

The Syntax/Semantics Interface: Compositionality Issues

2nd East Asian Summer School in Language, Logic and Computation
Chongqing, China
August 27-31, 2012

Pauline Jacobson
Dept. of Cognitive, Linguistic, and Psychological Sciences
Brown University
Providence, RI, USA
pauline_jacobson@brown.edu

1. *Compositionality in its broadest sense:*

- meanings of larger expressions computed in some predictable way from the meanings of smaller expression which compose them
- But: actual content of this depends on: Just what is the object that is interpreted?
Put differently: how does the syntax of natural language (which proves expressions well-formed) interact with the semantics (which assigns expressions a meaning)?

2. *Three models of how the syntax and semantics of natural language interact*

1. Direct Compositionality (see, e.g., Richard Montague – “English as a Formal Language”, reprinted in R. Thomason (ed.), *Selected Papers of Richard Montague*, Yale University Press, 1974).

syntax: a system of rules which prove expressions well-formed (often proving one expressions well-formed on the basis of two or more other expressions)

semantics: works in tandem with this to provide an interpretation of each expression as it is built in the syntax

So: each linguistic expression a triple: <[sound], Category, [[meaning]]>
linguistic rules: take one or more triple as input and return a triple as output

Is this all?

Do the rules refer to “structure”?

In what ways can the input string(s) be modified to give the output?

- Strong Direct Compositionality:
Rules do nothing more than concatenate expressions
(hence equivalent to context free phrase structure rules)
 - *no reference to “structure” at all in the grammar*

- “structure” (e.g. trees) represent the way the rules work, and are only a convenience for the linguist; the grammar does not “see” structure

- Slightly weaker version: rules can concatenate expressions and also can also infix one expression into another (as in morphology)

Hence: any string must also have a little bit of structure: way to keep track of the “infixation point”

Various formalisms for this

e.g. C. Pollard *Generalized Phrase Structure Grammar, Head Grammars and Natural Language*, Ph.D. Dissertation, 1984) -

headed strings

A. Joshi et al. - divided strings

- Weak Direct Compositionality: inputs have structure, and rules can modify this in various ways (Montague, “The Proper Treatment of Quantification in Ordinary English”, reprinted in R. Thomason, (ed.), 1974) – substitution rules, deletion, etc.)
2. Generative Semantics LF (called “Deep Direct Compositionality” in Jacobson, “The (dis)organization of the grammar: 25 years” in *Linguistics and Philosophy*, 2002.

Generative Semantics: for work relevant to the points here, see especially works in the late 1960s through mid 1970s by G. Lakoff, J. McCawley, and P. Postal

- semantics is compositional, but what is interpreted by the semantics is different from the surface (pronounced) sentence
- rules such as phrase structure rules “build” Logical Forms (LFs)
- this could be coupled with direct compositional architecture as above
- *but* then a series of rule mapping this to surface structure

3. LF view in GB/Minimalism

- syntax “builds” structures – various models of just what the syntax can do
- these themselves are not interpreted
- these are mapped into another level of representation – LF – which is then compositionally interpreted

there are other conceivable and actual models – but these are representative; will not say much here about (2)

Goals of this course:

elucidate the views in (1) and (3) more

elucidate empirical phenomena (drawn only from English) that seem to present a challenge to Direct Compositionality

elucidate the kinds of tools that can be used in order to preserve DC

Overview of Cases to be considered

1. **Transitive Verbs**
2. **Quantified NPs in subject position – and interaction with conjunction**
3. **Conjunction of subsentential expressions: generalized Conjunction**
4. **“Non-constituent coordination”**
5. **Quantified NPs in object position**
6. **Coordination and “Binding”**
 A brief introduction to Variable Free semantics, and its connection to DC
7. **“Elliptical” phenomena: (e.g., short answers to questions) and, so-called “Binding Theory”**
 why these bear on Direct Compositionality and vice-versa
 Principle A, Bound variable connectivity
8. **The analysis of idioms**

Background:

assume basic semantic building blocks:

set of individuals - call that set e

set of truth values $\{1,0\}$ – call that set t

set of worlds - w

set of times – i

(will mainly ignore worlds and times and do semantics extensionally)

since worlds and times operate independently, use s for world time pairs

each linguistic expression has a meaning which is a member of one of these sets or (more usually) a member of a set of functions built from these sets

notation: $\langle e,t \rangle$ is the set of functions with domain e and codomain t

assume: each syntactic category corresponds to a unique semantic type

i.e., all expressions of the same category have meanings drawn from the same set (usually, some set of functions)

every S – denotes a truth value (actually – its *intension*: a function from worlds and times to truth values, hence of type $\langle s,t \rangle$)

extension: a truth value – hence, of type t

every NP – denotes an individual (extensionally) – i.e., semantic type e

VP - semantic type $\langle e,t \rangle$

Notation:

Given an expression α , meaning of α will be represented as $[[\alpha]]$ or α' (whichever is

more convenient)

(1) $S \rightarrow NP VP$; $[[S]] = [[VP]]([[NP]])$

that is the extensional rule: the full rule:

$$[[S]] = \lambda s[VP'(s)(NP'(s))]$$

in general, when two expressions combine, one has a meaning of type $\langle a, b \rangle$ and the other of type a , to give as value of the new expression an meaning of type b

This can be stated generally within a Categorical Grammar syntax; will not do this here

semantic composition of

(2) *Marie jumped*

$[[Marie]]$ = some individual, call it m

technically: a function from world/time pairs to individuals

assume it is a constant function

$[[jump(ed)]]$ some function (in any given world and at any time) of type $\langle e, t \rangle$

$[[Marie jumped]]$ (at a given world-time pair s) = $[[Maire]](s)([[jumped]](s))$

extensionalizing:

$[[jump]](m)$

CASE 1. The analysis of transitive verbs

what is the semantic type of a transitive verb, e.g., *love*

intransitive verbs: $\langle e, t \rangle$ - map individuals to truth values

transitive verbs:

first pass: follow the lead of first order logic

(3) *Marie likes Kelley* $L(m, k)$

so maps a set of order pairs of individuals to a truth value

type $\langle (e, e), t \rangle$

$[[Marie likes Kelley]] = [[likes]](m, k)$

But: evidence that the syntax puts together V and object to form a VP constituent

e.g. any two expressions of category X can conjoin to give an expression of category X

hence: $X \rightarrow X$ and X also $X \rightarrow X$ or X

(4) a. Marie ate fish and ran.

b. Marie ate fish or ran

ate fish has exactly the same distribution as *ran*, and can coordinate with *ran*

so: VP \rightarrow V NP

But if $[[\text{ate}]]$ of type $\langle (e,e),t \rangle$ then:

