The Resource Logic of Complex Predicate Interpretation

Mary Dalrymple
Angie Hinrichs
John Lamping
Vijay Saraswat

{dalrymple,hinrichs,lamping,saraswat}@parc.xerox.com

Xerox PARC
Palo Alto CA 94304

August 13, 1993

1 Introduction

Complex predicate constructions involve a mismatch between syntax and semantics: they are syntactically monoclausal, but involve a complex semantic structure. Mohanan (1993) gives the following definition of complex predicates:

> A COMPLEX PREDICATE construction is one in which two semantically predicative elements jointly determine the structure of a single syntactic clause.

Consider the following example, due to Butt (1993a):[1]

(1) Anjum-ne Saddaf-ko xat likhne diyaa
Anjum-ERG Saddaf-DAT letter-NOM write-INF-OBL let
'Anjum let Saddaf write a letter.'

Butt (1993a) shows that the two semantically predicative elements *likhne* 'write' and *diyaa* 'permit' jointly make up a single complex predicate of a syntactically monoclausal sentence, while giving rise to a complex semantic representation. A monoclausal structure arises when the two predicative elements combine to form a single syntactic unit; the relation between the monoclausal syntactic representation and the complex semantic representation is thus nonisomorphic. We propose an approach to syntactically-derived complex predicates which produces the correct syntactic structure and correctly specifies the relation between the syntax and the meaning.

Like other recent approaches within the Lexical-Functional Grammar (LFG) framework, we use *mapping rules* to specify the relation between grammatical functions like 'subject' and 'object' and semantic/thematic roles. Our approach differs in employing a resource logic, *linear logic*, to guide the composition of meanings; this meshes well with the treatment of other semantic phenomena, such as quantification and modification, and gives rise to a clean characterization of *completeness* and *coherence* constraints (Dalrymple, Lamping, and Saraswat 1993; Dalrymple, Lamping, Pereira, and Saraswat 1993). The key principles of a theorem prover which efficiently carries out the linear deductions necessary for handling complex predicates are described in the Appendix.

[1] ERG = ergative case marking; DAT = dative case marking; NOM = nominative/zero case; INF-OBL = oblique infinitive.

2 The nature of complex predicates

Much work has been devoted to the analysis of complex predicate constructions (Butt, Isoda, and Sells 1990, Manning 1992, Alsina 1993a, Butt 1993a, and works cited therein, among many others), and in particular to the syntactic features of these constructions. Manning (1992) discusses Romance 'restructuring verbs' appearing in constructions such as the following Spanish example:

(2) a. Luis trató de comer-las
Luis try of eat-INF-them
'Luis tried to eat them.'

b. Luis las trató de comer
Luis them try of eat-INF
'Luis tried to eat them.'

Manning argues that the appearance of the clitic *las* on the higher verb *trató* in example 2b indicates that the two predicative elements *trató* 'try' and *comer* 'eat' have formed a complex predicate, and that the clitic *las* 'them' appears as its object. He provides evidence against the view that structures involving restructuring verbs, such as 2b, are biclausal and involve subordination of the verb 'eat'; the conclusion he draws is that structures such as 2b are syntactically monoclausal, though semantically complex.

Manning's evidence regarding adverbial interpretation is especially important, since it provides evidence for both the syntactic monoclausality and the semantic complexity of complex predicates. Generally, adverbials modify the meaning of the clause in which they are contained. In this light, note that the following Catalan example (Alsina 1991, cited by Manning) is ambiguous:

(3) volia tastar amb molt d'interès la cuina tailandesa
I wanted to taste with much interest the cuisine Thai
'I wanted to taste Thai food with much interest.'

The adverbial appears adjacent to the verb *tastar* 'taste'; if the construction were biclausal, the adverbial 'with much interest' would be expected to modify only *taste*. However, two readings are available for this sentence, and in fact the reading in which *with much interest* modifies *want to taste* is preferred. This is expected if the structure is syntactically monoclausal (the modifier appears in the single clause which contains both verbs) but semantically complex (two semantic predicates are available for modification).

As discussed by Butt (1993a) and Alsina (1993b), phrase structure criteria do not differentiate between complex predicates and multiclausal structures. Within the framework of LFG, the syntactic monoclausality of Romance restructuring verb constructions is manifested at the level of *functional structure* or *f-structure*. This is the level at which syntactic argument structure is represented. The f-structure representation for an example like 2b is:

(4)
$$\begin{bmatrix} \textsc{pred} & \text{`TRY}\langle\text{EAT}\rangle\text{'} \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{`Luis'} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{`THEM'} \end{bmatrix} \end{bmatrix}$$

At this level, the two verbs 'try' and 'eat' combine to form a single complex predicate with a single array of grammatical functions. The complex predicate requires a subject ('LUIS') and an object ('THEM').

The analysis presented here addresses two issues: (1) determining the correct inventory of grammatical functions for syntactic realization of the arguments of the complex predicate (for example 2b, the grammatical functions SUBJ and OBJ), and (2) associating an appropriate complex semantic representation with the complex predicate. Related issues, such as the question of determining the PRED value for complex predicates, will be discussed in Section

6. Our discussion draws primarily from work by Butt (1990; 1993a; 1993b) on complex predicates in a dialect of the Indo-Aryan language Urdu spoken in Lahore, Pakistan.

3 Urdu complex predicates

Butt (1990; 1993a; 1993b) discusses the 'permissive construction' in Urdu, illustrated in 5:

(5) Anjum-ne Saddaf-ko xat likhne diyaa
Anjum-ERG Saddaf-DAT letter-NOM write-INF-OBL let
'Anjum let Saddaf write a letter.'

This construction, which Butt demonstrates to be a complex predicate construction, is superficially similar to the 'tell construction':

(6) Anjum-ne Saddaf-ko xat likhne-ko kahaa
Anjum-ERG Saddaf-DAT letter-NOM write-INF-OBL-ACC told
'Anjum told Saddaf to write a letter.'

The constituent phrase structure of these examples is the same, and the same scrambling possibilities are available for both. In particular, both constructions allow scrambling of the embedded VP to postverbal position:

(7) Anjum-ne diyaa $_{VP}$[Saddaf-ko xat likhne]
Anjum-ERG let Saddaf-DAT letter-NOM write-INF-OBL
'Anjum let Saddaf write a letter.'

(8) Anjum-ne kahaa $_{VP}$[Saddaf-ko xat likhne-ko]
Anjum-ERG told Saddaf-DAT letter-NOM write-INF-OBL-ACC
'Anjum told Saddaf to write a letter.'

Note in particular that neither construction requires the two verbs to appear adjacent to one another, nor even within the same VP. Since tests for constituency do not distinguish the 'let construction' from the 'tell construction', Butt concludes that at the level of constituent structure, complex predicates are not distinguishable from constructions involving subordination.

