Pied Piping Semantics in Relative Clauses

Pauline Jacobson
Brown University
pauline_jacobson@brown.edu

Workshop on the Syntax and Semantics of Relative Clauses
Tel Aviv University
June, 2000

I. Defending the program of "direct compositionality" against some obvious challenges drawn from the domain of Pied-Piping
(tools will make use of - surprise (!) - Variable-Free Semantics)
(a) the basic semantics
(b) the interaction of Pied-Piping with functional NPs
(c) direct compositionality without QR for ACD, and the interaction of ACD with Pied-Piping

II. Some implications for the semantics of relative pronouns and of *wh* words in general, and concomitant implications for type-shifting vs. "radical lexicalism"

III. Two more subtle challenges for the hypothesis of "direct compositionality" of Pied-Piping:
(a) Pied-Piping and WCO effects
(b) "Principle C" effects and Reconstruction
Preliminary answers to these apparent challenges (and open questions)

IV. A puzzle surrounding scope ambiguities of Pied-Piped material in relative clauses

Part I: *Defending the hypothesis of direct compositionality (Montague, 1974, etc.)*
Subtext: Propoganda for Variable-Free Semantics

- the syntactic combinatorics build well-formed expressions; the semantics works "in tandem" to supply a model-theoretic interpretation
  - no mediating level of syntactic representation, such as LF
  - no rules mapping structures into LFs
  - no need to do the compositional analysis twice
    variations on this: depend on how rich the syntactic combinatorics can be
    and concomitantly: on how rich (how much structural infomation) are the objects which combine in the syntax
    cf., Montague - Quantifying-In vs. a less rich syntax with type-shift rules (as in Categorial Grammar)

- much (assumed) evidence for an abstract level of LF: from facts about bound pronouns
  - Claim: based on the mistaken idea that binding is a relationship between two linguistic expressions (NPs, DPs, trace, variable, whatever)
  - if instead binding is a relationship between argument "slots" (via a manipulation on meanings) many of these arguments disappear
  - Hence: "Variable Free Semantics" (cf., Jacobson, *L&P* 1999; *NALS* to appear)
  - no need for indices in the syntax (and no variables or assignment functions in the semantics)

Background: The semantics of garden variety relative clauses:

(1) the man who Mary invited

(2) *Mary invited* or *who Mary invited:* standard semantics: $\lambda x[\text{invite}'(x)(m)]$

(3) relatively standard view:
trace in object position, translates as variable
variable eventually gets λ-abstracted over
who contributes nothing to the meaning, or:
who_x means λx
or, a variant:
who interpreted in situ - either before movement or via "reconstruction"
who is the variable
λ-abstraction happens higher up in the compositional interpretation
the result - after λ-abstraction - is a function of type <e,t>
which intersects with the head

(4) a slight modification of this view (variable-free, and compatible with Direct Compositionality) (cf., Steedman, 1987)

Mary ; $\lambda P[P(m)]$ of category $S/_R(S/_L NP)$
function composes with *invite'*
Mary invite ; $S/_R NP$; $\lambda P[P(m)] \circ \text{invite}' = \lambda x[\lambda P[P(m)](\text{invite}'(x))]$
$= \lambda x[\text{invite}'(x)(m)]$

who: we'll return to this later, but informally:
(a) does nothing at all
(b) supplies the intersection direction (says: give me two <e,t>'s and I'll intersect them)

NOTE: there are refinements of the syntax here within CG (see, e.g., Oehrle, 1990) and other refinements are imaginable (e.g., "Geach" rule instead of function composition)

The obvious Pied-Piping challenge:

(5) a. the man whose mother Mary invited
b. the man the mother of whom Mary invited

(6) the mother of whom Mary invited: $\lambda x[\text{invite}'(\text{the-mother-of}'(x))(m)]$

the usual intuition: no way to compose that meaning without sticking *the mother of (whom)* into the position of the gap - Hence:

(a) treat *whom* like an ordinary pronoun, do the interpretation before movement (as in "classical TG", Generative Semantics, etc.) or:
(b) do movement, but stick the phrase back at LF ("Reconstruction")

Answer: Pied-Piping via functional "gaps" (Sharvit, 1997, Jacobson, 1997, others?)