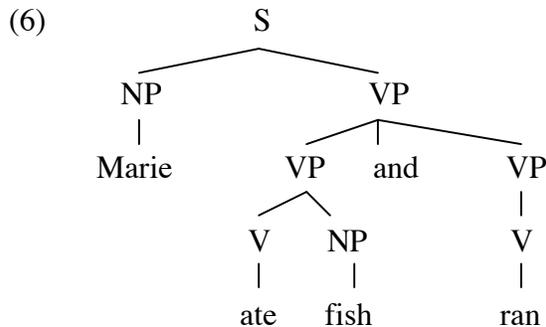
- (a) *ate fish* has no meaning; only *Marie ate fish* has a meaning
- (b) rule in (1) not applicable in composition of (3), only in composition of (2)
- (c) no direct compositional way to put it together

would need a new rule interpreting trees instead:

$$(5) \quad \left\| \begin{array}{c} \text{S} \\ \swarrow \quad \searrow \\ \text{NP}_1 \quad \text{VP} \\ \quad \swarrow \quad \searrow \\ \quad \text{V} \quad \text{NP}_2 \end{array} \right\| = [[V]](([[NP_1]], [[NP_2]]))$$

not direct compositional

But still gives no way to give a semantics for (4):



- would have to say that this “hiddenly” two sentences

- (7) a. Marie ate fish and Marie ran
- b. Marie ate fish or Marie ran.

see early “Conjunction Reduction” accounts in early Transformational grammar, or, map (6) into the two Ss at LF, and then compositionally interpret

strategies like this are common – but introduce more and more complexity into the system, also note:

- (8) Some student ate fish and ran. \neq

- (9) Some student ate fish and some student ran.
 (10) No student ate fish and ran. \neq
 (11) No student ate fish and no student ran.

are further strategies to solve this, but it introduces even more complexity into the system

But there is a very simple direct compositional solution: revise the original assumption about the semantic type of transitive verbs – this assumption made only because we mistakenly took the syntax of first order logic as the model for the syntax of English

“Curry” the transitive verb meaning:

of type $\langle e, \langle e, t \rangle \rangle$ - it takes an individual (the object) to give a function from individuals (the subject) to truth values

$VP \rightarrow V NP; [[VP]] = [[V]]([[NP]])$

$[[\text{ate fish}]] = \text{characteristic function of set of fish-eaters}$

(12) $VP_1 \rightarrow VP_2 \text{ and } VP_3; [[VP_1]] = \lambda x [[[VP_2]](x) \ \& \ [[VP_3]](x)]$

notation: for any function F which is the characteristic function of some set, let f_w stand for the set characterized by f
 for any set S, let S_F be the (total) function characterizing it
 (i.e., S_F maps each member of S to 1 and all else in its domain to 0)

then: (12) $VP_1 \rightarrow VP_2 \text{ and } VP_3; [[VP_1]]_S = [[VP_2]]_S \cap [[VP_3]]_S$

or rule = union

Rule in (12) not really a part of the grammar, since generally:

$X \rightarrow X \text{ and } X$

So the semantics should be stated more generally which will be done later; assume for now semantics of intersection for “and” and union for “or”

Moreover: so far have introduced *and* and *or* “syntactegorematically” – only via the rule, without listing it in the lexicon

Ultimately both are lexical items

and evidence that they combine with things one at a time, so their meaning will be Curry’ed

Case 2: Quantified NPs in Subject Position

- (13) a. Every student left the party.

- b. Some professor fell asleep.
- c. No professor fell asleep.

simple first order rendering:

- (14)
- a. $\forall x[\text{student}(x) \rightarrow \text{left party}(x)]$
 - b. $\exists x[\text{professor}'(x) \ \& \ \text{fell-asleep}(x)]$
 - c. $\sim \exists x[\text{professor}'(x) \ \& \ \text{fell-asleep}(x)]$

But - is this a reasonable approximation of the natural language composition?

- (a) requires sentence in (13) to be mapped into a Logical Form like this; hence additional rules of grammar to do this
 - (b) claims that *every student*, *some professor*, etc. - while well-formed syntactic expressions, have no meaning (their contribution is just in how they trigger the rules to map into the relevant Logical Form)
 - (c) *every* and *some* don't make the same contribution (connective is different)
 - (d) though these distribute syntactically exactly like ordinary NPs (*the professor*, *Carolyn*, etc.) they have a completely different semantic contribution
 - (e) Moreover, these conjoin with other quantified expressions – generalized conjunction syntactic rule would allow this, but again no way to put the meanings together without mapping it to two sentences:
- (15)
- a. Every student and no professor left the party.
 - b. Some student and every professor fell asleep
- (f) Moreover, these conjoin with “ordinary NPs” – such as proper names and definite NPs – same problem as above:
- (16)
- a. Carolyn and every professor left the party.
 - b. The smartest student and some professor fell asleep.

In fact, even two ordinary NPs can conjoin:

- (17) Cao and Zhang won diving medals.

no intersection - this case alone not conclusive, since coordination of two individual denoting NPs can be seen as a “plural individual” - much like a plural NP *the divers*

and in particular in a case like this where it can have a “collective” reading as in the (actual) synchronized diving scenario

But: also conjoins with *or* - no sensible way to view this as set union if only individuals

(18) Cao or Zhang won diving medals.

--> not directly compositional, and quite complex in the grammar

The classic solution from Montague, 1974- "Proper Treatment of Quantification in Ordinary English"

assume that these expressions do have a meaning

i.e., take the syntax seriously, and model the semantics to allow the syntax and semantics to work together in direct compositional fashion

what type of meaning? -

can't be an individual (on ordinary understanding of that)

can't be a set of individuals

one would be tempted to have *no dog* be \emptyset , but the *no dog* and *no cat* would have the same meaning

solution: (Montague, 1974 and others):

let these be sets of sets (generalized quantifiers)

or, equivalently, functions from functions from individuals to truth values to truth values

thus, functions in $\langle\langle e, t \rangle, t \rangle$

such that:

every student is the set of all sets which have [[student]] as subset

some professor is the set of all sets which have a non-empty intersection with [[professor]]

no professor is the set of all sets which have an empty intersection with [[professor]]

etc.

Determiners: *every*, *some*, *no* denote relations between sets

this solves (a) - (c) and (e) above

(a) no need for a mapping to a different representation

semantic combinatorics: *Some professor fell asleep* : [[fell asleep]] denotes (given a world and time) a set: - this in set of sets [[some dean]]

[[some professor]] set of sets with non-empty intersection with professors

hence: if [[fell asleep]] has non-empty intersection with [[professor]], we get the right truth conditions

Every student left the party says [[left the party]] set is in [[every student]] set the latter in turn is the set of sets with [[student]] as subset

hence: [[student]] is subset of [[left the party]]; hence get the right truth conditions

No professor fell asleep - [[fell asleep]] set in [[no professor]] set – so it is in the set of sets with empty intersection with [[professor]]; hence we get the right truth conditions

Thus: in these cases, VP-meaning is argument of subject

- (b) each quantified NP has a meaning
- (c) all determiners are relations between sets - *every, some, no, many* etc. all have the same basic semantic type
- (e) all of these quantified NPs denote sets of sets – so *and* intersects them as in the case of VPs

Take (15a): Every student and no professor left the party.