However, Butt shows that the 'tell construction' is syntactically complex, involving subordination of a nonfinite clause, and is thus not a complex predicate construction. F-structures for these two examples are:

(9) Examples 5, 7:

$$\begin{bmatrix} \text{PRED} & \text{'PERMIT}\langle\text{WRITE}\rangle\text{'} \\ \text{SUBJ} & [\text{PRED} \ \ \text{'ANJUM'}] \\ \text{OBJ2} & [\text{PRED} \ \ \text{'SADDAF'}] \\ \text{OBJ} & [\text{PRED} \ \ \text{'LETTER'}] \end{bmatrix}$$

(10) Examples 6, 8:

$$\begin{bmatrix} \text{PRED} & \text{'TELL'} \\ \text{SUBJ} & [\text{PRED} \ \ \text{'ANJUM'}] \\ \text{OBJ} & [\text{PRED} \ \ \text{'SADDAF'}]_{1} \\ \text{COMP} & \begin{bmatrix} \text{PRED} & \text{'WRITE'} \\ \text{SUBJ} & [\ ]_{1} \\ \text{OBJ} & [\text{PRED} \ \ \text{'LETTER'}] \end{bmatrix} \end{bmatrix}$$

Evidence for this comes from verb agreement. The Urdu verb agrees with the nominative argument that is highest on a hierarchy of grammatical functions. If none of its arguments is nominative, the verb shows default (third person masculine singular) agreement. Agreement patterns differ for the two constructions (Butt, 1993b, p. 3):

(11) Anjum-ne Saddaf-ko ciṭṭhi likhne dii
Anjum-ERG Saddaf-DAT note(FemSg)-NOM write-INF-OBL let-FemSg
'Anjum let Saddaf write a note.'

(12) Anjum-ne Saddaf-ko ciṭṭhi likhne-ko kahaa
Anjum-ERG Saddaf-DAT note(FemSg)-NOM write-INF-OBL-ACC told-MascSg
'Anjum told Saddaf to write a note.'

The verb *dii* 'permit' agrees with the feminine singular noun *ciṭṭhi* 'note', showing that 'note' is a syntactic argument of 'permit'. However, the verb *kahaa* 'told' shows default agreement, indicating that 'note' is not an argument of 'tell'.

Evidence from anaphora and the control properties of participial adverbials also indicates that these two constructions differ: the 'let construction' is syntactically monoclausal, whereas the 'tell construction' is syntactically complex and involves subordination. See Butt (1993a; 1993b) for discussion of these examples.

While the syntactic structure of the 'let construction' is simple, the semantic structure is demonstrably complex. The following example is ambiguous along the same lines as the Catalan example discussed above (Butt, 1993a):

(13) Anjum-ne Saddaf-ko haar jaldi-se banaane diyaa
Anjum-ERG Saddaf-DAT necklace-NOM quickly make-INF-OBL let
'Anjum quickly let Saddaf make the necklace.' OR
'Anjum let Saddaf make the necklace quickly.'

The ambiguity of this example shows that there are two semantic predicates available for modification: that the semantic representation is complex, although the syntactic structure is monoclausal. For purposes of exposition, we can assume the following complex semantic representation for a sentence such as 5:

(14) $permit(Anjum, write(Saddaf, letter))$

The analysis of complex predicates requires, then:

- an explanation of how two different verbs can combine to license a single array of grammatical functions: the *syntactic monoclausality* problem
- a characterization of how a syntactically monoclausal structure can correctly relate to a semantically complex representation: the *syntax-semantics nonisomorphy* problem

We turn now to a discussion of the framework within which we can provide an answer to these questions.

4 Semantic composition as deduction in linear logic

Dalrymple, Lamping, and Saraswat (1993) and Dalrymple, Lamping, Pereira, and Saraswat (1993) present an approach to semantic composition within the LFG framework which makes use of a fragment of *linear logic* (Girard, 1987) to state constraints on the composition of meanings.[2] This section gives an overview of our approach, presenting the basic assumptions that will be needed in our treatment of complex predicates.

A key principle of our approach is to distinguish meanings from the formalism that assembles meanings. The *language of meaning* can be any suitable logical language for expressing truth-conditional meanings of words and phrases; for current purposes, first-order logic will suffice. The *language of assembling meanings* or *glue language* is a fragment of linear logic (the *multiplicative fragment*) used to express constraints on the composition of meanings. That is, the glue language expresses how the meanings of words and phrases can combine.

Consider the following lexical entry:

(15) Anjum NP $(\uparrow \text{PRED}) = \text{'ANJUM'}$
$\uparrow_\sigma \rightsquigarrow Anjum$

[2] See Scedrov (1993) for a tutorial introduction to linear logic; see also Saraswat (1993).

Intuitively, we read the expression $\uparrow_\sigma \rightsquigarrow$ *Anjum* as saying "the meaning of $\uparrow$ is *Anjum*'. The up arrow '$\uparrow$' is, as usual, a metavariable representing the f-structure introduced by a use of the lexical item (Kaplan and Bresnan, 1982).

Our analysis requires a means of specifying the meanings of particular syntactic objects and the relations between these meanings. We do this by use of the *projection architecture* of LFG (Kaplan, 1987; Halvorsen and Kaplan, 1988), which postulates a functional relationship between f-structures and the meanings they bear. Notationally, the subscript σ represents the *projection function* from f-structures to semantic representations. The expression '$\uparrow_\sigma$' represents the semantic projection of the f-structure $\uparrow$, for example, and the expression '$\uparrow_\sigma \rightsquigarrow$ *Anjum*' indicates that the meaning *Anjum* is recorded as the meaning of $\uparrow$. The above entry states, then, that *Anjum* is a lexical item of category NP, that it introduces an f-structure whose PRED is 'ANJUM', and that the meaning of that f-structure is *Anjum*.

The lexical entry for a verb like *likhaa* 'write' is:

(16) likhaa V ($\uparrow$ PRED)= 'WRITE'
$\forall X, Y.\ agent((\uparrow \text{PRED})_\sigma, X) \otimes theme((\uparrow \text{PRED})_\sigma, Y) \multimap \uparrow_\sigma \rightsquigarrow write(X, Y)$

The verb *likhaa* introduces an f-structure with a PRED value of 'WRITE'. Additionally, it specifies information about the thematic roles of its arguments and how these arguments go into making up the meaning of a sentence with *likhaa*. This information is given by the linear logic expression in the second line of the lexical entry. For the present, we can think of the linear multiplicative connective $\otimes$ as analogous to standard conjunction $\wedge$, and linear implication $\multimap$ as analogous to standard implication $\rightarrow$. We will see below, though, that the linear versions of these connectives have properties that are different from their classical counterparts in just the ways that we will need. This expression can then be seen to carry the following specification: given an agent X and a theme Y, we can deduce that the meaning of a sentence containing *likhaa* 'write' is $write(X, Y)$.

Since we will not be concerned with the details of the internal semantics of noun phrases, we will make the simplifying assumption that the meaning of a noun phrase like *xat* 'letter' is *letter*:

(17) xat N ($\uparrow$ PRED) = 'LETTER'
$\uparrow_\sigma \rightsquigarrow letter$

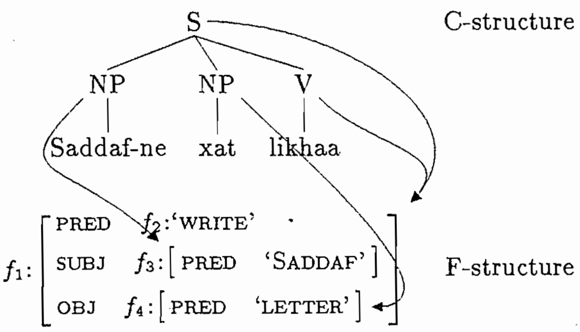
For discussion of the semantics of noun phrases within this framework, including a treatment of quantification, see Dalrymple, Lamping, Pereira, and Saraswat (1993).

