Exploring the Feature Space

Ash Asudeh
ICS & SLALS, Carleton University

Bilingual Workshop in Theoretical Linguistics 12
University of Ottawa
December 5, 2008

Introduction

- Features play an important role in many current syntactic theories, but especially in constraint-based syntactic theories, which have precisely articulated feature theories.
- In the first, shorter part of the talk, I consider some general aspects of syntactic features, attempting to tie certain aspects of Minimalist features to constraint-based features.
- In the second, longer part, I present a novel feature-based analysis (in the sense of constraint-based syntax) of Comp-Trace Effects as a constraint at the syntax-phonology interface.

Features in Syntactic Theory

Honest Accounting

If we have any general methodological message in this book, it is to urge honest accounting.
(Culicover & Jackendoff 2005: 50)

[Culicover and Jackendoff propose that] the evaluation and comparison of analyses should be guided by a principle of 'honest accounting' that counts global as well as local consequences of analytic choices.
(Blevins 2008: 730)

A principle of honest accounting would dictate that any benefits obtained by restricting the X-bar conventions to word-class features should be balanced against the cost of reclassifying entire inventories of morphosyntactic properties as word-class features.
(Blevins 2008: 731)

Features and Explanation

- The sorts of features that are associated with functional heads in the Minimalist Program are well-motivated morphosyntactically, although other theories may not draw the conclusion that this merits phrase structural representation (cf. Blevins's comments).
- Care must be taken to avoid circular reasoning in feature theory:
  - The 'strong' meta-feature: "This thing has whatever property makes things displace, as evidenced by its displacement."
  - The 'weak' meta-feature: "This thing lacks whatever property makes things displace, as evidenced by its lack of displacement."
  - The EPP feature: "This thing has whatever property makes things move to subject position, as evidenced by its occupying subject position."

Features and Simplicity

- Adger (2003, 2008) considers three kinds of basic features:
  - Privative, e.g. [singular]
  - Binary, e.g. [singular +]
  - Valued, e.g. [number   singular]
- Adger considers the privative kind the simplest in its own right.
- This may be true, but only if it does not introduce complexity elsewhere in the system ('honest accounting').
- Notice that only the final type of feature treats number features as any kind of natural class within the theory (as opposed to meta-theoretically).

Feature-Value Unrestrictiveness & Free Valuation

- Asudeh & Toivonen (2006) argue that the Minimalist feature system of Adger (2003) has two undesirable properties.

  Feature-value unrestrictiveness
  Feature valuation is unrestricted with respect to what values a valued feature may receive.

  Free valuation
  Feature valuation appears freely, subject to locality conditions.

- This results in a very unconstrained theory of features.
- This may sound good, because it's less stipulative and hence more Minimal, but from a theory perspective it is bad: unconstrained theories are less predictive.

Two Contrasting Feature Theories

- HPSG (Pollard & Sag 1994): features are not just valued, the values are also *typed*
  - If two values can unify, they must be in a typing relation (one must be a subtype of the other).
  - Feature values in HPSG are thus tightly restricted by types.
- LFG (Kaplan & Bresnan 1982, Bresnan 2001): features are not restricted, but there is no free valuation
  - A feature cannot end up with a given value unless there is an explicit equation in the system.

Feature Simplicity and Constraint Types

- LFG offers the opportunity to consider Adger's three feature types in light of a single feature type, with varying constraint types.
  - LFG features are valued (*f* is an LFG f(unctional)-structure):

$$f\begin{bmatrix}\text{NUMBER} & \text{singular}\end{bmatrix}$$

  - Types of LFG feature constraints.
    - Defining equation: $(f\ \text{NUMBER}) = \text{singular}$
    - Existential constraint: $(f\ \text{NUMBER})$
    - Negative existential constraint: $\neg(f\ \text{NUMBER})$
    - Constraining equation: $(f\ \text{NUMBER}) =_c \text{singular}$
    - Negative constraining equation: $(f\ \text{NUMBER}) \neq \text{singular}$

Feature Simplicity and Constraint Types

- All features treated as valued features: no restriction on constraint types
- All features treated as binary features: only positive and negative constraining equations allowed
- All features treated as privative: only negative and existential constraints allowed
  - This understanding of privative features actually does treat number as a natural class.
- This treats the notion of feature simplicity as a kind of meta-theoretical statement in an explicit, non-ad-hoc feature theory.

Syntactic Features and the Comp-Trace Effect at the Syntax-Phonology Interface

Introduction

- In various languages, including English, an unbounded dependency (‘*wh*-movement’) cannot be formed on the subject of a finite clause only if the clause is introduced by a complementizer:

  (1) Who do you think sneezed?

  (2) * Who do you think that sneezed?

- These effects are commonly referred to as ‘That-Trace’ Effects, or more generally, ‘Comp-Trace’ Effects.
- This nomenclature derives from transformational analyses that seek to explain the contrast based on the ungrammaticality of a trace of movement immediately following a complementizer.
- I’ll use the theory-neutral descriptive term ‘complementizer-adjacent nominal extraction’ (CANE).

Introduction

- There have been many attempts in the transformational literature to address this phenomenon, including: Perlmutter (1968,1971), Langendoen (1970), Bresnan (1972), Chomsky & Lasnik (1977), Kayne (1981), Pesetsky (1982), Koopman (1983), Sobin (1987,2002), Rizzi (1990,1997), Culicover (1991a,b,1992,1993), Browning (1996), Roussou (2002), Ishii (2004), among others.
- There have also been various attempts in the non-transformational, constraint-based literature to address the phenomenon, notably: Gazdar (1981), Pollard & Sag (1994), Bouma, Malouf & Sag (2001), Falk (2000, 2001, 2002).

Introduction

- In this part of the talk, I want to offer a new lexicalist, constraint-based account of CANE Effects, including certain quite tricky subtleties that have previously proven difficult to explain.
- The account is cast in the framework of Lexical-Functional Grammar (LFG; Kaplan & Bresnan 1982, Bresnan 2001, Dalrymple 2001).
- I will show that once we assume the Correspondence Architecture of LFG, CANE Effects can be explained without introducing any theoretical machinery that is not a priori available or necessary, while maintaining robust empirical coverage.

Outline

1. Background
   a. Data
   b. Previous approaches
2. Brief overview of relevant aspects of LFG
   a. Architecture of LFG
   b. Interrogatives and relative clauses in LFG
   c. Inverse Correspondences
3. A new analysis of CANE Effects (a.k.a. Comp-Trace Effects)

Background

Data and Generalizations

Data: CANE Effects

(1) Who did Kim say __ saw Sandy?

(2) * Who did Kim say that __ saw Sandy?

(3) Who did Kim say that Sandy saw __?

(4) * Who did Kim wonder __ saw Sandy?

(5) ? Who did Kim wonder whether/if Sandy saw __?

(6) * Who did Kim wonder whether/if __ saw Sandy?

Data: Adverb Effect

(1) * Who did Kim say that __ eats meat?

(2) Who did Kim say that just yesterday __ ate meat?

(3) Who did Kim say that under certain circumstances __ would eat meat?

(4) Who did Kim say that under no circumstances __ would eat meat?

(5) Who did Kim say just yesterday __ ate meat.

Note: Sentences like (5) are sometimes reported as ungrammatical (Rizzi 1997), but systematic questionnaire studies do not support this contention (Sobin 2002).

(6) * Who did Kim wonder whether/if __ eats meat?.

(7) ? Who did Kim wonder whether/if just yesterday __ ate meat?

(8) ? Who did Kim wonder whether/if under certain circumstances __ would eat meat?

Data: Relative Clause Paradox

(1) Who did Kim say __ saw Sandy?

(2) * Who did Kim say that __ saw Sandy?

(3) Who did Kim say that Sandy saw __?

(4) * The person __ saw Sandy is Robin.

(5) The person that __ saw Sandy is Robin.

(6) The person that Sandy saw __ is Robin.

(7) The person Sandy saw __ is Robin.

Note: Sentences like (4) are reported as grammatical in some dialects, including varieties of British English (Sobin 2002) and African American Vernacular English (Chomsky & Lasnik 1977, Pesetsky 1982).

Generalizations

1. Subject extraction after a complementizer, e.g. *that*, leads to degraded grammaticality, over and above other possible sources of degraded grammaticality (cf. *whether* examples).

2. The ungrammaticality of CANE is alleviated if a sentential adverbial intervenes between the complementizer and subject extraction site.

3. Paradoxically, in relative clause subject extraction, *that* is obligatory.

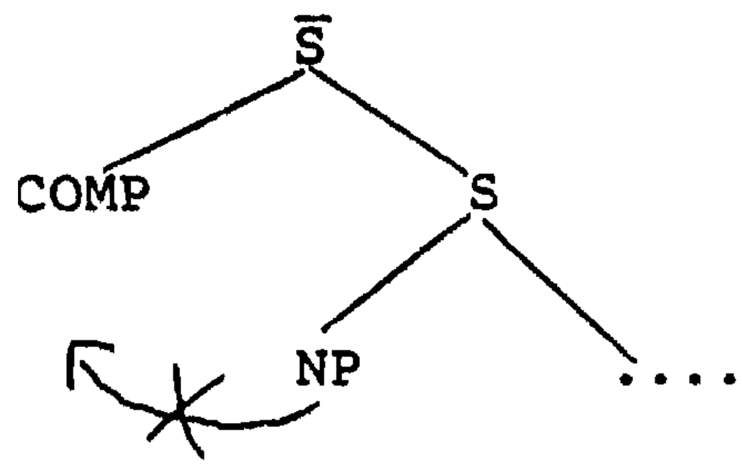
Background

Previous Approaches

Fixed Subject Constraint

- Bresnan (1972):

  Fixed Subject Constraint
  No NP can be crossed over an adjacent complementizer:



- This is the preliminary version of the constraint, which is subsequently revised as a constraint on deletion, based on facts from comparative deletion.