[[every student]] = set of sets with [[student]] as subset

[[no professor]] = set of sets with empty intersection with [[professor]]

intersection of these: = set of sets with [[student]] as subset and with empty intersection with [[professor]]

if [[left the party]] is in that intersection, it must have [[student]] as subset and have empty intersection with [[professor]]
and this is the right truth conditions

But, what about (d) and (f)? (centering on “ordinary NPs”)

first pass (Montague, 1974): ordinary NPs - which appear to denote individuals - can also be given meanings in which they denote sets of sets

let o be the individual we associate with the name *Barack Obama*

then, rather than that NP having o as its denotation, let [[Barack Obama]] be the set of sets containing o as member

in function terms: $\lambda P_{\langle e,t \rangle}[P(o)]$

i.e., maps any function P which is a function in $\langle e,t \rangle$ to the value that P assigns to the individual o

similarly for ordinary definite NPs such as *the smartest student*

second pass: A variant of this (roughly that of Partee and Rooth, 1983 - but generalized here

The general tool:

a linguistic expression is a triple: $\langle [\text{sound}], \text{Cat}, \text{meaning} \rangle$
 rules take one or more triple as input, return a single triple as output
builds in “direct compositionality”

unary rules: take one triple only as input

a special case: rules whose input and output is the same in terms of the phonology – semantics not (category possibly not too – depends on theory of syntactic categories)

hence: such rules “change” meaning

type shifting rules: are a special case of those - these change meaning and also the semantic type; may or may not change syntactic category (this depends in part on particulars of the theory); and do not change phonology.

Thus: let any expression α with phonology $[\alpha]$, category NP, and meaning (extension) be an individual a shift to one with the same phonology, category xxx (depends on one’s theory here) and meaning (extension) be the set of sets with α as member

Lexical meaning of proper names: type e – they denote individuals
 but can freely shift to generalized quantifier meaning

This can be generalized to allow any expression to “lift” -

For any set A and any set of functions in $\langle A, B \rangle$ let a be a member of A . Then $\text{lift}(a)$ is a function from functions in $\langle \langle A, B \rangle, B \rangle$ such that for any function f in $\langle A, B \rangle$, $\text{lift}(a)(f) = f(a)$.

A note on the “type shifting as a last resort” slogan.

- that slogan usually assumes a view whereby syntax computes representations which are then “sent” to the semantics for interpretation
- however, can think of this as a processing principle
 - listeners will generally compute meanings in the lowest (i.e., the lexical) type of the expressions involved. But they will revise their hypothesis and use the lifted meanings if:
 - this is the only way to put things together (as when an ordinary NP is conjoined with a generalized quantifier) or
 - when the lowest types meaning is not pragmatically sensible

Then: $[[\text{Barack Obama}]] = o$

but can lift to give $\lambda P[P(o)]$ (i.e., set of sets with o as member)

Then: solution to (d):

need to combine it with a theory of syntax, but basically: since every ordinary NP can also be a generalized quantifier, it follows that wherever generalized quantifiers can occur, so can ordinary NPs

The reverse does not follow; will return

Solution to (f): an ordinary proper name and an ordinary definite NP can also be recast as a set of sets
so it too can intersect with a generalized quantifier or with another definite NP

e.g.: $[[\text{Cao or Zhang}]] = \text{the union of set of sets with c as member and set of sets with z as member}$
(hence each such set has at least one as member)
hence right truth conditions for *Cao or Zhang won a diving medal*

Case 3 – Generalizing conjunction of subsentential expressionsj

VP Coordination (review from earlier):

(19) Carmen sang or danced.

very early transformational grammar:

interpretation must involve a level (LF) at which this is actually S coordination
key assumption: $[[\text{and}]]$ can only be defined for propositions

But: well-known that this not necessary - let $[[\text{and}]]$ for VP coordination be intersection of sets

Can extend intersection to the case of generalized quantifiers, since these also sets of sets

But coordination is more general - any two expressions of the same type can coordinate to give one of that type, even if they are not set-denoting

Example: Transitive verbs (as seen above, of type $\langle e, \langle e, t \rangle \rangle$)

(20) Carmen watered and fertilized the garden.

Solution - based on Partee and Rooth (1983) though with some revisions:

define $[[\text{and}]]$ recursively, such that it can conjoin any two expressions such that: (a) the type of each is the same, and (b) the type of each ultimately is a function into truth values

Note: syntax will be ignored here

Background pieces:

- assume that *and* and *or* both listed in the lexicon and that they are “Curry’ed” functions
- assume that lexical meanings are of type $\langle t, \langle t, t \rangle \rangle$ - i.e., they take one truth-value denoting expression, and then a second, to yield a truth-value denoting expression
Note: this ignores intensions; they really are of type $\langle \langle s, t \rangle, \langle \langle s, t \rangle, \langle s, t \rangle \rangle$

- Thus, lexical meaning of *and* is “Curry’ed” &

$\lambda p_{\{1,0\}}[\lambda q_{\{1,0\}}[q \ \& \ p]]$
(and similarly for *or*)

- The “Geach” rule (**g**):

Take any function f in $\langle a, b \rangle$. $\mathbf{g}(f)$ is a function in $\langle \langle c, a \rangle, \langle c, b \rangle \rangle$ such that, $\mathbf{g}(f)$ takes as argument a function h in $\langle c, a \rangle$ and then an x in c , and returns as value $f(h(x))$

An example:

NOT as a sentential operator: maps a S to true if S is false and vice-versa
(ignoring intensions: of type $\langle t, t \rangle$)

$\mathbf{g}(\text{NOT})$ - of type $\langle \langle e, t \rangle, \langle e, t \rangle \rangle$ - and hence a VP operator

takes a function h (the meaning of some VP) as input, and then takes an individual (the subject) x and returns true if NOT ($h(x)$) is true

Oversimplifying: English *didn’t* combines with VPs; can take its semantics to be $\mathbf{g}(\text{NOT})$

(21) Carmen didn’t run.