The lexical entry for the ergative particle specifies that it attaches only to subjects (Butt and King, 1991), and makes no semantic contribution. Nominative (uncasemarked) noun phrases can only bear the grammatical functions of subject or object. From casemarking information, then, we know that *Saddaf* bears the grammatical function SUBJ in this sentence, and that *xat* 'letter' must be either the subject or the object. Since the presence of two SUBJs is disallowed, we can conclude that *xat* 'letter' is the OBJ of the syntactic predicate WRITE.

The c-structure, f-structure, and semantic information for example 18 is given in 19:

(18) Saddaf-ne xat likhaa
Saddaf-ERG letter-NOM wrote
'Saddaf wrote a letter.'

(19)



$$f_{3\sigma} \rightsquigarrow \mathit{Saddaf}$$ Semantic information
$$f_{4\sigma} \rightsquigarrow \mathit{letter}$$
$$\forall X, Y.\ \mathit{agent}(f_{2\sigma}, X) \otimes \mathit{theme}(f_{2\sigma}, Y) \multimap f_{1\sigma} \rightsquigarrow \mathit{write}(X, Y)$$

Note that the f-structure contains only information deducible from the phrase structure configuration and lexical properties of the words involved. In this case, the phrase structure rules and lexical entries for casemarkers unambiguously determine the grammatical functions borne by the noun phrases *Saddaf-ne* and *xat* in the sentence.

Arrows leading from the c-structure to the f-structure represent the projection function between nodes of the c-structure and f-structures; for example, the NP dominating *Saddaf* corresponds to the f-structure labeled f_3, whose PRED is 'SADDAF'. The semantic projections are, as above, indicated by a subscript σ. Thus the f-structure labeled f_3 has $f_{3\sigma}$ as its semantic projection, and *Saddaf* is recorded as its meaning.

The verb *likhaa* 'write' requires an agent X and a theme Y; providing these arguments allows the deduction that the meaning of the f-structure for the sentence, labeled f_1, is $\mathit{write}(X, Y)$. What is needed is a rule linking these thematic roles to the proper grammatical functions. This rule will specify that the subject expresses an argument with the thematic role of agent, and the object expresses a theme. Given this information, *Saddaf* can be identified as the subject/agent of *likhaa* 'write', and *xat* 'letter' can be identified as the object/theme.

The following *mapping rule* accomplishes this:[3]

$$!(\forall f, X, Y.\ ((f\ \text{SUBJ})_\sigma \rightsquigarrow X) \otimes ((f\ \text{OBJ})_\sigma \rightsquigarrow Y) \\ \multimap \mathit{agent}((f\ \text{PRED})_\sigma, X) \otimes \mathit{theme}((f\ \text{PRED})_\sigma, Y))$$

This rule associates subjects with agents, and objects with themes. It states that for all f-structures f, if the SUBJ of f is X and the OBJ of f is Y, we can conclude that X is the f-structure's PRED's agent, and Y is the f-structure's PRED's theme.

Recall that from casemarking and configurational information, we know that *Saddaf* must be the subject of the sentence and that *xat* must be the object. If instead the grammatical functions borne by these elements were underspecified and multiple possibilities existed, we would select one alternative at this point, and see whether we can produce a successful derivation of the meaning of the sentence by making this choice. In the case at hand, only one selection is possible, and we choose the mapping rule given above.

Instantiating the mapping rule involves replacing the universally-quantified f by f_1. Given this choice and our knowledge that f_1's SUBJ is f_3, we can replace (f SUBJ) by f_3, and similarly (f OBJ) by f_4. This produces the following instantiated mapping rule:

$$(\forall X, Y.\ (f_{3\sigma} \rightsquigarrow X) \otimes (f_{4\sigma} \rightsquigarrow Y) \multimap \mathit{agent}(f_{2\sigma}, X) \otimes \mathit{theme}(f_{2\sigma}, Y))$$

From the scaffolding provided by the lexical entries and the mapping rules, we look for a logical deduction of a meaning for the sentence. The deduction proceeds as follows, with

[3] The notation '!' preceding this rule should be ignored for now.

premises labeled in boldface:

Saddaf: $(f_{3\sigma} \rightsquigarrow Saddaf)$
letter : $(f_{4\sigma} \rightsquigarrow letter)$
mapping : $(\forall X, Y.\ (f_{3\sigma} \rightsquigarrow X) \otimes (f_{4\sigma} \rightsquigarrow Y) \multimap agent(f_{2\sigma}, X) \otimes theme(f_{2\sigma}, Y))$
write : $(\forall X, Y.\ agent(f_{2\sigma}, X) \otimes theme(f_{2\sigma}, Y) \multimap f_{1\sigma} \rightsquigarrow write(X, Y))$

Saddaf $\otimes$ **letter** $\otimes$ **mapping** $\otimes$ **write** (Premises.)

$\vdash agent(f_{2\sigma}, Saddaf) \otimes theme(f_{2\sigma}, letter) \otimes$ **write** (Universal Instantiation (UI), Modus Ponens.)

$\vdash f_{1\sigma} \rightsquigarrow write(Saddaf, letter)$ (UI, Modus Ponens.)

Analyzing *Saddaf* as the subject and *xat* 'letter' as the object, and choosing the mapping rule that specifies subjects as agents and objects as themes, enables a well-formed derivation of the meaning of this sentence.

The use of linear logic instead of classical logic as the 'glue language' provides additional advantages, since it allows us to capture the intuition that lexical items and phrases contribute uniquely to the meaning of a sentence. As noted by Klein and Sag (1985, page 172):

> Translation rules in Montague semantics have the property that the translation of each component of a complex expression occurs exactly once in the translation of the whole. ... That is to say, we do not want the set S [of semantic representations of a phrase] to contain *all* meaningful expressions of IL which can be built up from the elements of S, but only those which use each element exactly once.

Similar observations underlie the work of Lambek (1958) on categorial grammars and the recent work of van Benthem (1991) and others on dynamic logics.

It is this 'resource-conscious' property of natural language semantics—a meaning is used once and once only in a semantic derivation—that linear logic allows us to capture. The basic insight underlying linear logic is to treat logical formulas as finite *resources*, which are consumed in the process of deduction. This gives rise to a notion of *linear implication* $\multimap$ which is resource-conscious: the formula $A \multimap B$ can be thought of as an action that can *consume* (one copy of) A to *produce* (one copy of) B. Thus, the formula $A \otimes (A \multimap B)$ linearly implies B — but it does not imply $A \otimes B$ (because the deduction consumes A) or $(A \multimap B) \otimes B$ (because the linear implication is also consumed in doing the deduction). The resource consciousness not only disallows arbitrary duplication of formulas, but also arbitrary deletion of formulas. This causes the notion of conjunction we use ($\otimes$) to be sensitive to the multiplicity of formulas: $A \otimes A$ is not equivalent to A (the former has two copies of the formula A). For example, the formula $A \otimes A \otimes (A \multimap B)$ does linearly imply $A \otimes B$ (there is still one A left over) — but does not linearly imply B (there must still be one A present). Thus, every formula of the premises is used once and only once in reaching the conclusion, reflecting the resource-consciousness of natural language semantics. Finally, linear logic has an *of course* connective ! which turns off accounting for its formula. That is, $!A$ linearly implies an arbitrary number copies of A, including none. We use this connective on the background theory of mapping rules to indicate that they are not subject to accounting; they can be used as often or seldom as necessary.