- **Note**: Predicts the Adverb Effect!

Problems with the Fixed Subject Condition

- Theoretical Problems:
  - Constraints on transformations were abandoned in transformational grammar of the Government and Binding variety. In more recent work (Minimalist Program), constraints on transformation must be completely general, not specific to certain movements, etc. Anything specific must fall out of general constraints.
- Empirical Problems:
  - The FSC does not predict the Relative Clause Paradox: relative clause subject extraction is predicted to be ungrammatical unless the relativizer *that* is not a COMP.

Surface Filters

- Chomsky & Lasnik (1977):
  (Surface) filters restrict the transformational component by marking as ungrammatical a subset of the set of outputs.

- Their filter for that-trace (C&L, 1977: 451):

  (1) $*[_{\overline{S}}$ that $[_{NP}$ e$]\ldots]$, unless $\overline{S}$ or its trace is in the context: $[_{NP}$ NP $\_\_\ \ldots]$

- **Note**: Predicts the Adverb Effect (Culicover 1993)!

- Part of the motivation of the filter is that it entails the following universal (based on observations in Perlmutter 1968,1971):

  (2) The filter (1) is valid for all languages that do not have a rule of Subject-Pronoun Deletion, and only these.

Subject-Pronoun Deletion Universal

Subject-Pronoun Deletion Universal
The That-Trace Filter is valid for all languages that do not have a rule of Subject-Pronoun Deletion, and only these.

(1) ¿Quién creiste que vio a Juan? — Spanish
'Who do you believe that saw Juan?'

(2) * Qui crois-tu qu'a vu Jean? — French
'Who do you believe that saw Juan?'

- Relevant aspect of the derivation of (1):

  quién tú creiste que [$_{NP}$ e] vio a Juan → [Deletion]

  quién ~~tú~~ creiste que ~~[$_{NP}$ e]~~ vio a Juan

- Crucial: [$_{NP}$ e] ≢ ~~[$_{NP}$ e]~~

Problems with Surface Filters

- Theoretical problems:
  - Potentially computational expensive: why (over)generate a structure that is known to be ungrammatical?
  - Stipulative, ad hoc exception (‘unless’ clause) necessary to allow *that* in relative clauses
  - Implausible under current transformational assumptions (Minimalism): the that-trace structure would have to be generated for a reason, but then removed from consideration; adds opacity.
  - The Subject-Pronoun Deletion universal rests on having multiple kinds of ‘emptiness’ in the theory.

Problems with Surface Filters

- Empirical problems:
  - An additional, stipulative filter required to capture the Relative Clause Paradox.
  - The subject-pronoun deletion universal is not a universal (in particular, certain non-null-subject Scandinavian languages and dialects allow That-Trace violations). Since the That-Trace Filter entails it (by design), the filter cannot be correct.

Gazdar’s GPSG Metarule Analysis

- Gazdar (1981) proposes a GPSG metarule for subject extractions.
    - In other word, subject extraction in general works differently than other forms of extraction.

$$[_{\alpha}\ \mathrm{X}\ \ \underset{[-\mathrm{C}]}{\Sigma}\ \ /\mathrm{NP} \ldots] \Rightarrow [_{\alpha}\ \mathrm{X}\ \ \underset{[+\mathrm{FIN}]}{\mathrm{VP}}\ \ /\mathrm{NP} \ldots]$$

where X contains at least one major category symbol, where α is anything, and where Σ ranges over sentential categories.

- The statement that X must contain at least one major category symbol excludes That-Trace, because the S-bar rule directly introduces *that*, so:

$$* \ [_{\bar{\mathrm{S}}/\mathrm{NP}}\ \textit{that}\ \ \underset{[+\mathrm{FIN}]}{\mathrm{VP}}\ ]$$

- Problem: Does not capture the Adverb Effect (equally ruled out, due to the inapplicability of the metarule)

HPSG’s Trace Principle

- Pollard and Sag (1994: 173-174) essentially adopt and update Gazdar’s (1981) proposal.
- They posit the following principle:

  Trace Principle (parametrized for English)
  Every trace must be *strictly* subcategorized by a substantive head.
- This essentially entails that subjects are not extracted like other arguments in English and commits them, like Gazdar, to an extra condition to capture subject condition.
- In this case the relevant mechanism is a lexical rule called the Subject Extraction Lexical Rule, which crucially applies only to type *unmarked* clauses, where clauses introduced by *that* have the type *marked.*
- Problem: Does not capture the Adverb Effect

ECP Approaches

- **Empty Category Principle** (Chomsky 1981)
  A nonpronominal empty category must be properly governed.
- The intervening complementizer in Comp-Trace configurations interferes in proper government of the trace (Kayne 1981, Pesetsky 1982, Koopman 1983, Lasnik and Saito 1984, Rizzi 1990, among others).
- Problems:
  - Does not capture the Adverb Effect, since addition of extra material cannot make a positive difference to the relations involved.
  - Does not resolve the Relative Clause Paradox, unless stipulations are made about the relativizer *that*.
    - The stipulations also suffer from general and theory-internal problems, as well as various empirical failings.

Culicover's Polarity Phrase Approach

- Culicover (1991a,b) argues for a functional projection PolP between CP and IP, based on the Adverb Effect.
- For a sentence like (1), Culicover proposes that the adverbial *for all intents and purposes* is adjoined to PolP.

(1) Robin met the man Leslie said that for all intents and purposes was the mayor of the city.

- Culicover argues that an empty Pol head (which otherwise hosts the modal in negative inversion) licenses the subject trace (the structure below is from Browning 1996):

$\ldots$the man [$_{\mathrm{CP}}$ OP$_i$ [$_{\mathrm{IP}}$ Leslie said [$_{\mathrm{CP}}$ t$_i''$ [$_{\mathrm{C'}}$ that [$_{\mathrm{POLP}}$ Adv [$_{\mathrm{POLP}}$ t$_i'$ [$_{\mathrm{Pol'}}$ e$_i$ [$_{\mathrm{IP}}$ t$_i$ ...

Culicover's Polarity Phrase Approach

- Problems:
  - Nothing prevents the empty Pol head from appearing without an adverbial, so the account really predicts no Comp-Trace Effect at all.
  - In sentences involving a negative adverbial, such as (1), the auxiliary would have to occupy Pol (since hosting auxiliaries in negative inversion is the motivation for the head). This wrongly predicts that such examples are ungrammatical, since the movement of the auxiliary results in the subject trace being ungoverned/unlicensed (Culicover 1993).

    (1) Leslie is the person who I said that under no circumsances would run for president.

  - Does not resolve the Relative Clause Paradox.

CP Recursion

- Browning (1996) proposes that the Adverb Effect obtains because the adverbial is in SpecCP, which forces 'CP Recursion', i.e. creation of another CP layer.
- She assumes, following Cheng (1991) and Watanabe (1992), that clauses are 'typed' such that non-*wh*-clauses cannot have a SpecCP.
- Thus, if the following structure is to be the complement to a verb such as *say* or *think*, something must happen to vacate the SpecCP.

$[_{\text{CP}}$ for all intents and purposes $[_{\text{C}'}$ that $[_{\text{IP}}$ Op$_i$ was the mayor ...

CP Recursion

- Given the clause-typing assumption mentioned above, something must happen to vacate SpecCP in order for the CP to be the complement of *say*, *think*, etc.

- The complementizer moves, targeting its own CP:

  $[_{CP}$ $[_{C'}$ that$_C$ $[_{CP}$ for all intents and purposes $[_{C'}$ t_C $[_{IP}$ Op$_i$ was the mayor ...

- Subsequently the relative operator moves, yielding:

  Op$_i$ $\ldots[_{CP}$ t'_i $[_{C'}$ that$_C$ $[_{CP}$ for all intents and purposes $[_{C'}$ $t_{C/i}$ $[_{IP}$ t_i was the mayor $\ldots$

CP Recursion

- Problems:
  - It is crucial for Browning that the adverb in question be in SpecCP, but this is problematic from a theory-internal perspective.
    - Browning states that she argues for this position (1996: 241), but as far as I can tell, she just assumes it.
  - It is crucial for Browning that the complementizer not have an index (hence the subscripted c), but it is also crucial that the trace of the complementizer govern the subject trace. This basically seems contradictory. Furthermore, in other cases it seems that the complementizer should have a (real) index according to the assumptions of the theory in question (Sobin 2002).
    - Related to this: the theory does not account for the Relative Clause Paradox
  - It is not clear why the complementizer must move rather than the structure just being ruled out. The theory provides no a priori baseline for this kind of decision.

Fuse

- In contrast to attempts by, e.g., Browning (1996) and Rizzi (1997) to resolve the contradiction between transformational accounts of That-Trace Effects and Adverb Effects through *expansion* of the CP layer, Sobin (1987, 2002) argues based on this data for a *collapsing* or *thinning* of CP (cf. also Pesetsky 1982).
- Following a proposal by Carnie (2000) based on other phenomena, Sobin (2002) proposes that, under relevant conditions, the Spec and head elements of CP can collapse into a single indexed head ('Fuse').
- Crucially, the adverbs involved in Adverb Effects fuse with the complementizer, through adjunction, creating an articulated structure that has a lexical category, C.

(a) Who did you say, that without a doubt, would hate the soup?
(b) … $[_{CP}$ t_i' $[_{C'}$ $[_{C}$ $[_{C}$ that] AvP] $[_{IP}$ t_i … →
(c) … $[_{CP}$ $[_{Ci}$ t_i' $[_{C}$ $[_{C}$ that] AvP]] $[_{IP}$ t_i …

Fuse

- Sobin requires two versions of Fuse, one for chain heads and for traces (non-chain-heads):

  Fuse a Chain head
  A Chain head (in SpecCP) may collapse with C if one of these elements (SpecCP or C) is overt (that is, phonetic).