Note: This is oversimplifying; *not* and *n’t* really combine with auxiliary verbs, but further applications of \mathbf{g} can give the right semantics for these.

the crucial part of this tool: takes something which appears to operate on a large domain – such as the meaning of a sentence (a proposition) – and maps it into function which operates on a smaller domain (e.g., the meaning of a VP)

successive applications of this can keep getting something to combine with “smaller and smaller” bits – getting things like negation, tense – etc. to combine with verbs even though their ultimate semantic effect is on sentences

and hence allow for a direct compositional treatment of a variety of phenomena – where the semantic combinatorics reflects what the syntax and morphology are doing

intuitively: allows an expression which – at first glance – seems to semantically scope over something like a proposition (a full S meaning) - to combine with *an incomplete object* (say, an “incomplete” proposition – i.e. a function into propositions) and inherit the “incompleteness” - it “holds off” on some argument position

Key tool here: Generalize the \mathbf{g} rule:

Take any function f in $\langle a, \langle b, c \rangle \rangle$

Then $\mathbf{g}(f)$ is a function in $\langle \langle d, a \rangle, \langle \langle d, b \rangle, \langle d, c \rangle \rangle \rangle$ as follows: it takes a function h in $\langle d, a \rangle$, then a function j in $\langle d, b \rangle$ and then an x in d , and returns as value

$f(h(x))(j(x))$

this is the operation Partee and Rooth use for generalized conjunction

lexical meaning of *and* - of type $\langle t, \langle t, t \rangle \rangle$ (ignoring intensions)

this can connect Ss - call this AND

this takes meaning of one S and then of another and returns true iff both are true,

Apply the generalized “geach” rule once – and we can derive a new meaning of type

$\langle \langle e, t \rangle, \langle \langle e, t \rangle, \langle e, t \rangle \rangle \rangle$ which is the VP *and*

in set terms this amounts to Curry’ed intersection, in function terms:

$VP[[and]] = \lambda h [\lambda j [\lambda x [AND (h(x)) (j(x))]]]$

so it takes both VP meanings and then an individual (meaning of the subject) and returns

true iff each VP meaning applied to that individual yields true

(in set terms = intersection)

similarly for *or* (for the case of VPs, in set terms this will amount to U)

an interesting case: interaction with generalized quantifiers in subject position

Examples in (8) and (10) were unambiguous, and their meaning given as shown earlier

But (22) has a primary reading where $[[bark \text{ or } howl]] \varepsilon [[\text{every dog}]]$; this put together as follows:

(22) Every dog barks or howls.

$VP\text{-}or - \lambda P [\lambda Q [\lambda x [OR (P(x)) (Q(x))]]]$

$[[barks \text{ or } howls]] = \lambda x [OR (bark(x)) (howl(x))] = \lambda x [bark(x) \vee howl(x)]$

(set of individuals in union of $[[bark]]$ and $[[howl]]$)

for (22): this set in $[[\text{every dog}]]$ set:

this set has $[[dog]]$ as subset

However – this case also has a secondary reading (see, e.g., Partee and Rooth for discussion) (the similar cases in (8) and (10) do not):

(23) Every dog barks or howls (and I don’t remember which) =

(24) Every dog barks or every dog howls.

Does this mean that we need a rule mapping (23) to something like (24) to give this additional reading?

NO – this meaning is also derivable by the use of *lift*

if the VPs “lift” over generalized quantifier subjects, we get this

if one takes the view of processing cost of these operations: get the right prediction that wide scope *or* here is the less preferred meaning

Unsolved problem: Why is this secondary reading present in (22) but not in (8) or (10)? e.g. for (10) - this cannot mean “no student ate fish and no student ran”

back to transitive verbs:

applying the generalized “Geach” rule one more time gives the transitive verb connective

the new meaning will apply to two transitive verbs (each of type $\langle e, \langle e, t \rangle \rangle$ and then apply to the meaning of the object and then the subject and yield true iff the meaning of each transitive verb applied to the individual denoted by object and then individual denoted by subject yields true

INTERIM SUMMARY AND CONCLUSIONS SO FAR:

Transitive verbs, quantified NPs in subject position, and conjunction of subsentential expressions all look at first glance like they challenge Direct Compositionality

- interpretation of transitive verbs: If we follow the lead of first order logic, then VPs with transitive verbs have no meaning; whole structures (trees) need to be interpreted
- interpretation of *Every professor fell asleep*: If we follow the lead of first order logic, this is mapped into a very different representation which is then interpreted
- interpretation of *Carmen swam and dove, Yang or Zhang won a diving medal*, or *Carmen watered and fertilized the garden* - if we follow the lead of first order logic, “and” conjoins only propositions. So it looks like this must be two sentences. (Similar point for negation)

These are all simple and well-known cases (and in reality, very few semanticists argue that first order logic is the appropriate model for English semantics).

These all handled by very simple tools allowing “fancier” packages of meanings:

- Curry’ing the transitive verb meanings
- quantified NPs as sets of sets
- lift of ordinary individuals to sets of sets
- generalized conjunction; possibly by treating *and* as of type $\langle t, \langle t, t \rangle \rangle$ in lexicon and letting it shift by generalized **g** operation

Note also: **lift** and **g** are simple “natural” operations on functions

Hence, the illusion that these require a more complex level of representation in the syntax for interpretation comes from an overly naive view of the semantics

e.g., that quantificational looking like expressions in language translate into their first order counterparts, that *and* translates into $\&$, etc.

Case 4: “Non-constituent Conjunction”

(25) Carmen loves and Sally hates formal semantics.

traditional analysis: must also be “hiddenly”

(26) Carmen loves formal semantics and Sally hates formal semantics.

two reasons this assumed:

(a) syntax: assumes that *Carmen loves* is not a constituent; no way to put this together in the syntax

(b) semantics: no way to compose the meaning of *Carmen loves*: *loves* a transitive verb, of type $\langle e, \langle e, t \rangle \rangle$ and so expects to combine with its object first

Categorial Grammar approaches: can combine *Carmen* and *loves* in both the syntax and in the semantics - will not look at syntax here, it parallels the analysis in the semantics

simplest approach (see, e.g., Dowty 1987; Steedman 1987): function composition

lexical meaning of *Carmen* picks out the individual c - hence of type e
 can lift to a generalized quantifier - type $\langle \langle e, t \rangle, t \rangle$ $\lambda P[P(c)]$
 CG syntax: will mirror all of this in the syntax

then this can function compose with $[[\textit{loves}]]$ of type $\langle e, \langle e, t \rangle \rangle$

$$\lambda P[P(c)] \circ [[\textit{loves}]] = \lambda x[\lambda P[P(c)]([\textit{loves}]](x))] = \lambda x[[[\textit{loves}]](x)(c)]$$

(set of individuals that c loves)

same for *Sally hates*

generalized *and* conjoins these (intersection of these two sets)
 applies to $[[\textit{formal semantics}]]$

a slight variant: instead of direct function composition, do this in two steps:

Recall \mathbf{g} from earlier: it is a unary (Curry’ed) version of function composition

\mathbf{g} maps a function f of type $\langle a, b \rangle$ into a function g of type $\langle \langle c, a \rangle, \langle c, b \rangle \rangle$

which is such that when g applies to a function h of type $\langle c, a \rangle$, this is equivalent to having composed f and h

so – slight variant of function composition is to apply \mathbf{g} to lifted *Carmen*

$\langle\langle e, \langle e, t \rangle \rangle, \langle e, t \rangle \rangle$ - it then takes the transitive verb as argument, and returns a function of type $\langle e, t \rangle$

i.e. Carmen – lexical meaning = c

$\mathbf{lift}(\text{Carmen}) = \lambda P_{\langle e, t \rangle} [P(c)]$ (type $\langle\langle e, t \rangle, t \rangle$)