The use of linear logic enables a clean semantic definition of *completeness* and *coherence*.[4] In the present setting, the feature structure f corresponding to the utterance is associated with the ($\otimes$) conjunction ϕ of all the formulas associated with the lexical items in the

[4] 'An f-structure is *locally complete* if and only if it contains all the governable grammatical functions that its predicate governs. An f-structure is *complete* if and only if all its subsidiary f-structures are locally complete. An f-structure is *locally coherent* if and only if all the governable grammatical functions that it contains are governed by a local predicate. An f-structure is *coherent* if and only if all its subsidiary f-structures are locally coherent.' (Kaplan and Bresnan, 1982, pages 211–212)

utterance.[5] The conjunction is said to be *complete* and *coherent* iff

$$Th \vdash \phi \multimap f_\sigma \rightsquigarrow t$$

(for some term t), where Th is the background theory containing, e.g., the mapping rules. Each t is to be thought of as a valid meaning of the sentence. This guarantees that the entries are used exactly once in building up the denotation of the utterance: no syntactic or semantic requirements may be left unfulfilled, and no meaning may remain unused.

This requirement is what ensures the selection of the correct mapping rule. In the derivation of the meaning of example 18, a different mapping rule might have been incorrectly chosen, one which relates experiencers to subjects, and themes to objects:

$$!(\forall f, X, Y.\ ((f\ \textsc{subj})_\sigma \rightsquigarrow X) \otimes ((f\ \textsc{obj})_\sigma \rightsquigarrow Y) \\ \multimap experiencer((f\ \textsc{pred})_\sigma, X) \otimes theme((f\ \textsc{pred})_\sigma, Y))$$

The use of this mapping rule would not allow a well-formed derivation. First, the meaning for a clause containing the verb *likhaa* can only be obtained in the presence of an agent, and this mapping rule does not provide one; the result is *incompleteness*, since all requirements must be fulfilled for a derivation to be successful. Second, this rule asserts the presence of an experiencer *Saddaf* which is not consumed during the derivation; the result is *inconsistency*, since the premises of the sentence do not lead to a single meaning associated with the sentence, with no premises remaining.

Alternatively, a mapping rule such as the following might have been incorrectly selected:

$$!(\forall f, X, Y.\ ((f\ \textsc{obl}_{agent})_\sigma \rightsquigarrow X) \otimes ((f\ \textsc{subj})_\sigma \rightsquigarrow Y) \\ \multimap agent((f\ \textsc{pred})_\sigma, X) \otimes theme((f\ \textsc{pred})_\sigma, Y))$$

This rule relates themes to subjects, and requires the agent to be realized as an oblique $\textsc{obl}_{agent}$ phrase. Again, a well-formed derivation would not be possible, since the rule allows the derivation to proceed only in the presence of an $\textsc{obl}_{agent}$ phrase, and such a phrase is not present. In this and in the above case, the interaction of the mapping rules with completeness and coherence constraints ensures that no other derivation produces a well-formed output.

Like other approaches that use mapping rules, we assume that their function is to relate thematic/semantic roles to grammatical functions. In other approaches, mapping rules are often thought of as taking thematic roles as their premises and producing grammatical functions as their conclusion (in other words, as mapping *from* thematic roles *to* grammatical functions). Our approach requires the opposite directionality. Since we treat arguments of verbs as resources that must be consumed during the derivation, and since mapping rules are seen as a mediating step in this process, mapping rules must produce the thematic roles that verbs require by consuming the grammatical functions in the f-structure. That is, we map *from* grammatical functions *to* thematic roles. The function of mapping rules in our analysis can be schematized in the following way:

(20) Grammatical functions $\xrightarrow{\text{mapping rules}}$ Thematic roles $\xrightarrow{\text{verb meanings}}$ Sentence meanings

This conception of mapping rules fits in cleanly with our overall framework, and allows mapping rules to participate dynamically in the derivation of sentence meanings in a clear and well-defined manner.

In the statement of particular mapping rules, thematic/semantic information may be specified as we have given it, as assertions about roles such as *agent* and *theme*. Alternatively, such information can be specified in terms of structures of Conceptual Semantics (Jackendoff, 1990), as in Butt's approach (Butt, 1993a), in terms of Proto-Roles (Dowty,

[5] The words in the utterance may, of course, be lexically ambiguous, and in that case there may be more than one such ϕ.

1991), as in Alsina's approach (Alsina, 1993a), or by some other means. Grammatical function information can be specified in terms of grammatical function labels such as SUBJ and OBJ, as is done here; alternatively, such information can be given in terms of a feature decomposition cross-classifying grammatical functions (Levin, 1986; Bresnan and Kanerva, 1989; Alsina, 1993a). Our claims relate to the formal properties of mapping rules, and thus our approach could be adopted as a part of any one of a number of specific approaches to mapping theory.[6] An outstanding research question is whether the formal apparatus we use for encoding mapping rules is adequate to capture the full range of linguistic generalizations encountered in natural language; in future work, we plan to integrate our approach with the findings of other researchers in an attempt to determine this.

5 Complex predicates, mapping rules, and deduction

We are now ready to consider how our approach applies to complex predicates. Given an example such as 5, repeated here, our task is to explain how the two verbs *likhne* 'write' and *diyaa* 'permit' combine to license a single syntactic argument structure, and to ensure that the syntactic representation for this sentence is related in the proper way to its (complex) meaning:

(21) Anjum-ne Saddaf-ko xat likhne diyaa
Anjum-ERG Saddaf-DAT letter-NOM write-INF-OBL let
'Anjum let Saddaf write a letter.'

For conciseness, we do not specify the lexical entries for the casemarkers or the phrase structure rules, but note simply that:

- Casemarking on *Anjum* determines that it is the SUBJ
- (Lack of) casemarking on *xat* determines that it is the SUBJ or OBJ; since only one SUBJ is allowed, it must be the OBJ
- Casemarking on *Saddaf* determines that it is the SUBJ, OBJ, or OBJ2; since multiple SUBJs and OBJs are disallowed, it must be the OBJ2

The f-structure for this example is:

(22)
$$f_1: \begin{bmatrix} \text{PRED} & f_2: \text{`PERMIT}\langle\text{WRITE}\rangle\text{'} \\ \text{SUBJ} & f_3: [\text{PRED} \quad \text{`ANJUM'}] \\ \text{OBJ2} & f_4: [\text{PRED} \quad \text{`SADDAF'}] \\ \text{OBJ} & f_5: [\text{PRED} \quad \text{`LETTER'}] \end{bmatrix}$$

The lexical entries that give rise to this f-structure are:

[6]Note, then, that we intend no claims about the correctness of the specific details of the mapping rules presented above; rather, our claim is that mapping rules should be of the general form we have illustrated, specifying possible relations between thematic roles and grammatical functions. In particular, no theoretical significance should be attached to the choice of thematic role labels used here; for the verb *write*, for example, labels such as 'writer' and 'written-thing' would do as well.