  Fuse a trace (*a non-chain head*)
  A trace (in SpecCP) may collapse with C if neither of these elements (SpecCP or C) is overt (that is, phonetic).

- The first of these deals with relative clauses and the second deals with Comp-Trace Effects.

- Together they deal with the Adverb Effect.

Fuse

(35) (a) the person who ordered the anchovies …
(b) … $[_{CP}$ who_i $[_{C'}$ $[_C$ −WH] $[_{IP}$ t_i … →
(c) … $[_{CP}$ $[_C$ $who]_i$ $[_{IP}$ t_i …

(36) (a) the person that ordered the anchovies …
(b) … $[_{CP}$ $Ø_i$ $[_{C'}$ $[_C$ that] $[_{IP}$ t_i … →
(c) … $[_{CP}$ $[_C$ $that]_i$ $[_{IP}$ t_i …

(37) (a) *the person ordered the anchovies …
(b) … $[_{CP}$ $Ø_i$ $[_{C'}$ $[_C$ −WH] $[_{IP}$ t_i … ↛

(38) (a) the person who Mary saw …
(b) … $[_{CP}$ who_i $[_{C'}$ $[_C$ −WH] $[_{IP}$ Mary … →
(c) … $[_{CP}$ $[_C$ $who]_i$ $[_{IP}$ Mary …

(39) (a) the person that Mary saw …
(b) … $[_{CP}$ $Ø_i$ $[_{C'}$ $[_C$ that] $[_{IP}$ Mary … →
(c) … $[_{CP}$ $[_C$ $that]_i$ $[_{IP}$ Mary …

(40) (a) the person Mary saw …
(b) … $[_{CP}$ $Ø_i$ $[_{C'}$ $[_C$ −WH] $[_{IP}$ Mary … ↛

Fuse

(47) (a) Who did you say would hate the soup?
(b) $\text{Who}_i \ldots \text{say} [_{CP}\ t_i{}' [_{C'} [_{C}\ -\text{WH}] [_{IP}\ t_i \ldots \rightarrow$
(c) $\text{Who}_i \ldots \text{say} [_{CP} [_{C}\ -\text{WH}]_i [_{IP}\ t_i \ldots$

(48) (a) %Who did you say that would hate the soup?
(b) $\text{Who}_i \ldots \text{say} [_{CP}\ t_i{}' [_{C'} [_{C}\ \text{that}] [_{IP}\ t_i \ldots \not\rightarrow$

Fuse

- Sobin's account reconciles all three phenomena (Comp-Trace Effect, Adverb Effect, Relative Clause Paradox).
- Problems:
    - Needs to postulate multiple kinds of *that*
        - No evidence from variation
        - *That* is implausible as a subject place-holder or relative pronoun which 'refers' (Sobin 2002: 546) to the nominal head modified by the relative clause (let's be generous and allow 'refers to' to go proxy for 'is bound by').

          (1) There is nobody that believes the claim.

          (2) Nobody$_i$ said that he$_i$ / *that$_i$ believes the claim.

          (3) Nobody$_i$ is such that he$_i$ / *that$_i$ believes the claim.
        - If there is a relativizer 'that' and a complementizer 'that', how do we prevent:

          (4) * This is the person that that ate the soup.

Fuse

- Problems:
  - It is necessary on Sobin's account that *who*, a +WH element, be allowed to fuse with a -WH element (cf. his (35–40) above).
  - In order for the Adverb Effect to be captured by Fuse, it is necessary for Sobin to assume that the C created by adjunction of AdvP to *that* counts as null. Why should addition of overt structure make an element null?
  - Furthermore, he requires that the structure in question have a lexical category — C — but that the syntax not treat it as a lexical item. How is the distinction drawn by the rest of the syntax?
  - He requires two different kinds of Fuse, which is not only inelegant, but also potentially contradictory, especially if the copy theory of movement is assumed.

A Constraint-Based Alternative

Background on LFG

Lexical-Functional Grammar

- Lexical-Functional Grammar (Kaplan and Bresnan 1982, Bresnan 1982, Dalrymple et al. 1995, Bresnan 2001, Dalrymple 2001) is a constraint-based, model-theoretic theory of grammar.
- Structural descriptions are constraints — statements that can be evaluated for truth (true or false) — that must be satisfied by structures (models).
- LFG postulates multiple structures, each having properties relevant to the linguistic aspect it models.

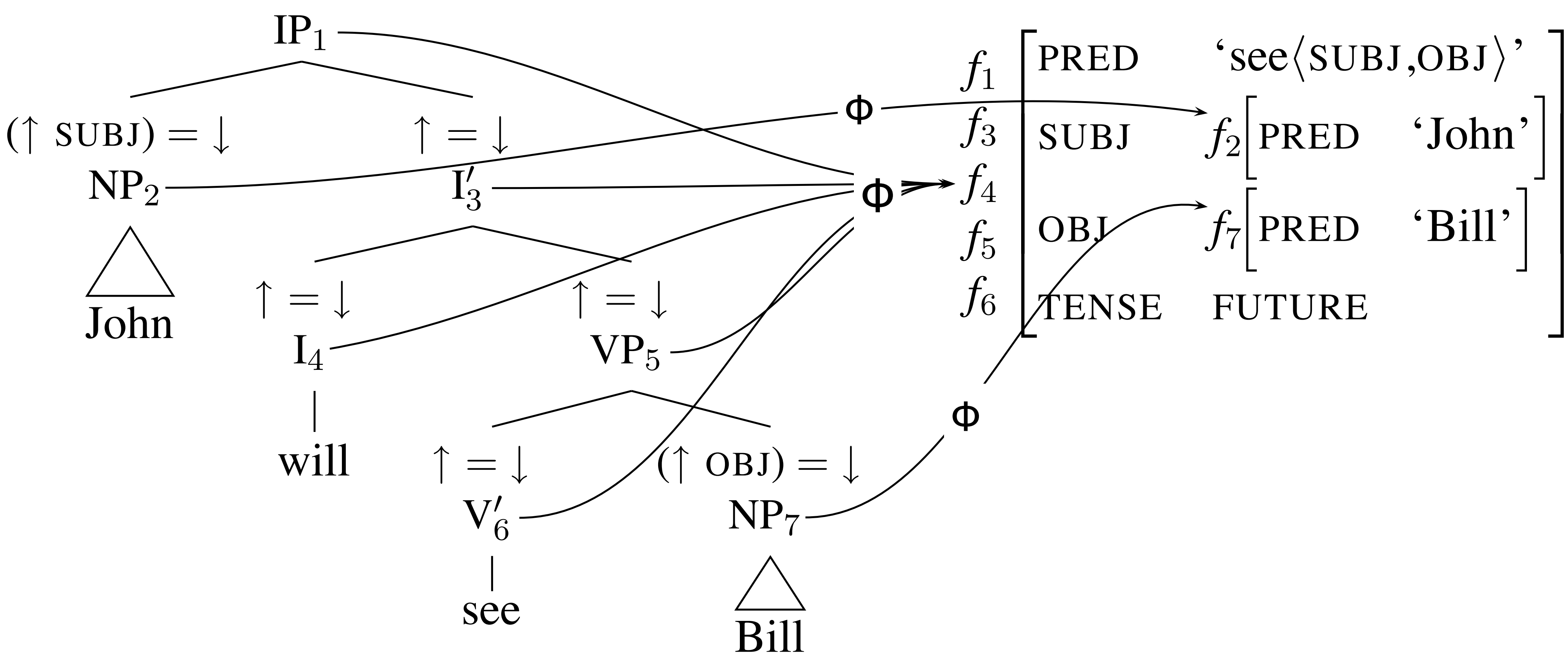
Lexical-Functional Grammar

- For example, constituency, dominance, and word order are described by phrase structure rules that define tree structures. This level of structure is called ‘constituent structure’ or ‘c-structure’ for short.
- Other, more abstract aspects of syntax — such as grammatical functions, predication, agreement, unbounded dependencies, local dependencies, case, binding, etc. — are described by quantifier-free equality statements and define attribute value matrices, a.k.a. feature structures. This level of structure is called ‘functional structure’ or ‘f-structure’ for short.