$\mathbf{g}(\mathbf{lift}(\text{Carmen}))$: type $\langle\langle e, \langle e, t \rangle \rangle, \langle e, t \rangle \rangle$

i.e., wants a two place relation to give an expression of type $\langle e, t \rangle$

$= \lambda R_{\langle e, \langle e, t \rangle \rangle} [\lambda x_e [\lambda P_{\langle e, t \rangle} [P(c)](R(x))]] = \lambda R_{\langle e, \langle e, t \rangle \rangle} [\lambda x_e [R(x)(c)]]$

this combines with *loves* to give $\lambda x [\text{loves}'(x)(c)]$

which is the function characterizing the set of things that c loves

Sally hates: same – ends up set of things that Sally hates

usual *and* intersects these

formal-semantics' is in that intersection

What this case tells us:

one argument for abstract levels has come from the assumptions that (a) all functions must get their arguments (the only mode of semantic combination is applying a function to an argument) and/or (b) functions themselves cannot be arguments

But these assumptions are not necessary, and follow from nothing

function composition approach denies (a)

Geach + application is compatible with (a), but denies (b)

(a) could be a constraint on *binary* combinatory rules, but as long as one allows for unary rules this doesn't matter

Given any binary operation $*$ which operates on a and b , this can be translated (“Curry’ing”) into a unary operator $**$ on a such that $**a(b) = *(a,b)$

A related observation: theories which allow for “silent operators” allow the same things: a “silent operator” can be recast as a unary rule, and vice-versa

claim that *Carmen loves* can never be a “constituent” need not be correct

Note: earlier saw assumption:

any expression of category X can combine with another expression of the same category to give an expression of category X

if one assumes this is the *only* kind of conjunction, these cases are counterexamples, and are called “Non-constituent conjunction”

But Categorical grammar gives the tools to make *Carmen loves* a constituent; have done only the semantics here but the syntax is similar

consequences: no need for movement analyses or traces for things like

- (27) a. Fine wine, Carmen loves.
 b. The wine that Carmen loves (best) was served at the banquet
 c. What does Carmen love?

“standard” view of, e.g. (27a):

level of representation *Carmen loves fine wine*

where *fine wine* moves

But no need for this; let *fine wine* combine directly with *Carmen loves* to its left

CASE 5: Quantified NPs in object (and other NP) positions

Generalized quantifier treatment of *every dog* etc. – solves problem of (a) giving them a meaning, and (b) predicting how they combine with VP meanings

hence: smooth semantic composition in subject position

but: These occur wherever ordinary NPs do

- (28) Carmen read every book.

Analysis in II of *every book* as a generalized quantifier (set of sets, or function of type $\langle\langle e,t \rangle, t \rangle$) does not immediately explain how this can combine with a transitive verbs

Assume *read* wants an individual in object position

hence, of type $\langle e, \langle e, t \rangle \rangle$

[[every book]] of type $\langle\langle e,t \rangle, t \rangle$

how do they combine?

compare this to the case of subject position – where the lowest obvious type to give *every husky* which gives it a coherent meaning is one which is compatible with VP-type meanings

strategies for this come in a great variety; hence will just informally sketch the debate and the types of strategies which have been used for these

Common solution (many textbooks have this): *this shows that we need a level of “LF” (logical form) which is interpreted – quantified NPs in object position are not really interpreted at that position by the semantics*

hence:

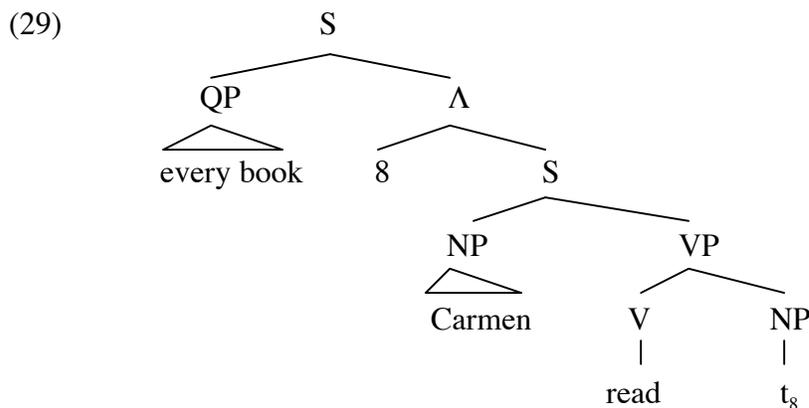
- syntax – computes (proves well-formed) not just strings but structured representations

- which are then mapped by a series of rules into other representations; the output of such a series of rules is the LF

Crucial operation “Quantifier Raising” (e.g., May, 1977) - which is a variant of earlier proposals of “Quantifier Lowering” - , Lakoff (1968), McCawley (1972) and many others

LFs then compositionally computed (from the bottom-up)

For the case in point – (28) mapped into:



NOTE the compositional details not so easy
 need to give a procedure for assigning values to the indexed traces, i.e., to variables
 --> variables (indexed) have meanings with respect to an assignment function

all meanings are with respect to an assignment function g
 each assignment function g is a function from variable names (call them integers) to individuals

$[[t_8]]^g = g(8)$
 $[[\text{Carmen read } t_8]]^g = [[\text{read}]](g(8))(c)$
 informally: c trained $g(8)$

so how do we do the “binding” - how does this representation help?

next step – step of interpreting Λ - is the semantics of “lambda abstraction” :

$[[[\lambda 8 \ S]]]^g =$ is a function mapping each individual x to the value of S on the assignment function g' , where g' is just like g except that 8 is assigned to x

the result is thus – with respect to any assignment function – a meaning of type $\langle e, t \rangle$ -
 this then taken as argument of the generalized quantifier

requires tool of variables and hence assignment functions –
 usual assumption that these needed anyway to interpret pronouns and traces
 but – are variable-free accounts of pronouns (see especially my work)
 and no need for traces in the general program(s) of, e.g., Generalized Phrase
 Structure Grammar (Gazdar, Klein, Pullum and Sag, 1985), Head Driven
 Prase Structure Grammar (e.g., Pollard and Sag, 1993), Categorical
 Grammar (much work, see e.g., Steedman 1989) and related theories

alternate strategies - in the direct compositional tradition

- Strategy 1: simply let *Carmen read* be put together by function composition, and then take that as argument of *every book*

this will work, but by itself not enough to get narrow scope reading of the quantifier in object position

(30) Some student read every book

if *some student* function composes with *read* we get:

$$\lambda x[\text{some-student}'(\text{read}'(x))] =$$

set of individuals such that the set that read them contains at least one student

take that as argument of *every book*:

the above set has book' as subset

this gives the reading:

$$\forall x[\text{book}'(x) \rightarrow \exists y[\text{student}'(y) \ \& \ \text{read}'(x)(y)]]$$

so this gives only wide scope object reading (for every book, some student read it)
 not only does this not give both readings, but object wide scope is the least preferred one