(7) Who does every Englishman love? His mother.

Groenendijk & Stokhof, 1983; Engdahl, 1986: functional question analysis:

• gap: a complex meaning f(x)

(8) what is the function f such that every Englishman'(λx[love'(f(x))(x)])

f bound in the way that gaps in questions bound in general
x bound in the normal way for variable-binding

Sharvit, Jacobson, etc: since we can have a functional gap, part of the Pied-Piping semantics will come "for free" from this fact

(9) the man the mother of whom Mary invited
Mary invite $t_{f(x)}$
λ-abstract over f; λ-abstract over x: Mary invited $t_{f(x)}$ ---> λf[λx[invite'(f(x))(m)]]

then, somehow get *the mother of whom* to be of type <e,e>

A closer look: it all comes "for free" in variable-free semantics

• NOTE: the basic idea does not require Variable-Free semantics. Given the existence of functional gaps in questions, part of this will come for free in any theory (see, e.g., Sharvit)
• However, just about everything comes for free in VFS; implementation is much simpler
(a) the very existence of functional "gaps" follows for free from the mechanisms needed for binding in general
(b) the functional reading of the Pied-Piped constituent is also automatic
• with one key assumption: relative pronouns are pronouns

A Crash Course on Variable Free Semantics

• no assignment functions or variables as part of the semantic machinery
• a constituent C containing a pronoun unbound within C denotes a function from individuals to something else (not a function from assignment functions to something else)
e.g. *he-lost'* = function from individuals to propositions (= lost')
his-mother' = the-mother-of'

A. meaning of pronouns: identity function on individuals (possibly with gender information built in)
• hence, of type <e,e>

B. two type-shift rules to regulate the combinatorics:

(i) "Geach" rule **g** - allows a function to combine with a pronoun (of type <e,e>) or with an expression that contains an unbound pronoun
hence: maps a function α of type <a,b> to a function **g**(α) of type <<c,a>,<c,b>>

(10) **g**(α) = λV[λC[α(V(C))]] (for V of type <c,a> and C of type c)

NOTE: the **g** operation is just a unary (Curry'ed) version of function composition. Thus **g**(h)(f) = h ∘ f

(11) he lost: lost' <e,t> ---> **g**(lost') = λf[λx[lost'(f(x))]]
(for f of type <e,e>)
he lost; = **g**(lost')(he') = λf[λx[lost'(f(x))]](λy[y])
= lost' (NOTE: = lost' o λx[x])

(12) his mother = the-mother-of' (of type <e,e>)
(13) his mother lost: **g**(lost')(the-mother-of') =
λf[λx[lost'(f(x))]](the-mother-of') =
λx[lost'(the-mother-of'(x))]

schematically: lost' of type <e,t> ---> **g**(lost') of type <<e,e>,<e,t>>
|_____|
semantically "links"
these two argument slots;
g(lost') thus takes a functional
subject (and returns a property)

(ii) Binding: the **z** rule - takes a function of type <a,<e,b>>, and shifts it to a function of type <<e,a>,<e,b>> in such a way as to "merge" the two e-argument slots
i.e., something that wants an argument of type a and a later individual argument gets instead something of type <e,a>, and "binds" the e-position to its later individual argument slot

(14) let α be a function of type <a,<e,b>>
Then **z**(α) is of type <<e,a>,<e,b>> and is λf[λx[α(f(x))(x)]] (for f of type <e,a>

(15) Every man$_i$ loves his$_i$ mother

loves' ---> **z**(loves') = λf[λx[loves'(f(x))(x)]]
loves-his-mother' = **z**(loves')(his-mother') = λf[λx[loves'(f(x))(x)]](the-mother-of') =
λx[loves'(the-mother-of'(x))(x)]
every man loves his mother: every-man' (λx[loves'(the-mother-of'(x))(x)])

schematically: loves' of type <e,<e,t>> ---> **z**(loves') of type <<e,e>,<e,t>>
|_____|
semantically "links"
these two argument slots;
z(loves') thus takes a functional
object and binds the argument slot
of the function to its subject

C. Free Pronouns

(17) He lost. λx[lost'(x)]
hence, a function of type <e,t> rather than a proposition
• to extract propositional information, listener applies this to a contextually salient individual

• standard account: also not a proposition, but a (non-constant) unction from assignment functions to propsoitions. To extract propositional information, listener must apply this to a contextually salient assignment function.