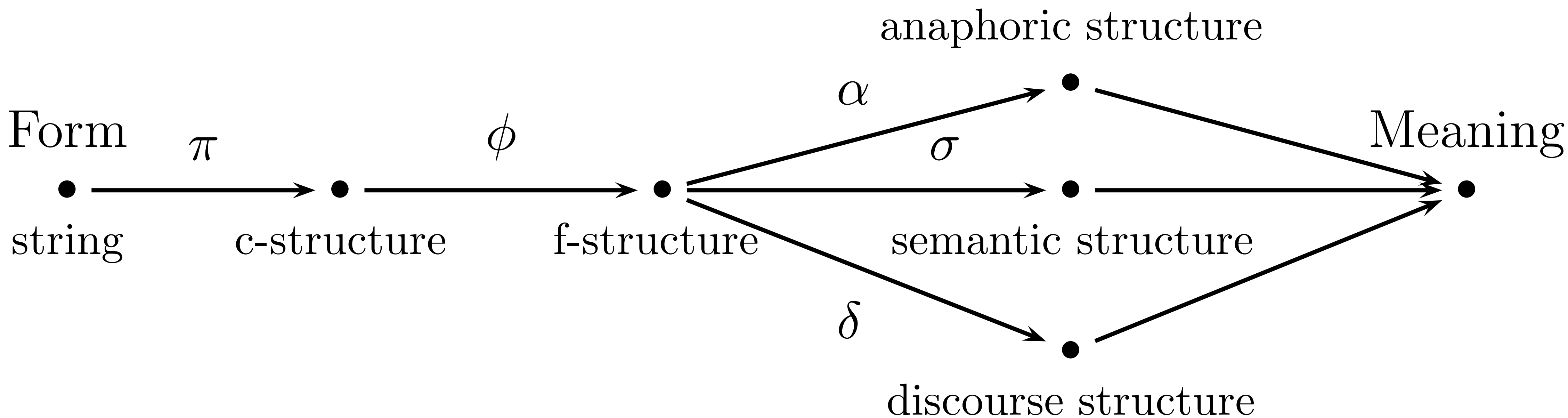
Lexical-Functional Grammar

- Structures are presented in parallel and elements of one structure ‘are projected to’ or ‘correspond to’ elements of other structures according to ‘projection functions’, which are also called ‘correspondence functions’. For example, the function relating c-structure to f-structure is the ϕ function.
- This was subsequently generalized to a ‘Correspondence Architecture’ (Kaplan 1987, 1989, Halvorsen & Kaplan 1988, Asudeh 2006, Asudeh & Toivonen 2008).
- Another term used in the literature is ‘Parallel Projection Architecture’, but this is perhaps best avoided to prevent confusion with Jackendoff’s recent proposals (e.g., Jackendoff 1997, 2002, 2007).

LFG: A Simple Example

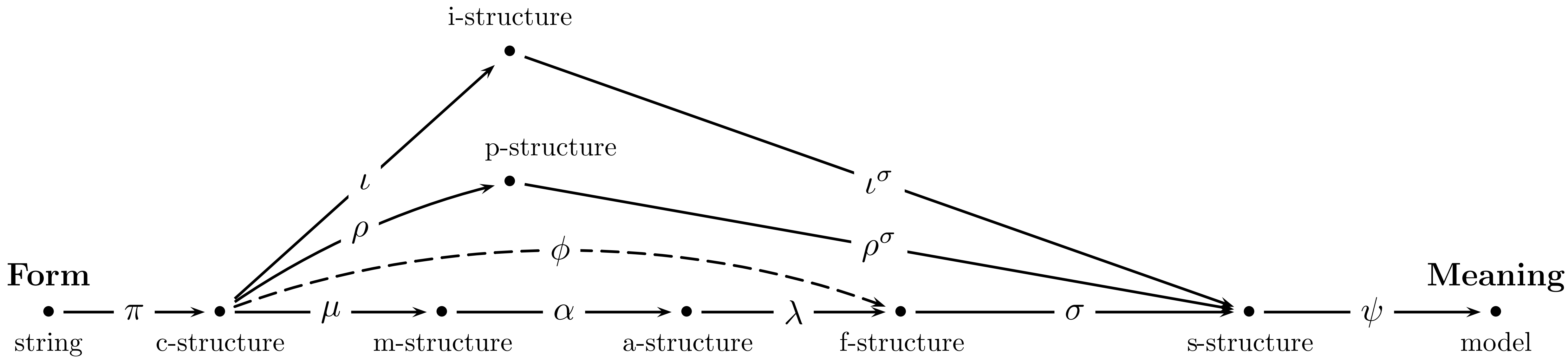


Correspondence Architecture: Programmatic



(Kaplan 1987, 1989)

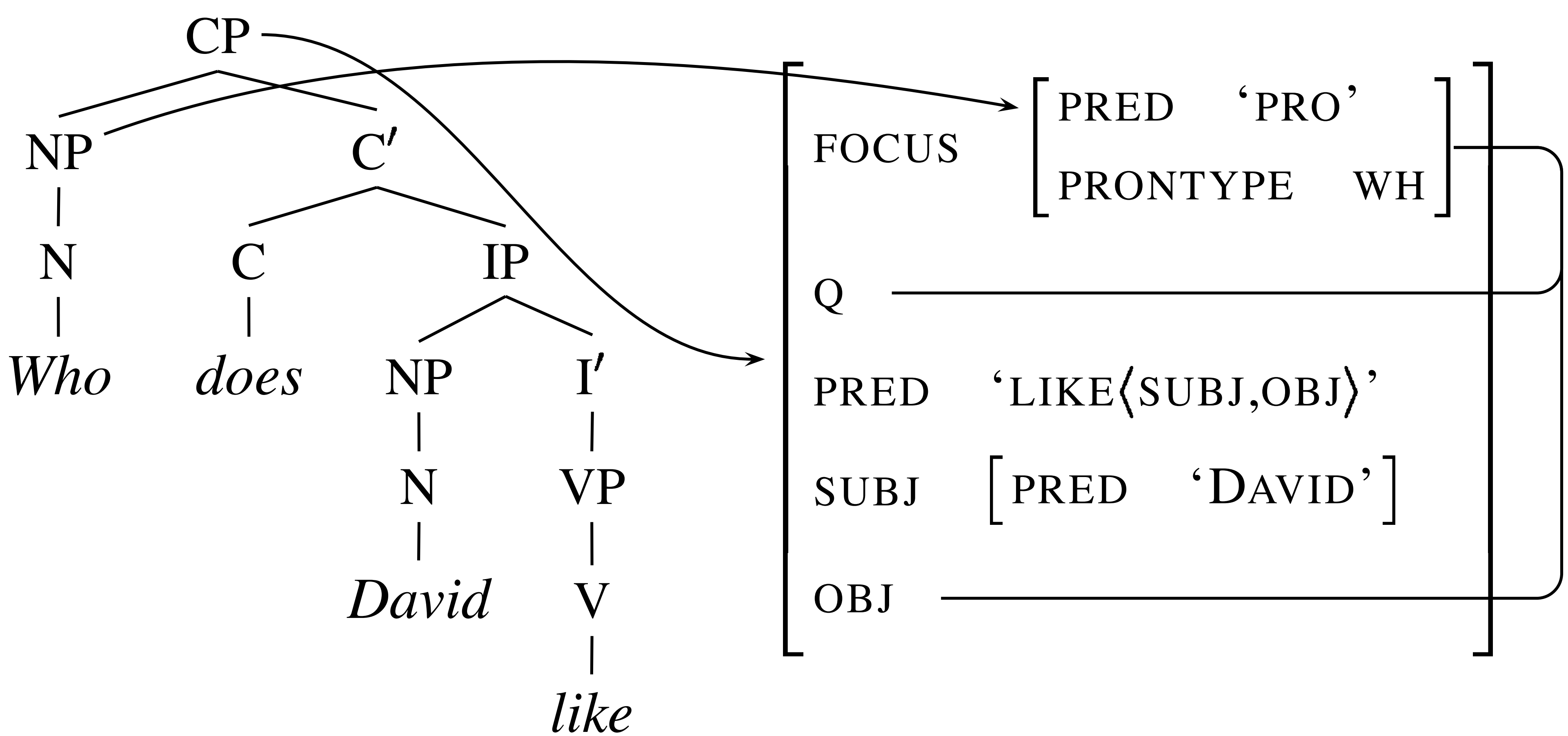
Correspondence Architecture: A Recent Synthesis



(Asudeh 2006, Asudeh & Toivonen 2008)

Unbounded Dependencies: Example

Who does David like?



Note: The examples and rules on this and the following 9 slides are from Dalrymple (2001: ch. 14).

Unbounded Dependencies: Annotated PS Rule

$$\text{CP} \longrightarrow \begin{pmatrix} \text{QuesP} \\ (\uparrow \text{ FOCUS}) = \downarrow \\ (\uparrow \text{ FOCUS}) = (\uparrow \text{ QFOCUSPATH}) \\ (\uparrow \text{ Q}) = (\uparrow \text{ FOCUS WHPATH}) \\ (\uparrow \text{ Q PRONTYPE}) =_c \text{ WH} \end{pmatrix} \begin{pmatrix} \text{C}' \\ \uparrow = \downarrow \end{pmatrix}$$

Unbounded Dependencies: QuesP Metacategory

$$\text{QuesP} \equiv \{\text{NP} \mid \text{PP} \mid \text{AdvP} \mid \text{AP}\}$$

(1) NP: Who do you like?

(2) PP: To whom did you give a book?

(3) AdvP: When did you yawn?

(4) AP: How tall is Chris?