Note: The “Quantifier Raising” view gets both readings by allowing both quantifiers to scope out in either order

informally:

every book, x [some student, y [y read x]] or

some student, y [every book, x [y read x]]

This informal, but the technique spelled out above in the LF view generalizes to this case

- Strategy 2: Montague (1974):

have *read* and transitive verbs in general have meanings of type $\langle\langle\langle e,t \rangle, t \rangle, \langle e, t \rangle\rangle$

i.e. – they expect a GQ in object position

taking liberties with Montague's actual notation, let read'_* to be the ordinary 2-place relation between individuals

then $[[\text{read}]] = \lambda\mathcal{P}_{\langle\langle e,t \rangle, t \rangle} [\lambda y [\mathcal{P} (\lambda x [\text{read}'_*(x)(y)])]]$

informally: this takes a GQ object, then an individual subject y and returns true if the property of being ordinary-read by y is true of the gen. Q

(i.e., of the set of things ordinary-read by y is one of the sets in the GQ set)

if the GQ set is *every book* - this is the set of sets with $[[\text{book}]]$ as subset

so $[[\text{read every book}]]$ will map an individual y to true if the set of things trained by y has $[[\text{book}]]$ as a subset

BUT: this by itself gives *only* wide scope reading on subject!

(31) Some student read every book

The property $[[\text{read every book}]]$ will be argument of $[[\text{some student}]]$
which gives wide scope subject

Note: This is not really Montague's treatment, as it was supplemented with a Quantifying-In rule for the wide scope reading - the general Quantifying-In strategy is a direct-compositional version of the QR treatment shown above, but requires a more complex syntax; won't be elaborated here

Strategy 3:

Partee and Rooth (1983) and then generalized in Hendriks (1992):

Let lexical meaning of $[[\text{read}]]$ be of type $\langle e, \langle e, t \rangle \rangle$ - and have a unary "argument position lift" rule which maps any e -argument position (of a function whose final result is of type t) into a generalized quantifier argument position

take the mapping from read'_* to Montague- $[[\text{read}]]$ given above and make it a productive rule in the grammar (in a CG syntax there will be a corresponding syntactic shift) – and generalize it to apply to any argument position

full technical details:

Let f be any function of type $\langle a^{\rightarrow}, \langle e, \langle b^{\rightarrow}, t \rangle \rangle \rangle$ where a^{\rightarrow} means any number of argument positions (including 0) of any type, and similarly for b^{\rightarrow} . Then ASL ("argument slot lift")(f) =

$$\lambda x_{a^{\rightarrow}} [\lambda \mathcal{P}_{\langle\langle e,t \rangle, t \rangle} [\lambda y_{b^{\rightarrow}} [\mathcal{P} (\lambda z [f(x_{a^{\rightarrow}})(z)(y_{b^{\rightarrow}})])]]]]$$

notation here not standard: in prose:

f is a function which takes a sequence of n arguments (for $n \geq 0$) of any type, then an individual argument, then m arguments (for $m \geq 0$) of any type. $ASL(f)$ is a new function which takes n arguments of the same type as f does, then a generalized quantifier, and then m arguments of the same type as f does.

That describes its type. As to the actual function $ASL(f)$ is that function which takes some sequence (of length n) of arguments, then a generalized quantifier, and then a sequence (of length m) arguments, and returns true if and only if the generalized quantifier maps to true the function that maps an individual z to true if and only if f applied to the original n arguments and then the individual z and then the original m arguments yields true.

This is obviously very complex to state in prose! But it is perfectly well-defined, and the prose description

Above a bit informal, in that if we have a function of – for example – type $\langle e, \langle e, t \rangle \rangle$, ASL is not uniquely defined. It could apply to the first e -position or the second. We can distinguish those. Let ASL_1 mean a case where the last e -position is lifted; ASL_e mean the case where the second to last e -position is lifted, etc.

The Partee and Rooth revision of Montague's lexical meaning for *read* is just ASL_2 – it lifts the second to last position (the object position), so that the lexical meaning of *read* - of type $\langle e, \langle e, t \rangle \rangle$ lifts to a meaning of type $\langle \langle \langle e, t \rangle, t \rangle, \langle e, t \rangle \rangle$.

But the subject position could also be lifted.

Or both could be. And if so they could lift in different orders. And that gives the scope ambiguity!

(32) Some student read every book.

Call lexical item (of type $\langle e, \langle e, t \rangle \rangle$) $read_1$. (This is Montague's *read*'*)

- Subject wide scope: This can then map by ASL on object position, to give something of type $\langle GQ, \langle e, t \rangle \rangle$ - call this $read_2$
This is exactly Montague's "read".

This then can combine with the object to give *read every book* (with meaning as in Montague derivation) which then is argument of the subject.

--> student-set has non empty intersection with set of individuals x such that being read by $x \in$ [[every-book]]

(even more informally: student-set has non empty intersection with set of x 's such that every book was read by x)

- Object wide scope

$read_1$ maps to $read_3$ by ASL on subject position – hence of type $\langle e, \langle GQ, t \rangle \rangle$

$$= \lambda x [\lambda \mathcal{P} [\mathcal{P} (\lambda y [\text{read}'_1(x)(y)])]]$$

wants an individual x and then a set of sets and gives true if the set that read x is in the GQ set.

example: take the individual book *Crime and Punishment*

suppose above takes this as object and then *every student* as subject

true if the set of individuals that read *C&P* has student as subset (here, no particular advantage to this derivation, get the same thing by applying ordinary-read to *C&P* and then taking that as argument of *every student*)

But $read_3$ can also map by ASL on object position to give $read_4$ of type $\langle GQ, \langle GQ, t \rangle \rangle$ and this gives wide scope object reading formal details can be worked out – but informal intuition:

this will take a GQ like *every book* and then a GQ like *some student* and will return true if the property of being read by some student is a member of the GQ $[[\textit{every book}]]$

NOTE: yet another derivation is to lift $read_2$ (= Montague's *read*, of type $\langle GQ, \langle e, t \rangle \rangle$ to $read_5$ by ASL on subject position – to give also something of type $\langle GQ, \langle GQ, t \rangle \rangle$

This gives wide scope subject – and gives exactly what the first derivaton above gives

a standard argument for the QR approach:

- the apparatus needed for allowing GQs in object position in general immediately generalizes so as to give both scopes – once QR is possible, predicted that both GQs can undergo it, and in either order which gives both scopes

- BUT: exactly the same holds for the AS-L approach.