Anjum NP $(\uparrow \text{PRED}) = \text{`ANJUM'}$
$\uparrow_\sigma \rightsquigarrow Anjum$

Saddaf NP $(\uparrow \text{PRED}) = \text{`SADDAF'}$
$\uparrow_\sigma \rightsquigarrow Saddaf$

xat N $(\uparrow \text{PRED}) = \text{`LETTER'}$
$\uparrow_\sigma \rightsquigarrow letter$

likhne V $(\uparrow \text{PRED}) = \text{`WRITE'}$
$\forall X, Y.\ (agent((\uparrow \text{PRED})_\sigma, X) \otimes theme((\uparrow \text{PRED})_\sigma, Y)) \multimap (\uparrow_\sigma \rightsquigarrow write(X, Y))$

diyaa V $\forall X, P.\ (permitter((\uparrow \text{PRED})_\sigma, X) \otimes (\uparrow_\sigma \rightsquigarrow P)) \multimap (\uparrow_\sigma \rightsquigarrow permit(X, P))$

It is the permissive verb *diyaa* that allows a complex predicate to form and associates the correct meaning with it.

We assume that 'light verbs' such as permissive *diyaa* act to *consume* the meaning of the original verb and its arguments, *producing* a new permissive meaning and requiring an additional argument, the permitter:

$$\forall X, P.\ (permitter((\uparrow \text{PRED})_\sigma, X) \otimes \uparrow_\sigma \rightsquigarrow P) \multimap (\uparrow_\sigma \rightsquigarrow permit(X, P))$$

In the case at hand, P is the meaning of the f-structure labeled f_1 above. This is the meaning introduced by the verb *likhne* 'write'. This use of verb *diyaa* 'permit' requires:

- an argument X with the thematic role of permitter
- a meaning P for f_1

Given a permitter and a meaning for f_1, *diyaa* consumes these meanings and produces a new meaning for f_1: $permit(X, P)$. This ability to consume a preliminary meaning for an f-structure and replace it by a new one follows from the resource-oriented nature of linear logic.

The following mapping rule **mapping2** associates the array of thematic roles associated with the combination of the verb *likhne* 'write' and *diyaa* 'permit' with the grammatical functions SUBJ, OBJ, and OBJ2:

$$!(\forall f, X, Y, Z.\ ((f\ \text{SUBJ})_\sigma \rightsquigarrow X) \otimes ((f\ \text{OBJ})_\sigma \rightsquigarrow Y) \otimes ((f\ \text{OBJ2})_\sigma \rightsquigarrow Z)$$
$$\multimap permitter((f\ \text{PRED})_\sigma, X) \otimes agent((f\ \text{PRED})_\sigma, Z) \otimes theme((f\ \text{PRED})_\sigma, Y))$$

Importantly, this mapping rule is no different in nature from mapping rules that apply in simpler cases. All mapping rules serve to associate a set of grammatical functions with an array of thematic roles, whether the array of thematic roles involved is lexically associated with a single verb or arises by the combination of two or more verbs into a complex predicate. Thus, the monoclausality of complex predicates follows from the availability of the appropriate mapping principles in a language; no other special rules are necessary.

The derivation proceeds as follows:

$$\mathbf{Anjum} : (f_{3\sigma} \rightsquigarrow Anjum)$$
$$\mathbf{Saddaf} : (f_{4\sigma} \rightsquigarrow Saddaf)$$
$$\mathbf{letter} : (f_{5\sigma} \rightsquigarrow letter)$$
$$\mathbf{mapping2} : (\forall X, Y, Z.\ (f_{3\sigma} \rightsquigarrow X) \otimes (f_{4\sigma} \rightsquigarrow Y) \otimes (f_{5\sigma} \rightsquigarrow Z) \multimap permitter(f_{2\sigma}, X) \otimes agent(f_{2\sigma}, Y) \otimes theme(f_{2\sigma}, Z))$$
$$\mathbf{write} : (\forall X, Y.\ agent(f_{2\sigma}, X) \otimes theme(f_{2\sigma}, Y) \multimap f_{1\sigma} \rightsquigarrow write(X, Y))$$
$$\mathbf{permit} : (\forall X, P.\ (permitter(f_{2\sigma}, X) \otimes f_{1\sigma} \rightsquigarrow P) \multimap (f_{1\sigma} \rightsquigarrow permit(X, P))$$

$$\mathbf{Saddaf} \otimes \mathbf{Anjum} \otimes \mathbf{letter} \otimes \mathbf{mapping2} \otimes \mathbf{write} \otimes \mathbf{permit} \qquad \text{(Premises.)}$$

(1) $\vdash permitter(f_{2\sigma}, Anjum) \otimes agent(f_{2\sigma}, Saddaf) \otimes theme(f_{2\sigma}, letter) \otimes \mathbf{write} \otimes \mathbf{permit}$

(2) $\vdash permitter(f_{2\sigma}, Anjum) \otimes (f_{1\sigma} \rightsquigarrow write(Saddaf, letter)) \otimes \mathbf{permit}$

(3) $\vdash f_{1\sigma} \rightsquigarrow permit(Anjum, (write(Saddaf, letter))$

The premises of the derivation are, as above, information given by lexical entries and the relevant mapping rule **mapping2**. By means of mapping rule **mapping2**, information about the possible array of thematic roles required by the complex predicate *permit-write* can be derived; this step, represented in (1), uses Universal Instantiation and Modus Ponens.

In (2), a (preliminary) meaning for f-structure f_1, *write(Saddaf, letter)*, is derived by Universal Instantiation and Modus Ponens. At this point, the requirements imposed by *diyaa* 'permit', labeled **permit**, are met: a permitter (*Anjum*) is present, and a complete meaning for f-structure f_1 has been produced. These meanings can be consumed, and a new meaning produced, as represented in (3); this provides the solution to the problem of specifying an appropriate complex semantic representation with a complex predicate, since the meaning of the verb *likhne* 'write' appears as an argument of the verb *diyaa* 'permit' in the resulting semantic representation. This meaning is the only one deducible from the premises, since completeness and coherence constraints are met only when all requirements are fulfilled and no extra information remains. The final step provides the only complete and coherent meaning for the utterance.

On our approach, the treatment of complex predicates is analogous to the treatment of simpler constructions. The availability of mapping rules allows the grammatical functions associated with the arguments of complex predicates to be determined straightforwardly, and the use of linear logic associates the correct semantic representation with complex predicates. We have developed a theorem prover which efficiently handles the linear deductions necessary for the treatment of complex predicates; the Appendix discusses how it accomplishes this.

Complex predicates are more problematic for other approaches; we now turn to a consideration of these issues.

6 The syntactic monoclausality problem

Many syntactic frameworks implicitly or explicitly assume that the array of grammatical functions licensed by a syntactic predicate is immutable and cannot be altered in the course of a derivation. Complex predicates seem to violate this assumption, raising the *syntactic monoclausality* problem: the problem of how two different verbs can combine to license a single array of grammatical functions.