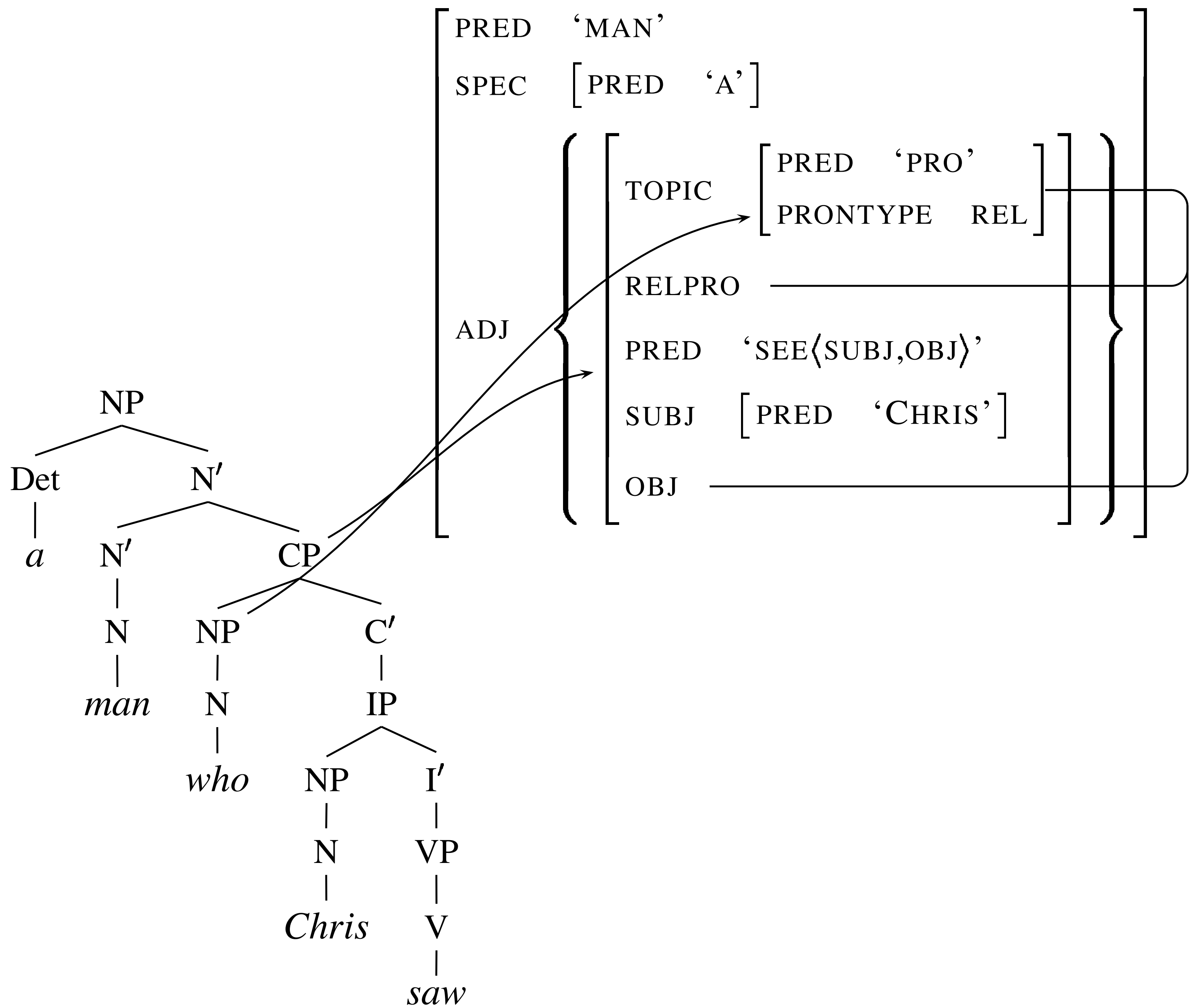
Unbounded Dependency Equation

English QFOCUSPATH:

$$\left\{ \textsc{xcomp} \;\middle|\; \begin{array}{c}\textsc{comp}\\ (\rightarrow \textsc{ldd}) \neq -\end{array} \;\middle|\; \begin{array}{c}\textsc{obj}\\ (\rightarrow \textsc{tense})\end{array} \right\}^{*} \quad \left\{ \left(\textsc{adj} \;\; \begin{array}{c}\in\\ \neg(\rightarrow \textsc{tense})\end{array} \right) \; (\textsc{gf}) \;\middle|\; \textsc{gf} \right\}$$

Relative Clauses: Example

a man who Chris saw



Relative Clauses: Annotated PS Rule

$$\text{CP} \longrightarrow \left(\begin{array}{c} \text{RelP} \\ (\uparrow \text{TOPIC}) = \downarrow \\ (\uparrow \text{TOPIC}) = (\uparrow \text{RTOPICPATH}) \\ (\uparrow \text{RELPRO}) = (\uparrow \text{TOPIC RELPATH}) \\ (\uparrow \text{RELPRO PRONTYPE}) =_c \text{REL} \end{array} \right) \left(\begin{array}{c} \text{C}' \\ \uparrow = \downarrow \end{array} \right)$$

$$\text{CP} \longrightarrow \left\{ \begin{array}{c} \text{RelP} \\ (\uparrow \text{TOPIC}) = \downarrow \\ (\uparrow \text{TOPIC}) = (\uparrow \text{RTOPICPATH}) \\ (\uparrow \text{TOPIC RELPATH}) = (\uparrow \text{RELPRO}) \\ (\uparrow \text{RELPRO PRONTYPE}) =_c \text{REL} \end{array} \middle| \begin{array}{c} \epsilon \\ (\uparrow \text{TOPIC PRED}) = \text{`PRO'} \\ (\uparrow \text{TOPIC}) = (\uparrow \text{RTOPICPATH}) \\ (\uparrow \text{TOPIC}) = (\uparrow \text{RELPRO}) \\ \end{array} \right\} \left(\begin{array}{c} \text{C}' \\ \uparrow = \downarrow \end{array} \right)$$

Relative Clauses: RelP Metacategory

$$\text{RelP} \equiv \{\text{NP} \mid \text{PP} \mid \text{AP} \mid \text{AdvP}\}$$

(1) NP: a man <u>who</u> I selected

(2) PP: a man <u>to whom</u> I gave a book

(3) AP: the kind of person <u>proud of whom</u> I could never be

(4) AdvP: the city <u>where</u> I live

Relative Clauses: Unbounded Dependency Equation

English RTOPICPATH:

$$\left\{ \text{XCOMP} \mid \underset{(\rightarrow\ \text{LDD}) \neq -}{\text{COMP}} \mid \underset{(\rightarrow\ \text{TENSE})}{\text{OBJ}} \right\}^{*} \left\{ \left(\text{ADJ} \underset{\neg(\rightarrow\ \text{TENSE})}{\in} \right) (\text{GF}) \mid \text{GF} \right\}$$

Relative Clauses: Pied Piping

English RELPATH:

$$\left\{ \text{SPEC}^* \;\middle|\; \left[\left(\text{OBL}_\theta \right) \text{OBJ} \right]^* \right\}$$

(1) the man [who] I met

(2) the man [whose book] I read

(3) the man [whose brother's book] I read

(4) the report [the cover of which] I designed

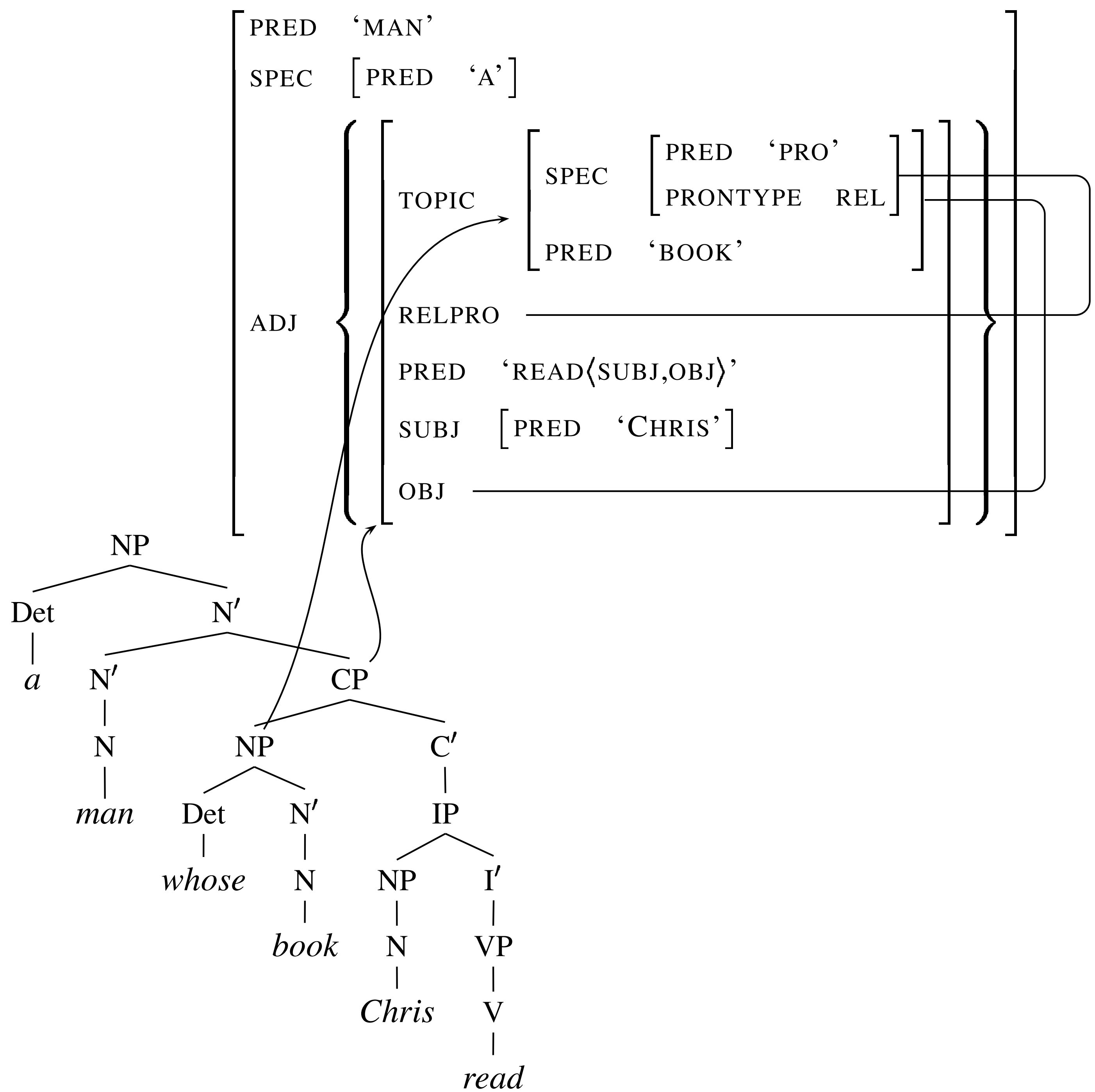
(5) the man [faster than whom] I can run