Once we allow one argument position to lift, straightforward to generalize that. If both can lift, we predict that the lifts can happen in either order (or just lift on the object position) and we get both scopes

Issues remain complex and murky:

- scopal interactions in embedded sentences make greater complications
- ASL approach by itself just manipulates argument “slots” of single verbs
- if wide scope possible in embedded Ss – can get some such effects with a combination of function composition + AL
- actual empirical facts complex and hence variety of unsettled issues

INTERIM SUMMARY AND CONCLUSIONS:

- basic facts about quantifiers in object position and quantifier scopes – can be handled with QR

the price: complex view of the organization of the grammar
requires introduction of variables and assignment functions

can be handled under direct compositionality (with, e.g., AL – though are other possibilities too whereby the quantified NP itself shifts)

- questions open, though, due to open questions about scope of embedded NPs

Case 6: More on Coordination: Interaction with “Binding”

Across-the-Board “Binding” in Right Node Raising

- (33) a. Every man_i loves but no man_j wants to marry his_{i/j} mother
b. Every man_i loves and no man_j marries his_{i/j} mother.

The “standard” (non-direct compositional) view

- (a) a “bound” pronoun must be c-commanded by its “binder”
A c-commands B if the mother of A dominates B
(b) moreover, no way to have one pronoun bound by two different things

therefore – even though ordinary Right Node Raising and other coordination cases can be handled without an abstract level, this case can’t

therefore (33b) (for example) would have to be mapped into an LF which is the same as the LF for:

- (34) Every man_i loves his_i mother and no man_j marries his_j mother.

- (a) c-command condition met
(b) two pronouns, two binders

- Claim here: neither (a) nor (b) is correct, in fact the notion of “binding” is not something that the grammar makes use of – all we should care about is getting the right truth conditions, and this can be done without (a) and without (b)

Background: what actually is meant by “binding” – how does the compositional semantics work for bound pronouns?

-- the “standard” view and one “variable-free” view

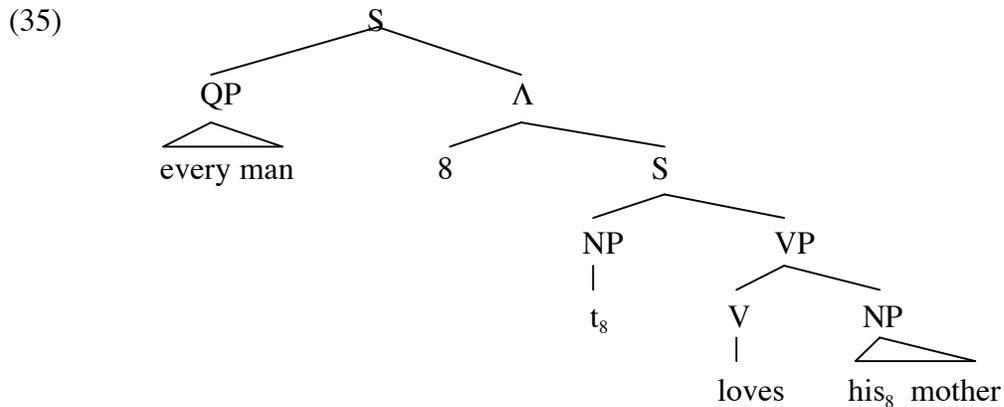
standard view: again – makes use of variables and hence assignment functions

all pronouns come in lexicon with an index (an integer)

all meanings relative to an assignment function g (a function from indices (integers) to individuals)

(actually: higher type variables – hence assignment functions more complex)

LF for first conjunct of (34) (and hence also for (33b)) - can use QR:



$$[[he_x]]^g = [[t_x]]^g = g(8)$$

(will treat *his* like *he*, ignoring contribution of genitive and of how this combines with the relational noun *mother*)

$$[[his_x \text{ mother}]]^g = \text{the-mother-of } g(8)$$

$$[[t_x \text{ loves } his_x \text{ mother}]]^g = g(8) \text{ loves the-mother-of } g(x)$$

next step; semantics of λ -abstraction, hence

$$[[\lambda]]^g = \text{function mapping each individual to proposition that that individual loves own mother-}$$

i.e., set of self's-mother-lovers

this then taken as argument of $[[\text{every man}]]^g$ (which is constant on all assignments: it maps each assignment function to set of sets with $[[\text{man}]]$ as subset}

Note: to say that *every man* “bind” the pronoun is misleading – “binding” of a pronoun or trace with index i is really the step at which the meaning is shifted from a non-constant function on the set of assignment functions which differ on the value assigned to i to a function which assigns all assignment functions that differ only on i the same value

hence “binding” can be defined at the “ λ -abstraction” step

to define a notion of “binding” holding between *every man* and *his* requires extra steps – not clear that there is any reason to do that

But: there are other ways to get the right truth conditions, making no use of “variables” and assignment functions in the semantics, and which thus work smoothly with direct compositionality

One implementation of a Variable Free Semantics (see especially Jacobson, “Towards a Variable Free Semantics”, *Linguistics and Philosophy*, 1999; Jacobson, “Paycheck Pronouns, i-within-i effects and Bach Peters sentences” in *Natural Language Semantics*, 2000; and several other papers of mine including those in *Proceedings of Semantics and Linguistic Theory 2, 4, 6, 8, 10, 12, 14*)

(other implementations include Szabolsci, 1993 in Sag and Szabolcsi, *Lexical Matters*, CSLI publications; Mark Hepple, dissertation, 1991; and recent work by Barker and Shan in *Linguistics and Philosophy* among others)

- no indices, no assignment functions
- any expression which contains a pronoun which is “unbound” within that expression is a function from individuals to something else (i.e., to the type it would have if that pronoun were replaced with a name)
two unbound pronouns: function from two individuals – etc.
- a pronoun itself: a function of type $\langle e, e \rangle$ - in particular $[[he]]$ is the identity function on individuals
- how does this combine with other things?

(36) He lost.

lost – a function of type $\langle e, t \rangle$ - wants an individual in subject position
but gets a function of type $\langle e, e \rangle$

let *lost* undergo the **g** rule (note: $\mathbf{g}(\text{lost})(he) = \text{lost} \circ he$)

$[[\text{lost}]]$ of type $\langle e, t \rangle$ maps to $\mathbf{g}([[\text{lost}]])$ of type $\langle \langle e, e \rangle, \langle e, t \rangle \rangle$

$[[he \text{ lost}]] = \lambda x[\text{lost}'(x)] (= \text{lost}')$

$[[his \text{ mother}]] = \lambda x[\text{the-mother-of}(x)]$

How effect “binding”?

unary rule on *loves*

Take any function f of type $\langle a, \langle e, t \rangle \rangle$ (for a any type) $-->$

$\mathbf{z}(f)$ is a function of type $\langle \langle e, a \rangle, \langle e, t \rangle \rangle$ such that $\mathbf{z}(f) = \lambda B_{\langle e, a \rangle}[\lambda x[f(B(x))(x)]]$

ordinary *loves* of type $\langle e, \langle e, t \rangle \rangle$

$z(\text{loves})$ of type $\langle\langle e, e \rangle, \langle e, t \rangle\rangle$ such that to z -love f is to be an x who loves $f(x)$

[[loves his mother]] – function characterizing set of individuals who are self’s-mother-lovers