Consequence 1: paycheck pronouns come for free from the **g** rule (no need for two types of pronouns: one of which is a variable over individuals and one of which is a paycheck pronoun and hence allows sloppy identity). Hence the **g** rule is independently motivated. (see Jacobson, *Natural Language Semantics,* to appear)

Claim: There couldn't be English minus paycheck pronouns
• NOTE: most other accounts: paycheck and regular pronouns are just accidentally homophonous

The basic intuition:

(18) Every 3d grade boy$_i$ loves his$_i$ mother; every fourth grade boy$_j$ hates her$_j$.

her: λf[f] - identity function over functions of type <e,e>
it can be this automatically by applying **g** to the basic meaning of a pronoun
g(λx[x]) = λf[f] (for f of type <e,e>)
every fourth grade boy hates her: λf[every-fourth-grade-boy'(**z**(hates')(f))
argument slot of f here is bound by the **z** rule
entire thing is a function of type <<e,e>,t>, so it is supplied to a contextually salient function of type <e,e>

Consequence 2:

• The existence of functional "gaps" comes for free from the basic mechanisms for variable binding
• Hence: functional questions come (almost?) for free
NOTE: the "almost" caveat depends on the details of question pronouns; the results in this paper suggests that nothing new will be needed for those, either

(19) Who does every Englishman love?
what is the function f such that every-Englishman'(**z**-loves'(f))

• no special "trace"; the gap is just a "missing argument" of type <e,e> - the **z** rule introduces the expectation of a functional argument
• so, this falls out from the apparatus for binding in general

Claim: There couldn't be English minus functional readings for questions

(20) every-Englishman' o **z**(love') =
every-Englishman' o λf[λx[love'(f(x))(x)]] = λf[every-Englishman'(λx[love'(f(x))(x)])]

Note further: we can also automatically get a functional gap via the **g** rule
(the analogue to this observation in the "standard" account: we can introduce f(x) in the position of the trace but fail to bind x by the subject)

• are various derivations which give this; I use the expositorily simplest one

(21) every-Englishman' o love' = λx[every-Englishman'(love'(x))] $\xrightarrow{\mathbf{g}}$
λf[λy[λx[every-Englishman'(love'(x))] (f(y))]] =
λf[λy[every-Englishman'(love'(f(y)))]]

hence: this is waiting for a function of type <e,e> as argument

Consequence 3: It is an automatic property of the system that we can have a "missing" argument of type <e,e>, but where the argument slot of that function is not bound by the subject (NOTE: This true in the standard account, too)

Consequence 4: If we assume that a relative pronoun is a pronoun, the Pied-Piped material automatically has the right meaning - no further type-shifting necessary (no new apparatus; no new rules)

Not claiming: We couldn't have English minus Pied Piping, since the existence of Pied-Piping depends also on the syntax. But nothing extra needed for the semantics.

• This part of a more general point:

(22) Who does every Englishman love? His mother.
his mother automatically has the functional reading

(23) the mother of whom = the mother of him =
mother' <e,<e,t>> ---> **g**(mother') = λf[λx[mother-of'(f(x))]]
him'/whom' <e,e> λy[y]
mother-of-whom' λf[λx[mother-of'(f(x))]](λy[y]) = λx[mother-of'(x)] (= mother-of')
the: <<e,t>,e> λP[ιz[P(z)]] ---> **g** $\lambda R_{<e,<e,t>>}$[λy[λP[ιz[P(z)]](R(y))]] =
λR[λy[ιz[R(y)(z)]]]
the mother of whom' λR[λy[ιz[R(y)(z)]]](mother-of') =
λy[ιz[mother-of'(y)(z)]]
hereafter: λy[the-mother-of'(y)] or, simply: the-mother-of'

(24) the mother of whom Mary invited

Mary invited: as above λf[λx[invite'(f(x))(m)]]
the mother of whom Mary invited: apply this to the-mother-of;
λx[invite'(the-mother-of'(x))(m)]

The moral so far: Not only can we do the basic semantics of Pied-Piping without any reconstruction, but in fact just the mechanisms that we need for pronoun binding in general will do the trick: no new rules, no new machinery. Since anyone needs to do pronoun binding, the **z** rule arguably adds nothing new over what anyone would need. The only possible "new" trick here is the **g** rule. However, this has independent payoffs: e.g., it gets paycheck pronouns for free.