(6) the kind of person [proud of whom] I could never be

(7) the report [the height of the lettering on the cover of which] the government prescribes

Relative Clauses: Pied Piping Example

a man whose book Chris read



CANE: A New Analysis

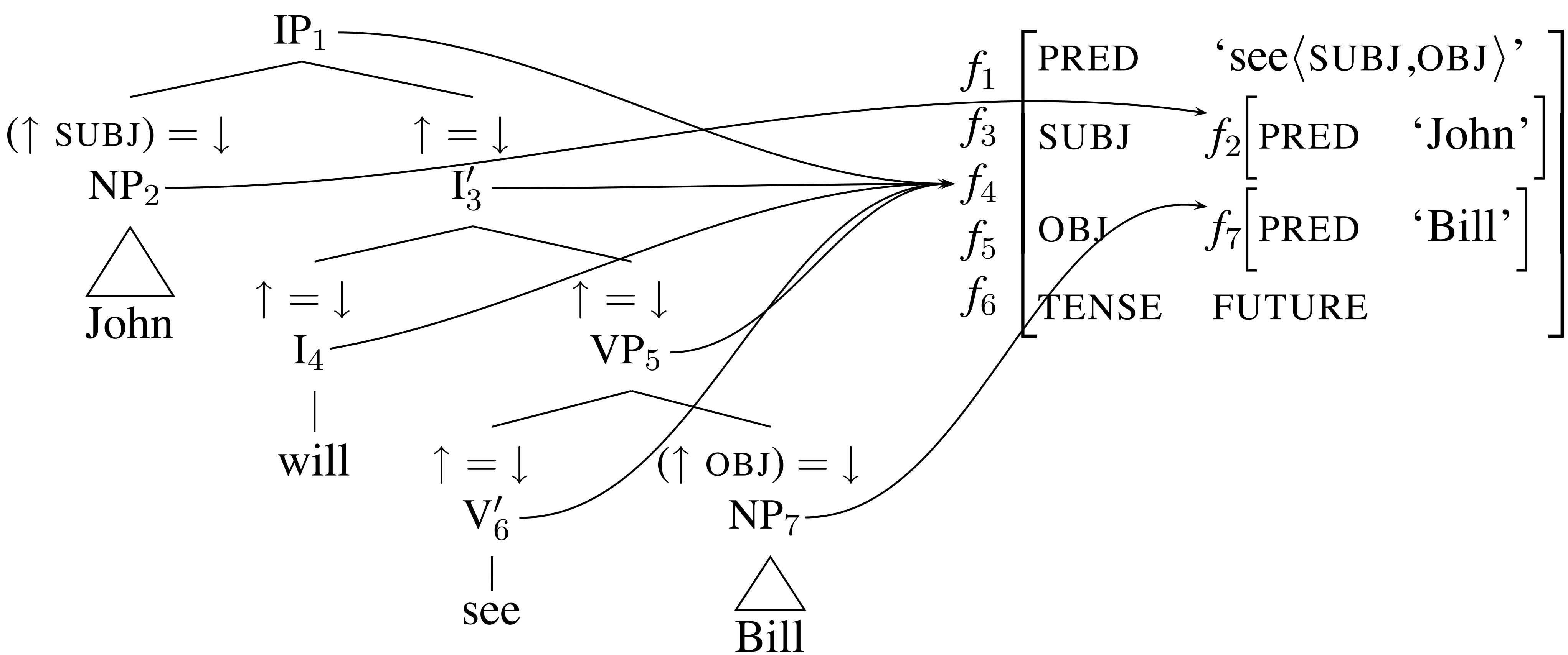
Overview

- **Key insight**:
  LFG's Correspondence Architecture has everything in place for a compact, elegant treatment of CANE Effects; in particular: a way to talk about *string adjacency*.
- This novel analysis stems from examining the architecture carefully and making explicit certain implicit, native mechanisms.
  - No extension of architecture or mechanisms
- Some highlights:
  - Mathematically simple, precise and tractable
  - Lexicalist analysis: variation
  - A single lexical entry for *that* in complement and relative clauses
  - Accounts for CANE Effect and the Adverb Effect
  - Relative Clause Paradox resolved

Inverse Correspondences

- We noted earlier that a central aspect of LFG's projection architecture are the correspondence functions that map one structure to another, such as the function ϕ that maps c-structure to f-structure.
- Inverse correspondences can then be defined as the inverse relation of the original correspondence function.
- For example, the inverse of the ϕ function is written ϕ^{-1} and returns the set of c-structure nodes that map to its argument f-structure node.

Inverse Correspondence: ϕ^{-1}



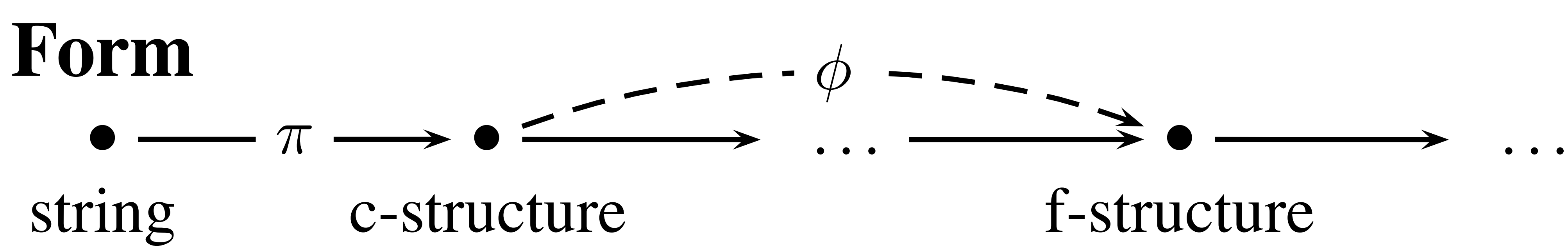
$$\phi(1) = f_1$$

$$\phi^{-1}(f_1) = \{1, 3, 4, 5, 6\}$$

⋮

CANE in LFG: The Basic Intuition

- We can leverage LFG's projection architecture to capture the fact that CANE Effects are a 'surfacey' phenomenon (cf. ECP as a PF constraint in recent Minimalism).
- The relevant relation for CANE seems to be *linear adjacency*, rather than structural superiority or other, more articulated syntactic notions.
- The part of the architecture that we need to pay special attention to is therefore the mapping from (tokenized) strings to c-structure, which we'll call π (pi), following Kaplan (1987,1989).



The Syntax-Phonology Interface

- Linear adjacency is not a syntactic notion, since syntactic relations are structural.
- Linear adjacency essentially concerns the phonological ordering of syntactic entities (words): linearization.
- The string in the Correspondence Architecture is the ordered yield of the syntactic tree and is therefore phonologically parsed (segmented).

The Syntax-Phonology Interface

- Another relevant aspect of the syntax-phonology interface is the notion of phonological realization of syntactic entities.
- Notice that the version of LFG I am assuming has no empty c-structure nodes, but there may be elements of f-structure that have no c-structural correspondent and are therefore phonologically unrealized. (If an element has no c-structural correspondent, it follows that it has no string correspondent).
- The inverse correspondence function, ϕ^{-1}, is used to define a predicate REALIZED:

$$\textsc{realized}(f) \;\text{ iff }\; \phi^{-1}(f) \neq \emptyset\text{, where } f \text{ is an f-structure}$$

Linear Adjacency as String Adjacency

- Assume a native precedence function on strings, yielding a notion of element that is string-adjacent to the right (‘next string element’):
  - N: W → W, where W is the set of words (string elements)
  - Notice that we’re here assuming a tokenized (i.e., phonologically parsed) string, but nothing much hinges on this. In any case, tokenization needs to be performed for lexical look-up and is almost certainly ‘psychologically real’ in some sense.
- If * is the current c-structure node, then $\pi^{-1}(*)$ is the string element that maps to * and $N(\pi^{-1}(*))$ is the string element that immediately follows $\pi^{-1}(*)$.
- The * notation may be somewhat unfamiliar, but it lies behind the more familiar f-structure metavariables, ↑ and ↓:
  $\phi(*) = \downarrow$ and $\phi(\mathcal{M}(*)) = \uparrow$, where $\mathcal{M}$ is the mother function on tree nodes.

String Adjacency and Mapping to F-Structure

- Note that π^{-1} returns string elements, not sets of string elements, because π is injective, since c-structures are trees.
  - In other words, each word in the string is mapped to a single (terminal) node in c-structure (cf. Lexical Integrity).
- We are going to use f-structural relations to explain CANE Effects, so it will be useful to define an f-structure metavariable for the f-structure of the following string element:
  - $> := \phi(\mathcal{M}(\pi(N(\pi^{-1}(*)))))$
  - The semantics of $>$ is ‘the f-structure of the mother of the c-structure correspondent of the string element that follows (the string correspondent of) the current c-structure node’.
    - Note: We need to refer to a mother node above, because terminal c-structure nodes are not typically directly mapped to f-structure. This will become clearer shortly.

CANE at the Syntax-Phonology Interface in LFG

- We can use REALIZED and > to capture the superficial nature of the CANE Effect, while both capturing the Adverb Effect and resolving the Relative Clause Paradox.
- Basically, CANE Effect languages, like English, have a (somewhat arbitrary) constraint that the right-adjacent string element to the complementizer must be locally realized.
- We may want to state this constraint on unbounded dependency functions (TOPIC, FOCUS), but for English we can make the simplifying assumption that a statement about SUBJECT will suffice.

CANE at the Syntax-Phonology Interface in LFG

- The necessary constraint then requires that if the subject of the next string element following the complementizer is realized, it cannot also fill an unbounded dependency function (UDF).
- In other words, there is a constraint against the subject being both phonologically realized and displaced.