In the context of LFG, this problem arises because of certain assumptions about the nature and function of semantic forms, the values of the attribute PRED. Kaplan and Bresnan (1982) assume PRED values for verbs of the following form:

(23) 'write< (↑ SUBJ), (↑ OBJ) >'
agent theme

According to Kaplan and Bresnan (1982), these semantic forms can be regarded as encoding four types of information:

1. Specification of the semantic relation

2. Mapping of grammatical functions to semantic roles
3. Subcategorization information (the governed grammatical functions)
4. Instantiation to indicate semantic distinctness (predicate uniqueness)

Encoding these kinds of information by means of a single formal device permits the syntactically relevant aspects of meaning to be confined to a single place in the f-structure. It also gives rise to the syntactic monoclausality problem, since it seems to imply that subcategorization information must be allowed to change in the course of a syntactic derivation: an Urdu verb in combination with a light verb such as permissive *diyaa* requires a different array of grammatical functions than it requires when it is used independently.[7]

Kaplan (personal communication) observes that the effect of our approach and the approach taken in much other recent LFG literature (Alsina 1993a, Butt 1993a, among others) is to treat the different kinds of information encoded by semantic forms with separate and independent mechanisms. Specification of the semantic relation is accomplished in the lexicon: a verb like 'write', for example, specifies that its meaning is $write(X, Y)$ when given an agent X and a theme Y. We assume that the mapping of grammatical functions to semantic roles, as well as the particular inventory of governed grammatical functions, are given by the mapping rules: the mapping rules might specify that for a verb like 'write', the subject bears the thematic role of agent and the object bears the role of theme.

Given these new mechanisms, complex predicates and simpler constructions are handled identically. If a rule exists to map a set of grammatical functions licensed by a complex predicate to a set of thematic roles, then the result will be a monoclausal syntactic structure, whether this array of thematic roles arose in association with a single lexical item or with multiple predicative elements. This aspect of the analysis of complex predicates is thus unproblematic for any approach which assumes that verbs are lexically associated not with an array of grammatical functions, but with an array of thematic roles; mapping rules will associate the appropriate array of grammatical functions with these thematic roles in all cases, for complex predicates as well as for simpler cases.

On these assumptions, only one function of the semantic form remains: instantiation to indicate semantic distinctness. On the classical LFG view, different PRED values are incompatible; this disallows the presence of multiple fillers of a single syntactic argument slot. For example, clitic doubling is disallowed in some languages because the clitic pronoun and the full noun phrase contribute incompatible PRED values, as in the following French example (Grimshaw, 1982):

(24) a. Je cherche Pierre
I am looking for Pierre
'I am looking for Pierre.'

b. Je le cherche
I him am looking for
'I am looking for him.'

c.*Je le cherche Pierre
I him am looking for Pierre

Note, however, that in the case of complex predicates the same situation seems to arise. Each of the verbs that make up the complex predicate construction would be expected to contribute a PRED with a unique value; for example 5, the two PRED values are PERMIT and WRITE.

We believe that the syntactic function of predicate uniqueness is an important one, and so we continue to permit semantic forms in the lexical entries of semantically contentful lexical

[7] If the number of times that a verb can combine with light verbs such as permissive *diyaa* is bounded, it is possible to enumerate the different alternatives in the lexicon. This is the approach taken by Kaplan and Wedekind (1993), to be discussed below.

items to specify the values of PRED attributes in the f-structure. However, we also believe in a necessary separation between syntax and semantics, so that *which particular* semantic form appears is irrelevant for syntactic purposes. That is, the syntax is concerned only with the presence or absence of a semantic form, and not with its internal structure. Thus, if the only remaining function of the PRED is to ensure predicate uniqueness, it would do as well to assume that the PRED value for a sentence with a complex predicate is contributed by the main verb (here, WRITE), and that the function of LET is to modify the argument structure but not to contribute to or change the PRED value of the construction. It is for this reason that we have not taken a position on how a semantic form such as 'PERMIT<WRITE>' is formed, since we believe that the particular syntactic shape of this predicate is not an issue.

7 The syntax-semantics nonisomorphy problem

Most linguistic theories that are concerned with issues of semantic interpretation provide some way of assigning a unique meaning to a syntactic representation; the *projection architecture* of LFG represents one way of attaining this goal. In the case of complex predicates the problem is particularly tricky: the *syntax-semantics nonisomorphy* problem is the problem of how a syntactically monoclausal structure containing two predicators can give rise recursively to a semantically complex representation and bear the proper relation to it. This problem and related problems have been discussed by Dalrymple et al. (1992) and, in the context of machine translation, by Kaplan, Netter, Wedekind, and Zaenen (1989).

The problem is evident in the approach of Kaplan (1987), which appeals to a projection relation from f-structures to semantic structures like the one assumed here. Unaugmented, this approach fails in the case of complex predicates. For a sentence such as 5, both permissive *diyaa* and the verb *likhne* 'write' are heads; the f-structure for both these verbs is the outermost f-structure, labeled f_1 below. However, the meanings of these two verbs are clearly different. The result is that the outermost f-structure comes to be associated with two different meanings, the meaning of *likhne* 'write' as well as the meaning of *diyaa* 'permit':

(25)
$$f_1: \left[\begin{array}{ll} \text{PRED} & \text{`PERMIT}\langle\text{WRITE}\rangle\text{'} \\ \text{SUBJ} & \left[\text{PRED} \quad \text{`ANJUM'}\right] \\ \text{OBJ2} & \left[\text{PRED} \quad \text{`SADDAF'}\right] \\ \text{OBJ} & \left[\text{PRED} \quad \text{`LETTER'}\right] \end{array}\right] \qquad permit(Anjum, write(Saddaf, letter))$$

The projection relating the f-structure to its meaning is thus no longer a function.

In this connection, further problems arise. A complex predicate may be embedded as a complement of a verb such as 'say', for example. Such verbs specify that the meaning of their complement must fill a certain semantic argument role. However, notice that on the view above the complex predicate has two meanings, one contributed by the light verb *diyaa* ($permit(Anjum, write(Saddaf, letter))$) and one contributed by *likhne* 'write' ($write(Saddaf, letter)$). Obviously, the meaning contributed by the light verb is the one that is the correct meaning for the clause, but it is unclear that any general method can be provided for determining this in such cases.

The use of a resource-oriented logic for semantic composition permits the association of the two different meanings with the outermost f-structure, but at different stages of semantic composition. Following is a portion of the semantic derivation of sentence 5:

$$\mathbf{permit} : (\forall X, P.\ permitter(f_{2\sigma}, X) \otimes f_{2\sigma} \rightsquigarrow P \multimap f_{1\sigma} \rightsquigarrow permit(X, P)$$

(1) $permitter(f_{2\sigma}, Anjum) \otimes \mathbf{permit} \otimes (f_{1\sigma} \rightsquigarrow write(Saddaf, letter))$

(2) $\vdash f_{1\sigma} \rightsquigarrow permit(Anjum, (write(Saddaf, letter))$

At stage (1), a preliminary meaning for f_1 has been derived. Recall that permissive *diyaa* requires the presence of a meaning for f_1 as well as a permitter; this is represented in the premise labeled **permit**. At stage (2), the preliminary meaning for f_1 is consumed and a new meaning for f_1 is introduced. At no stage of the derivation are two different meanings associated with the same f-structure; the meaning is always unique. The final result is that the correct meaning is associated with f_1, and the correct results will ensue if this f-structure is embedded as a complement. Given our assumptions, then, the syntax-semantics nonisomorphy that arises in the case of complex predicates is not problematic for us.