(this exactly the same as the meaning of the expression labelled Λ in the LF/”standard theory” account above – except that (a) in that account we got there by a shift on the meaning of a proposition, whereas here we get there by a much more local shift on the meaning of the verb, and (b) in that account things still have meanings relative to assignment functions; here no assignment functions ever)

this then argument of [[every man]], as usual

The direct compositional account of the Across-the-Board Binding case:

(37) Every man loves and no man marries the Queen.

every man loves = every-man \circ loves = $\lambda x[\text{every-man}'(\text{loves}'(x))]$

here: every-man $\circ z(\text{loves}) = \lambda f[\text{every-man}'(z(\text{loves})(f))] =$
 $\lambda f[\text{every-man}'(\lambda x[x \text{ loves } f(x)])]$

set of functions that every man z -loves

no man marries: similarly – set of functions no man z -marries

every man loves and no man marries - as usual: intersection of these two sets

his mother – function mapping each individual into their mother

whole S : mother-function is in the intersection of those

Note on functional readings in general – these are widespread and are predicted here; they are automatic consequence of the mechanisms for “binding” in general

Functional questions: Groenendijk and Stokhof (1983), Engdahl (1986)

(38) Who does every man love? His mother.

Temptation: analyze this as “for every man x , [tell me] who does x love the most

But - this won’t work:

(39) Who did no man invite (to his wedding)? His mother.

Can’t be: “for no man x , [tell me] who did x invite?

(this point in Groenendijk and Stokhof, 1983)

Hence: G&S and Engdahl: these are functional questions, they ask for the identity of a function from individuals to individuals:

(38') what is the function f such that every man is an x who loves $f(x)$

(39') what is the function f such that no man is an x who loves $f(x)$

one implementation: via a functional trace

trace: two parts: one as a function variable f of type $\langle e, e \rangle$

second as an individual variable which is argument of this function

individual variable part “bound” by the way binding takes place in general

function variable: “bound” when occurs as argument of *who*

in variable-free view: functional readings on questions automatic (no need for extra meanings of traces)

individual reading: *every man compose love* - gives $\lambda x[\text{every-man}(\text{love}(x))]$

then combines with *who*

functional reading – same is in binding in general: just $z(\text{loves})$

every man loves : $\lambda f[\text{every-man}(z\text{-loves}(f))]$

then combines with *who*

(do need to say that *who* is polymorphic, but depending on analysis this may also come “for free” from the shift rules available in the grammar)

Case 7: “Ellipsis”: short answers to questions

are many cases where it looks at first glance as if material is surrounded by “hidden” unpronounced material in order to assign it the right meaning

Classic example: short answers to questions: (see my talk at the International Conference preceding the summer school)
substantial expressions)

(39) a. Q. Who left the party at midnight?

b. A. Jill

“common wisdom” (see especially Merchant, 2007 in *Linguistics and Philosophy* and the references there):

These contain deleted or silent material, and (39) is “really” (40):

(40) Jill ~~left the party at midnight~~.

Call this the Silent Linguistic Material (SLM) analysis

What does this have to do with the debate about DC?

- (a) Phonological suppression by itself is compatible with DC: a unary rule taking an input triple and yielding as output a triple with the same syntax and meaning, but with the phonology suppressed.

HOWEVER: silencing/deleted is stated in virtue of an expression's identity with something else in the discourse context. This cannot be (easily) stated, as it is not a *local property* of the input expression.

- (b) On the other hand: once adopt DC analyses of things - many of the arguments *for* SLM disappear. They are based on assuming non-DC analyses

- Many arguments - especially about pronouns and binding - which *assume constraints on non-local chunks of representation*

These cannot be stated in any case if DC is right. So they need alternative analyses, but such alternatives are available.

A classic example - discussed above:

- (41) a. Q. Which woman does every man love the most?
b. A. His mother.

Standard "wisdom" -

A bound pronoun must be c-commanded by its binder (at LF)

- presumably means: no way to get the right interpretation without making use of something like the LF in (35)

But: have already seen that this not true - Variable-free analysis above makes no use of LF (or "binding"), or assignments

functional question: as above

answer: = a function of the right type $[[\text{his-mother}]] = \lambda x[\text{the-mother-of}(x)]$

no need for SLM to get the right meaning

Case 8: Separable idioms

- (42) a. Sally had to pull strings to get her son admitted to Brown.
b. Sally pulled a lot/an awful lot of strings to get her son admitted to Brown.
c. She refused to pull any strings, and her son still got admitted to Brown.

pull strings: roughly - use influence to (successfully) bring about a result

Common wisdom: this is interpreted as a unit; non-compositionally

meaning of *pull* and meaning of *strings* are not relevant - only when the two combine in this way do we get the “idiomatic” meaning

This in itself not threaten DC, since *pull strings* can be listed as a single lexical item (though see (b) and (c) - we will return)

But this is a separable idiom: one gets the idiomatic reading even if these are not separated:

(43) Strings were pulled in order to get the charges against Sam dropped.

(44) The strings that he pulled got me my job.

one analysis of (44):

strings is inside the relative clause at the level of interpretation, since it has to be object of *pull* for the idiom to be interpreted:

(45) the ___ [he pulled strings] got me my job

BUT: McCawley (1984) - one can turn this argument on its head :

(46) He pulled the strings that got me my job.

Common answer:

the strings can either originate inside or outside the relative clause

(44) = (45)

(46) = the strings [that ___ got me my job]

But this is of no help: The claim that *strings* has no meaning except in the single unit *pull strings* does not answer the following questions:

(a) in (44) - what is the agent of “got me my job”? Something has to be understood as the thing that brought that about

similarly - if *strings* external to the relative clause in (46), the same question arises

(b) (42b) (42c) above; what does *a lot of strings* mean? (or *any strings*?) how can this be quantified if *strings* has no meaning

Other hints that this truly is compositional:

(47) He pulled string after string to get a job.

attested example:

from <http://www.greekchat.com/gcforums/showthread.php?t=115243&page=4>

Speaking of the Steelers being a religion....

Both sides of my family come from Pittsburgh and the ONLY thing that anyone can agree on is supporting the Steelers. A few years ago my cousin got engaged and we were all appalled to learn that her fiance was from Cleveland and *gasp* a born and raised Browns fan. My uncle then proceeded to pull string after string to get the personal priest of the Rooney family, as in the man who literally travels with the Steelers and holds mass with them before each game, to perform the ceremony. In the middle of the very Catholic wedding said priest stops to announce to everyone that he agreed to marry them ONLY after making my cousin swear on a Bible to raise their children Steelers fans... Welcome to Pittsburgh.

a google search reveals many, many others

Almost all “separable” idioms show the same point

much literature on *pay attention* - as “idiomatic” - yet this rather clearly compositional

(48) He paid careful attention to the lecture.

if *pay attention* not composed of separable meanings, what is *careful* modifying?

(49) The attention that he paid to that lecture was great. Etc.

Compositional analyses: G. Nunberg, I. Sag, and T. Wasow, “Idioms”, *Language* 1994.