$\neg[\text{REALIZED}(\succ \text{SUBJ}) \;\wedge\; (\text{UDF}(\succ \text{SUBJ}))]$
where UDF is an unbounded dependency function (FOCUS or TOPIC)

Lexical Entries

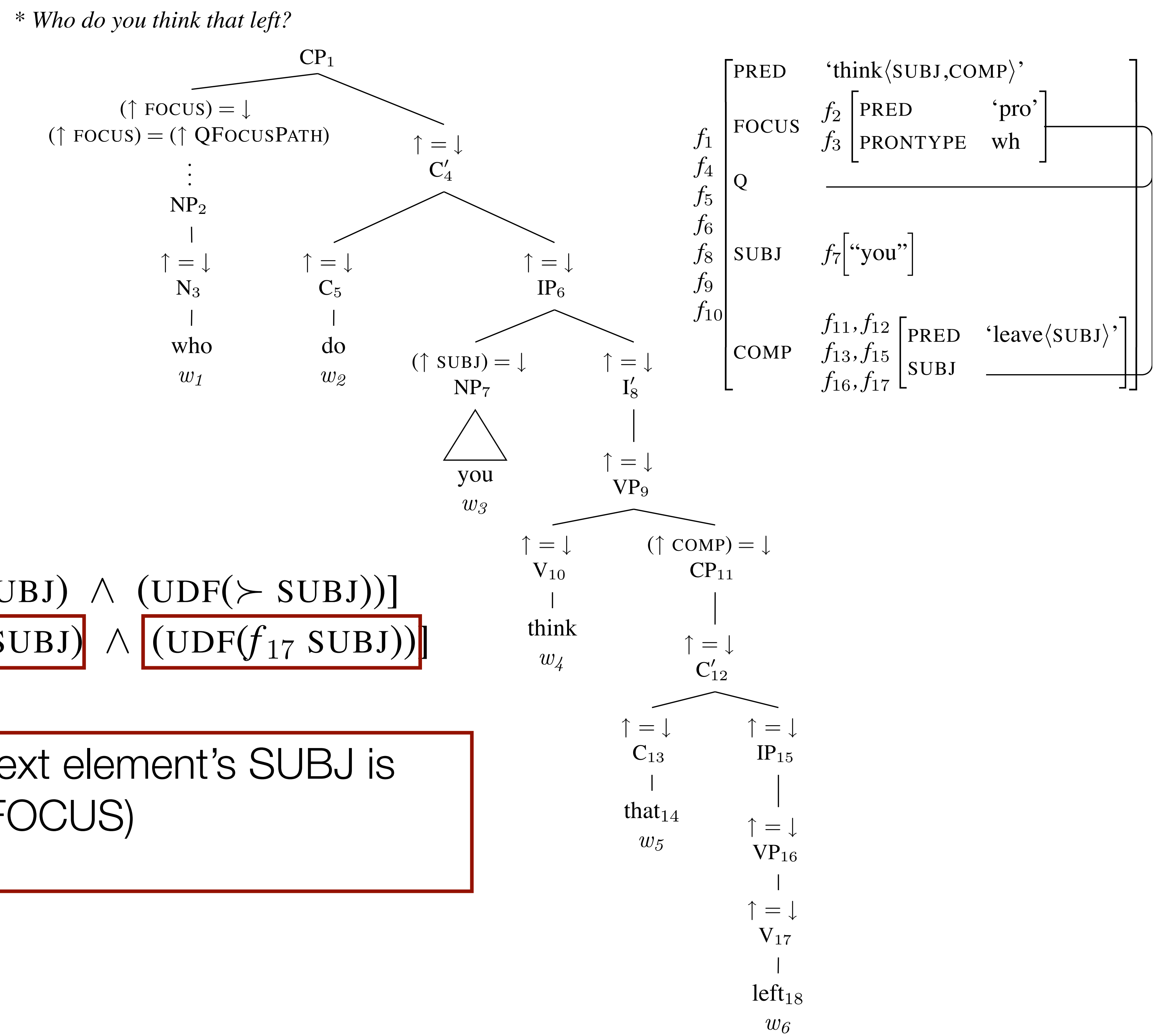
that C $(\uparrow \text{TENSE})$
$(\uparrow \text{MOOD}) = \text{DECLARATIVE}$
$\neg[\text{REALIZED}(\succ \text{SUBJ}) \;\wedge\; (\text{UDF}(\succ \text{SUBJ}))]$

if C $(\uparrow \text{TENSE})$
$(\uparrow \text{MOOD}) = \text{IRREALIS}$
$\neg[\text{REALIZED}(\succ \text{SUBJ}) \;\wedge\; (\text{UDF}(\succ \text{SUBJ}))]$

whether C $(\uparrow \text{MOOD}) = \text{INTERROGATIVE}$
$\neg[\text{REALIZED}(\succ \text{SUBJ}) \;\wedge\; (\text{UDF}(\succ \text{SUBJ}))]$

Note: The entries contain redundant information for clarity. The redundancies are eliminable through templates (Dalrymple, Kaplan & King 2004), which are also relevant to the lexicon-syntax interface (Asudeh, Dalrymple & Toivonen 2008).

Analysis: Basic CANE Effects



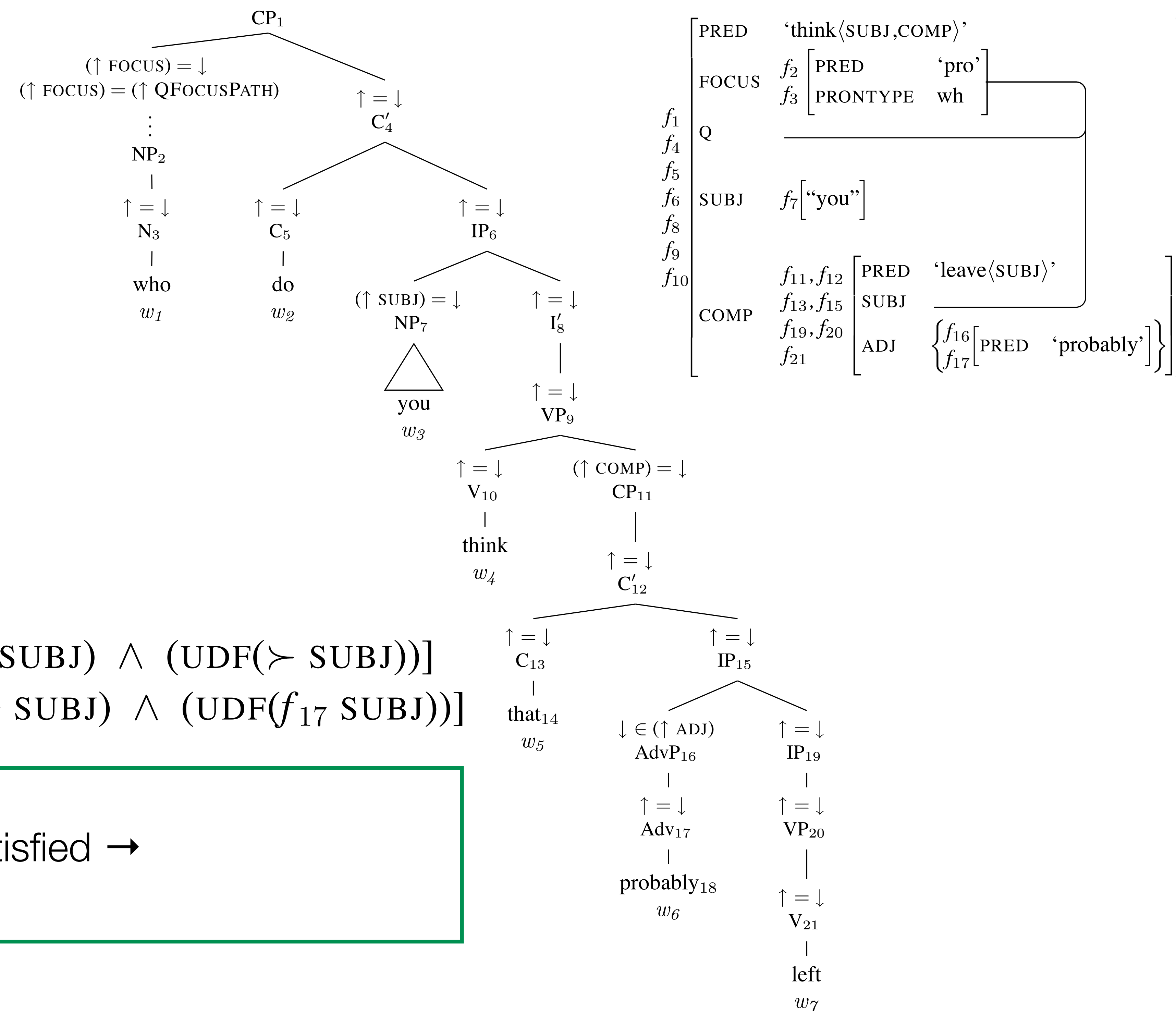
that: $\neg[\text{REALIZED}(\succ \text{SUBJ}) \;\wedge\; (\text{UDF}(\succ \text{SUBJ}))]$

$= \neg[\text{REALIZED}(f_{17}\ \text{SUBJ}) \;\wedge\; (\text{UDF}(f_{17}\ \text{SUBJ}))]$

Constraint not satisfied: Next element’s SUBJ is REALIZED and is a UDF (FOCUS)
→ **ungrammatical**

Analysis: The Adverb Effect

Who do you think that probably left?