Our proposal may be contrasted with the proposal of Kaplan and Wedekind (1993), involving the use of the *restriction operator*. They define the restriction operator as follows:

(26) If f is an f-structure and a is an attribute:

$$f|a = f|_{\text{Dom}(f)-\{a\}} = \{< s, v > \in f | s \neq a\}$$

Intuitively, $f|a$ is the f-structure f with the attribute a and its value removed.

On their approach, all verbs are multiply ambiguous; the verb *likhne* 'write', for example, has two subcategorization frames, related by a lexical redundancy rule. In its independent use, 'write' requires a SUBJ and and OBJ; in its use as a part of a complex predicate, it requires an OBJ and an OBJ2. The meaning associated with 'write' in its independent use is, as expected, $write(X, Y)$, where X is the meaning of the SUBJ and Y is the meaning of the OBJ. In the use of 'write' as a part of a complex predicate, the restriction operator is used to define its meaning: Kaplan and Wedekind propose to use the restriction operator to encode the meaning of a *subpart* of the verb's f-structure.

In particular, the meaning they propose for 'write' as a part of a complex predicate is given by:[8]

(27) $\uparrow|_{\text{SUBJ}_\sigma} \rightsquigarrow write(X, Y)$

where X is the meaning of the OBJ2, and Y is the meaning of the OBJ.

This specifies the meaning of the f-structure for *likhne* 'write' without the SUBJ attribute and its value. The following is the entry for permissive *diyaa*:

(28) $\uparrow_\sigma \rightsquigarrow permit(X, Y)$

where X is the SUBJ, and Y is the meaning of $\uparrow|_{\text{SUBJ}}$.

The first semantic argument of *permit* is the meaning of the SUBJ. The second argument is the meaning of the f-structure with the SUBJ removed, which is just the meaning given by *likhne* 'write' in its use as a part of a complex predicate.

Broadly, their analysis differs from the one presented here in that the meaning of an f-structure for a complex predicate does not change in the course of the derivation; instead, the meaning of a part of the f-structure is treated differently from the meaning of the f-structure as a whole. This is an interesting difference, as it implies a very different view of the process of semantic composition and how it is constrained. It should be noted that Kaplan and Wedekind's treatment of complex predicates predicts that a verb can combine with only a bounded number of light verbs, and entails that a different lexical entry must exist for a verb that combines with one light verb, with two light verbs, and so on. Indeed, light verbs themselves must be ambiguous according to whether they are the only light verb in the sentence or whether other light verbs also appear. As Butt (1993a) points out, many different kinds of complex predicates exist in Urdu, and these complex predicates interact; it remains to be seen whether the interactions between these complex predicates can be handled entirely in the lexicon, as Kaplan and Wedekind's analysis requires.

[8] In the following examples, we have changed some details of Kaplan and Wedekind's analysis for easier comparison with our own analysis.

8 Conclusion

We have illustrated the capability of our framework to handle the syntax and semantics of complex predicates. The use of linear logic provides a simple treatment of the requirements of completeness and consistency. Further, our deduction framework enables us to use linear logic to state such operations in a formally well-defined manner.

9 Acknowledgments

We would like to thank Joan Bresnan, Miriam Butt, Ron Kaplan, John Maxwell, Fernando Pereira, and Annie Zaenen for helpful discussion and comments.

Appendix: Efficient selection of mapping rules

This appendix explains the implementation of the selection of mapping rules and shows how the linguistic constraints of completeness and coherence combine with linear logic to allow mapping rules to be selected very efficiently.

The linear logic atoms used in the formulas are all of the form $Channel_\sigma \rightsquigarrow Term$, e.g. $f2_\sigma \rightsquigarrow Bill$. The channel is the σ-projection of the f-structure; channels control how different semantic elements can combine to form the meaning of the sentence. The term specifies the meaning associated with the channel. Each linear logic atom acts as a *producer* or *consumer* on its channel. For example, in $\forall x(A_\sigma \rightsquigarrow T_1 \multimap B_\sigma \rightsquigarrow T_2)$ (where T_1 and T_2 are terms containing x), a term on channel A is consumed and a term on channel B is produced. If we also have an atom that produces on channel A, $A_\sigma \rightsquigarrow T_3$ (where T_3 is a term containing some S), and T_1 can be matched to T_3 so that T_3 is the same as T_1 with x replaced by S, i.e. $T_3 = T_1[S/x]$, then we can discharge the consumer of A in the implication and end up with $B_\sigma \rightsquigarrow T_2[S/x]$.

For completeness and coherence, all sub-meanings must contribute to the meaning of the sentence once and only once. This means that all linear logic formulas must be used in the derivation (linear logic itself ensures that they are not used more than once). Stated differently, all producers and consumers must be able to match up, except for one remaining producer on the final output channel. In the examples presented in the paper, this is the channel corresponding to the sentence's f-structure.

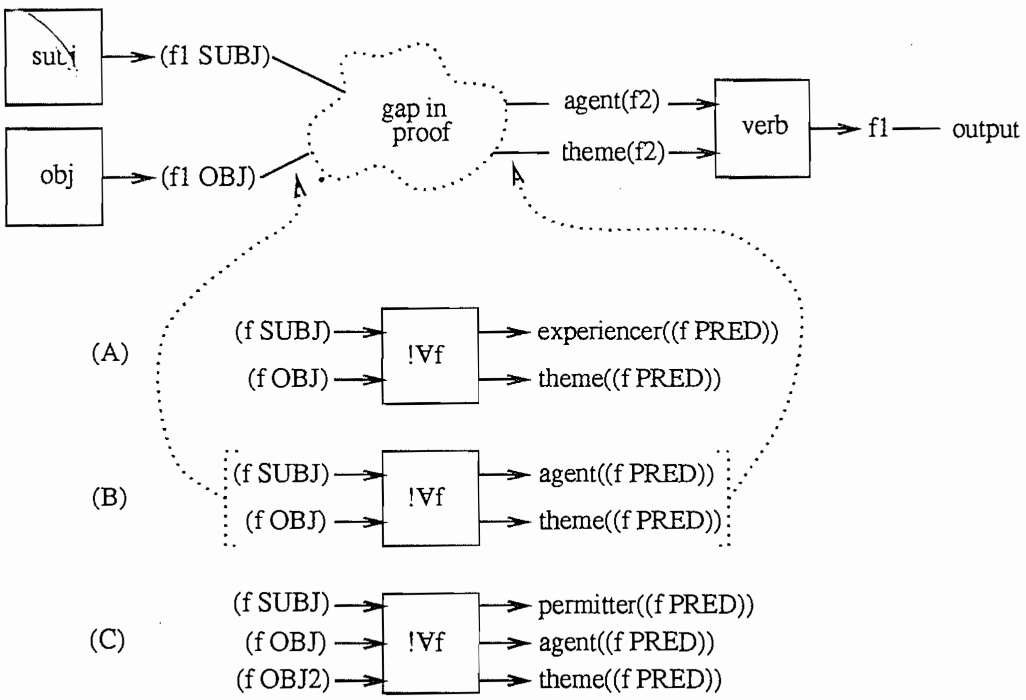


Figure 1: Selection of the Appropriate Mapping Rule

Mapping rules exist separate from the collections of formulas that contain meanings of sentences. They quantify over all f-structures, expecting the verb's f-structure. They are not subject to accounting; this is why they are preceded with the !, linear logic's of-course connective, which means they can be used arbitrarily many times, including none. Having to always consider mapping rules while solving the linear logic formulas is computationally expensive. At any step of the proof, all mapping rules must be checked in addition to the normal rules, and there could be many mapping rules. If we identify the necessary mapping rules before starting deduction, and proceed with only un-! linear logic rules, the search space is greatly reduced.