Reconsidering un-Pied-Piped relatives:

(25) the man who Mary invited

to get *the mother of whom* to have the functional reading "for free", we assumed that *whom* is like any other pronoun: the identity function, and hence of type <e,e>

Therefore - we also need to apply **g** in (25)

Put simply: this is just a matter of topicalizing a pronoun (or, in the Pied-Piping case, topicalizing a pronoun-containing constituent)
we need to expect the argument of the gap to be a pronoun, so we need to apply **g**

(26) who Mary invited:
Mary ^{g} invited = $\lambda P[P(m)]$ ^{g} invited = $\lambda x[invite'(x)(m)]$ --- **g**
(this says: instead of expecting an ordinary NP as argument, expect a pronoun or pronoun-containing NP as argument):
$\lambda f[\lambda y[\lambda x[invite'(x)(m)](f(y))]] = \lambda f[\lambda y[invite'(f(y))(m)]]$
who Mary invited: $\lambda f[\lambda y[invite'(f(y))(m)]](\lambda x[x]) =$
$\lambda y[invite'(y)(m)]$

Some more complex and hence more exciting applications of this analysis of Pied Piping:

(a) Interactions with functional NPs (Groenendijk & Stokhof, 1983; von Stechow, 1990; Harper, 1991; Jacobson, 1994; Sharvit, 1996, 1998):

observation: • NPs often shift into functional readings
• as with functional quesitons, this fact is (almost) automatic in VFS
(only one new piece needed to get functional NPs)

what functional NPs buy us:

(i) derive for free the "sloppy" inference in the following:

(27) a. John always buys the thing which Bill buys.
b. $Bill_i$ always buys his_i favorite car.
c. Therefore, $John_j$ always buys his_j favorite car.
(Reinhart, 1991; extending observations of Pollard and Sag, Chierchia, Higginbotham, and others)

(28) a. John always z-buys the function f with range thing such that Bill z-buys f
b. Bill z-buys the-favorite-car-of function.
c. Therefore, John z-buys the favorite-car-of-of function.
d. i.e., therefore: John is an x who buys the favorite-car-of function (x).

(ii) allows "sloppy binding" in general:

(28a) John loves the relative of his that Bill loves - namely his mother.

(iii) binding connectivity in copular sentences (von Stechow, 1990; Jacobson, 1994; Sharvit, 1999)

(29) The woman who every $Englishman_i$ loves (the most) is his_i mother. (Geach, 1974; see also Dahl, 1980)

Note: not a matter of scoping out *every Englishman:*

(30) The only $woman_i$ who no fourth $grader_i$ invited to his_i graduation was his_i grandmother.

(31) the function f with range woman' such that every-Englishman' **z**-loves f is the-mother-of function

Note also:

(31a) Every Englishman loves someone. It's his mother. (Groenendijk & Stokhof, 1983)

The general picture: there is lots of unexpected "sloppy" or "functional" phenomenon. On the program here, these emerge as an automatic consequence of the mechanisms for pronominal binding in general: ***z*** *and* ***g****.*

So, piece 1: with just type-shifting on *woman* we get the functional reading in (29), (30) for free

• internal composition of *the woman who every Englishman loves:* (ignore for now the contribution of *who)*

(32) every-Englishman' o **z**(love') =
every-Englishman' o λf[λx[love'(f(x))(x)]] =
λf[every-Englishman'(λx[love'(f(x))(x)])]

shift *woman* into set of functions with range woman'
intersect the two

λf[every-Englishman'(λx[love'(f(x))(x)]) & y[woman'(f(y))]]
let *the* map this into the unique (contextually salient) function in this set

Piece 2: as before, the fact that *his mother* has the right type of meaning to be the other argument of *be* here is automatic:
his mother = the-mother-of function

Question: Does this really require Variable-Free? Answer: same as before. No (cf., von Stechow, 1990; Sharvit, 1997), but it's simpler: (a) nothing extra needed to get the functional gap hence nothing but the type-shift on the head needed in VFS ; (b) getting *his mother* to have the right type of meaning is automatic. Standard account: needs to allow λ-abstraction to shift x's mother to λx[x's mother]. (c) Once we get the functional reading of the NP, no need to supply a "hidden variable" as argument of this NP; the binding of its argument slot is an automatic consequence of the **z** rule.