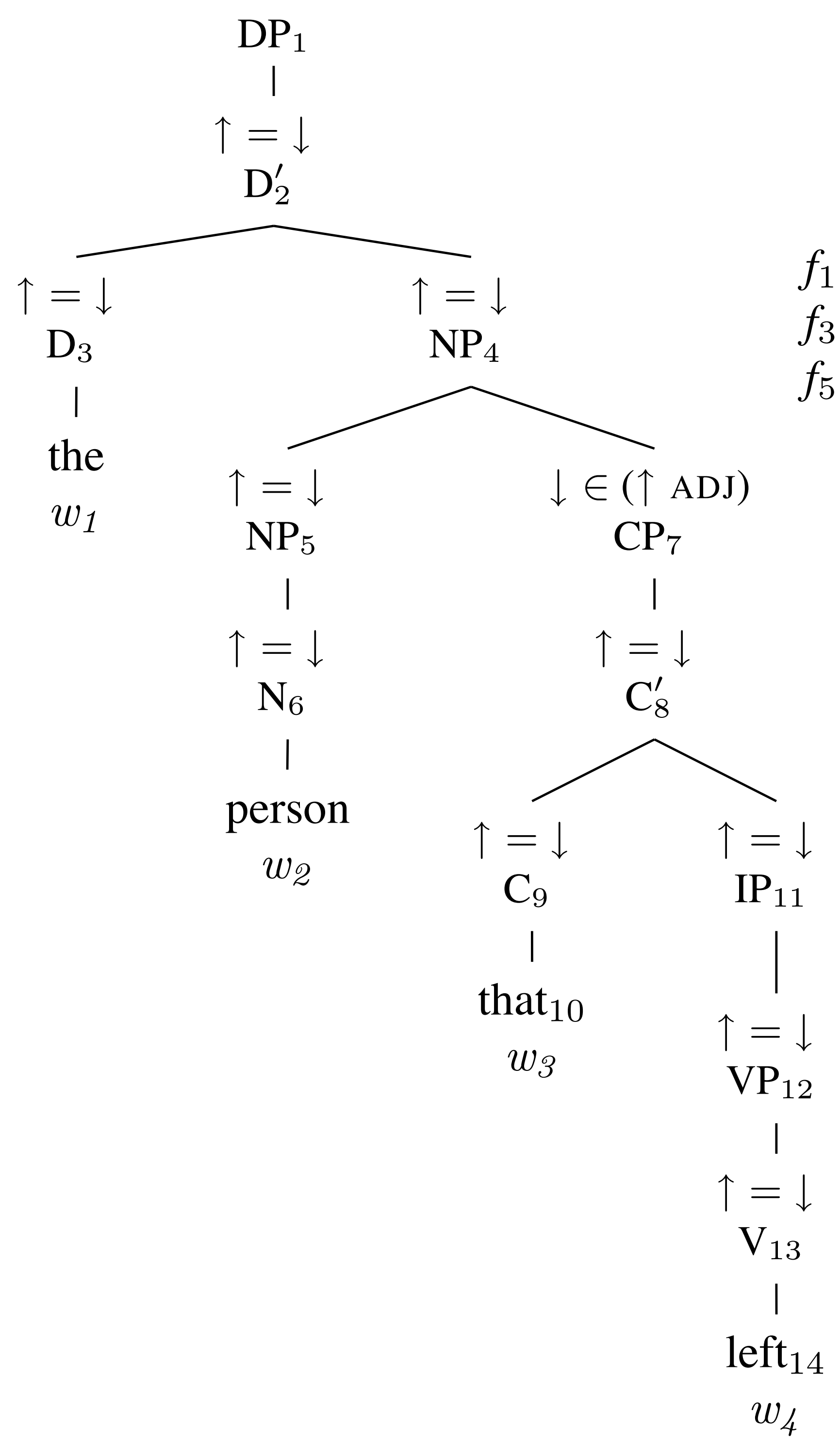


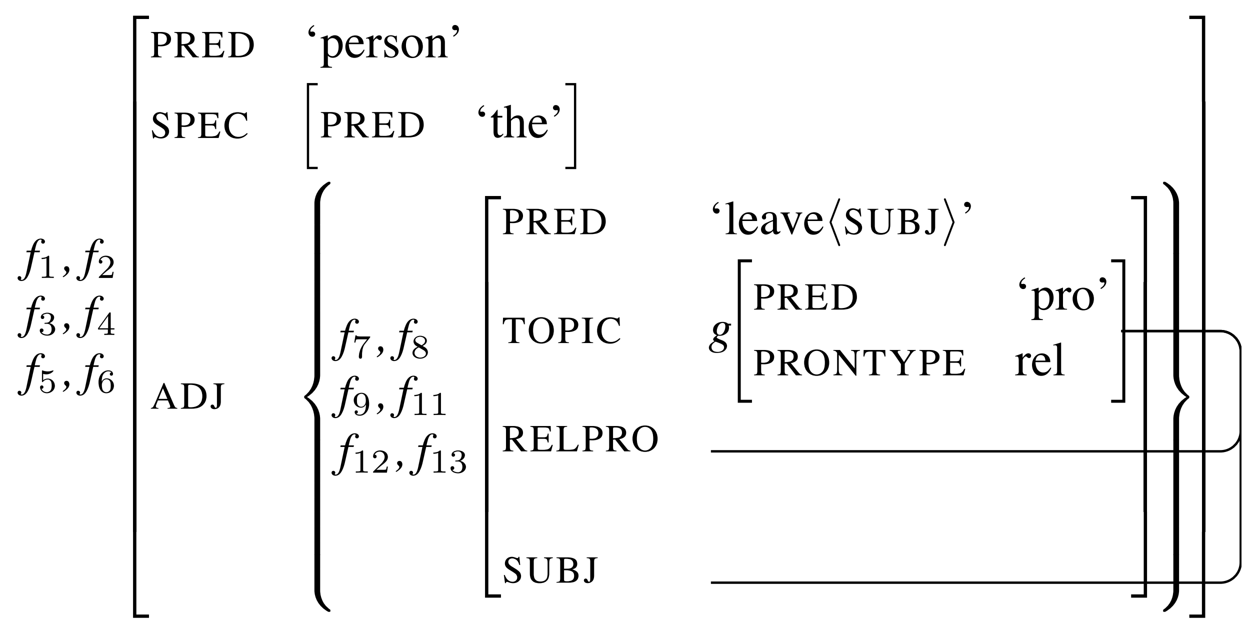
$$\textit{that}: \quad \neg[\text{REALIZED}(\succ \text{SUBJ}) \ \wedge \ (\text{UDF}(\succ \text{SUBJ}))]$$
$$= \neg[\text{REALIZED}(f_{17}\ \text{SUBJ}) \ \wedge \ (\text{UDF}(f_{17}\ \text{SUBJ}))]$$

f_{17} has no SUBJ →
constraint (vacuously) satisfied →
grammatical

Analysis: Resolving the Relative Clause Paradox

the person that left





$$\textit{that}: \quad \neg[\text{REALIZED}(\succ \text{SUBJ}) \wedge (\text{UDF}(\succ \text{SUBJ}))]$$

$$= \neg[\text{REALIZED}(g) \wedge (\text{UDF}(g))]$$

g is a null pronoun →
no c-structure correspondent →
not REALIZED →
constraint satisfied →
grammatical

Some Consequences

- No unmotivated, multiple *that* complementizers (contra Gazdar 1981, Pollard & Sag 1994, Sobin 2002, Bošković & Lasnik 2003, Branigan 2004).
- No unmotivated operations on CP structures.
- A wider range of empirical data captured (CANE and Adverb Effects and Relative Clause Paradox).
- Dialectal and cross-linguistic variation explained as lexical variation: if a complementizer lacks the constraint, there is no CANE effect.
  - This explains not just basic variation, but also the otherwise puzzling fact that certain Scandinavian dialects have CANE with (the equivalent of) *that*, but have no CANE with (the equivalent of) *if* (Branigan 2004), even though extraction would normally be expected to be much harder across an *if* complementizer.

A Fallacy

- Notice that it is a fallacy to expect that the same constraint that explains CANE Effects should predict that *that* is obligatory in relative clauses (1) or that extraction across an *if/whether* complementizer is degraded in general (2).

(1) * This is the man __ sells fish.

(2)* Who do you wonder whether he believes __ sells fish.

- First, (1) is grammatical in some dialects.
- Second, the obligatoriness of *that* (or a relative pronoun) in one circumstance is logically independent of its obligatory absence in other circumstances.
- Third, there are independently motivated constraints on extraction that explain (2).

Conclusion

- The Correspondence Architecture of LFG involves multiple structures.
- Interfaces between structures are captured by correspondence functions.
- Correspondence functions can be used to state relational constraints on parallel structures.
- The string-to-tree mapping, π, has not previously received much attention, but it facilitates an elegant solution to the CANE Effect (a.k.a. Comp-Trace Effect), without introducing new theoretical assumptions or architectural extensions.
- The solution furthermore captures the Adverb Effect and resolves the Relative Clause Paradox in a simple and precise fashion.
- The projection and precedence approach is thus arguably theoretically and empirically superior to previous solutions.

Future Work

- I made the simplifying assumption that the relevant constraint can be captured by reference to SUBJ. It would be interesting to see if CANE Effects are observed in any language for any other grammatical function, in which case we would likely need to make reference to unbounded dependency functions, instead.
- Wescoat (2002, 2005, 2007) has proposed a radical reconsideration of the string to tree mapping in which words can map to multiple categories. I think a modified version of the account given here would work in his system, but this needs to be investigated.
- There are other syntactic phenomena that superficially seem quite dissimilar to CANE, but which involve similar notions of adjacency; e.g. syntactically-conditioned mutation in Celtic. Could they receive a similar treatment in terms of the π mapping?

Thank you

Research supported by the
Social Sciences and Humanities Research Council of Canada,
Standard Research Grant 410-2006-1650

http://www.carleton.ca/~asudeh/