Before starting to prove a sentence meaning from our collection of linear logic rules, we can use completeness and coherence to determine which mapping rules will be necessary, instantiate the appropriate rule and proceed as if it were a normal formula. Formulas derived from the LFG f-structure give us producers of grammatical functions such as the subject and object of a verb, but lexical entry formulas are consumers of a verb's thematic roles such as the agent and theme. Mapping rules transform terms produced on grammatical function channels into terms produced on thematic role channels. The right mapping rule for a verb in a given set of linear logic formulas in our system is the one that consumes all of the verb's arguments' grammatical functions and produces all of the verb's thematic roles. A mapping rule that produces and consumes anything different will not lead to a complete and coherent derivation.

The figure shows how the correct mapping rule fills the gap between the grammatical functions and thematic roles. Linear logic formulas are represented as boxes with LHS atoms (consumers) as inputs and RHS atoms (producers) as outputs. The inputs and outputs of the boxes are labeled only with their channels. Mapping rule B is the only rule that applies to this verb, because it consumes exactly the grammatical functions produced by the sentence and produces exactly the thematic roles consumed by the verb. Mapping rules A and C will not lead to a complete and coherent derivation, because their producers and consumers are not the exact complement of the sentence's.

Given a set of linear logic formulas for the meaning of a sentence, it is easy to select the appropriate mapping rule or rules. Producers and consumers are tallied for each formula, and when the counts for all formulas are added together, the numbers of producers and consumers should balance out, except for the final output channel and channels for the grammatical functions and thematic roles. The unbalanced counts are sorted by verb. Each verb has a mapping rule selected for its unbalanced items. The right mapping rule's producers and consumers cancel out the extra consumers and producers associated with the verb.

References

Alsina, Alex. 1991. The monoclausality of causatives: Evidence from Romance. MS, Stanford University.

Alsina, Alex. 1993a. *Predicate Composition: A Theory of Syntactic Function Alternations.* Ph.D. thesis, Stanford University.

Alsina, Alex. 1993b. A theory of complex predicates: Evidence from causatives in Bantu and Romance. Paper presented at the Complex Predicates Workshop, CSLI, Stanford University.

Bresnan, Joan and Jonni M. Kanerva. 1989. Locative inversion in Chicheŵa: A case study of factorization in grammar. *Linguistic Inquiry*, 20(1):1–50. Also in E. Wehrli and T. Stowell, eds., Syntax and Semantics 26: Syntax and the Lexicon. New York: Academic Press.

Butt, Miriam. 1993a. *The Structure of Complex Predicates.* Ph.D. thesis, Stanford University. In preparation.

Butt, Miriam. 1993b. The structure of complex predicates in Urdu. Paper presented at the Complex Predicates Workshop, CSLI, Stanford University.

Butt, Miriam, Michio Isoda, and Peter Sells. 1990. Complex predicates in LFG. MS, Stanford University.

Butt, Miriam and Tracy H. King. 1991. Semantic case in Urdu. In Lise M. Dobrin, Lynn Nichols, and Rosa M. Rodriguez, editors, *Papers from the Twenty-Seventh Regional Meeting of the Chicago Linguistic Society*, pages 31–46, University of Chicago. Chicago Linguistic Society.

Dalrymple, Mary, Per-Kristian Halvorsen, Ronald M. Kaplan, Chris Manning, John Maxwell, Jürgen Wedekind, and Annie Zaenen. 1992. Relating projections. MS, Xerox PARC.

Dalrymple, Mary, John Lamping, Fernando Pereira, and Vijay Saraswat. 1993. A deductive account of quantification in LFG. Paper presented at the Second CSLI Workshop on Logic, Language, and Computation, Stanford University.

Dalrymple, Mary, John Lamping, and Vijay Saraswat. 1993. LFG semantics via constraints. In *Proceedings of the Sixth Meeting of the European ACL*, University of Utrecht, April. European Chapter of the Association for Computational Linguistics.

Dowty, David R. 1991. Thematic proto roles and argument selection. *Language*, 67(3):547–619.

Girard, J.-Y. 1987. Linear logic. *Theoretical Computer Science*, 45:1–102.

Grimshaw, Jane. 1982. Subcategorization and grammatical relations. In Annie Zaenen, editor, *Subjects and other subjects.* Indiana University Linguistics Club, Bloomington, IN, pages 35–56.

Halvorsen, Per-Kristian and Ronald M. Kaplan. 1988. Projections and semantic description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 1116–1122, Tokyo, Japan. Institute for New Generation Systems.

Jackendoff, Ray S. 1990. *Semantic Structures.* The MIT Press, Cambridge, MA.

Kaplan, Ronald M. 1987. Three seductions of computational psycholinguistics. In Peter Whitelock, Harold Somers, Paul Bennett, Rod Johnson, and Mary McGee Wood, editors, *Linguistic Theory and Computer Applications*. Academic Press, London, pages 149–188.

Kaplan, Ronald M. and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*. The MIT Press, Cambridge, MA, pages 173–281.

Kaplan, Ronald M., Klaus Netter, Jürgen Wedekind, and Annie Zaenen. 1989. Translation by structural correspondences. In *Proceedings of the Fourth Meeting of the European ACL*, pages 272–281, University of Manchester, April. European Chapter of the Association for Computational Linguistics.

Kaplan, Ronald M. and Jürgen Wedekind. 1993. Restriction and correspondence-based translation. In *Proceedings of the Sixth Meeting of the European ACL*, University of Utrecht, April. European Chapter of the Association for Computational Linguistics.

Klein, Ewan and Ivan A. Sag. 1985. Type-driven translation. *Linguistics and Philosophy*, 8:163–201.

Lambek, Joachim. 1958. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170.

Levin, Lori S. 1986. *Operations on lexical forms: Unaccusative rules in Germanic languages*. Ph.D. thesis, MIT.

Manning, Christopher D. 1992. Romance is so complex. Technical Report CSLI-92-168, Center for the Study of Language and Information, Stanford University.

Mohanan, Tara. 1993. Multidimensionality of representation: Noun-Verb complex predicates in Hindi. Paper presented at the Complex Predicates Workshop, CSLI, Stanford University.

Saraswat, Vijay A. 1993. A brief introduction to linear concurrent constraint programming. Technical report, Xerox Palo Alto Research Center, April.

Scedrov, Andre. 1993. A brief guide to linear logic. In G. Rozenberg and A. Salomaa, editors, *Current Trends in Theoretical Computer Science*. World Scientific Publishing Co. Revised and expanded version of the article originally appearing in *Bulletin of the European Assoc. for Theoretical Computer Science* 41, 1990.

van Benthem, Johan. 1991. *Language in Action: Categories, Lambdas and Dynamic Logic*. North-Holland, Amsterdam.