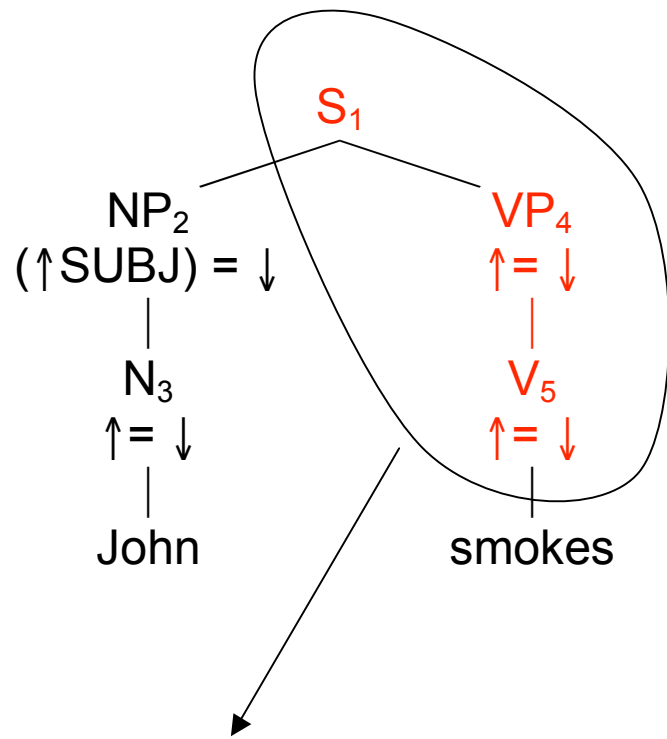
~~(f1 SUBJ PERS) = 3~~

~~f1 = f4~~

~~f4 = f5~~



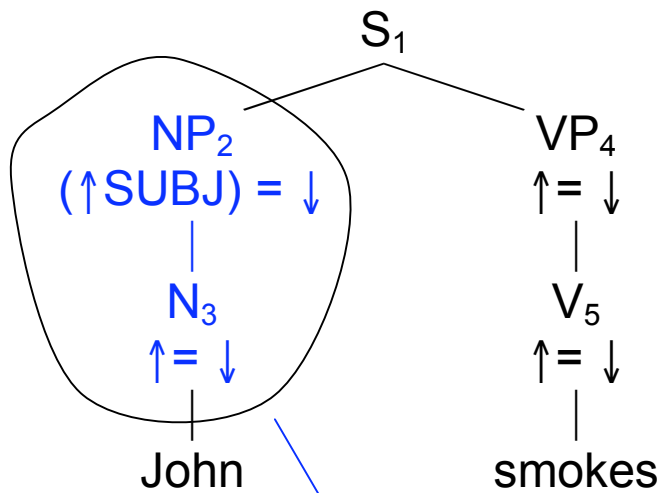
Voilà! We're done.



$f1 (= f4 = f5)$

$\uparrow = \downarrow$ annotations: these nodes all map to the same f-structure.

'non-compositionality': f-structure for VP contains SUBJ info.



f1	[PRED		'smoke<(↑SUBJ)>']
f4		TENSE		PRES	
f5					
		SUBJ			
		f3	[PRED	'John'
		f2	[NUM	SG
			[PERS	3
]]

Ungrammaticality as feature clash (violation of Uniqueness)

1. John smokes.
2. *They smokes.

they: N, (\uparrow PRED) = 'pro'
(\uparrow NUM) = PL
(\uparrow PERS) = 3

smokes: V, (\uparrow PRED) = 'smoke<(\uparrow SUBJ)>'
(\uparrow TENSE) = PRES
(\uparrow SUBJ NUM) = SG
(\uparrow SUBJ PERS) = 3