The interaction of functional NPs (and functional questions) with Pied-Piping: the Pied-Piped constituent needs a higher order meaning

• again, standard account can do it, but with extra machinery (need to shift by λ-abstractions in the right places)
• no extra stuff will be needed here: existence of **g** gets it for free

(33) ?The only woman the mother of whom every Englishman$_i$ loves is his$_i$ wife.

(34) The only man whose dog no girl$_i$ was willing to pet-sit for is the man who refuses to pay her$_i$.

(35) The man whose dog every girl$_i$ refused to pet-sit for was the man who offered her$_i$ the least money.

(36) Whose mother does every Englishman love the most?

some facts:

(a) the functional readings + Pied Piping are hard to get - but possible
(b) the apparatus needed for them is all in place
    The derivations look complex*, but the rules are not
    * and this is partly an artefact of using λ-notation vs. combinators
(c) fact (a) will hopefully follow from "interpret via the lowest types possible" hypothesis (cf., Partee and Rooth; in a bit the same spirit as Frazier, etc.)
    • as a perceptual principle, not a grammatical one

• the actual ways to put these together involve various combinations of **g** (so as to introduce higher-order argument slots) and **z** - but the interactions are as expected

"Higher Order Pied-Piping"

(37) the woman the mother of whom every Englishman loves:

in a nutshell:

every Englishman loves: **g**(every-Englishman' o **z**(loves'))
where **g** introduces an argument slot of type <e,e>
NOTATION: <e,e> hereafter as ee
Hence: of type <<ee,ee>,<ee,t>>

the mother of whom: **g**(λx[the-mother-of'(x)])
where **g** introduces an argument slot of type e
Hence: of type <ee,ee>

or: $\mathbf{g}_{ee}$(the-mother-of')($\mathbf{g}_{e}$(whom'))
where $\mathbf{g}_{e}$(whom') = λf[f] (i.e., the paycheck type)

spelling it out in λ-notation:

every-Englishman' o **z**(loves) =
λf[every-Englishman'(**z**(loves'(f))]
g applies to this - as in the case of normal relative clauses
Note: **g** here introduces an argument slot of type ee, instead of a slot of type e
λf[every-Englishman'(**z**(loves')(f))] is of type <ee,t> $\xrightarrow{\mathbf{g}}$ to become <<ee,ee>,<ee,t>>
Note: parallel with normal case: <e,t> --> <<e,e>,<e,t>>
This exactly alike except introduce ee argument instead of e
$\lambda H_{<ee,ee>}[\lambda h_{ee}[\lambda f[$every-Englishman'(**z**(loves')(f))](H(h))]] =
$\lambda H_{<ee,ee>}[\lambda h_{ee}[$every-Englishman'(**z**(loves')(H(h)))]]

the-mother-of-whom: λx[the-mother-of'(x)] (type <e,e>) (as shown earlier)
$\xrightarrow{\mathbf{g}}$ to be of type <ee,ee>
λf[λy[λx[the-mother-of'(x)](f(y))]] =

λf[λy[the-mother-of'(f(y))]]

take this as argument of above:
λh[every-Englishman'(**z**(loves')(λf[λy[the-mother-of'(f(y))]](h))))] =
λh[every-Englishman'(**z**(loves')(λy[the-mother-of'(h(y))]))] =
(by unpacking of the meaning of **z**):
λh[every-Englishman'(λx[loves'(λy[the-mother-of'(h(y))](x))(x)])] =
λh[every-Englishman'(λx[loves'(the-mother-of'(h(x))(x)])]

i.e.: λh[every-Englishman is an x who loves the-mother-of h(x))]

then intersect that with shifted woman: which is the set of functions with range woman'

Conclusion so far: Direct interpretation + variable-free semantics: two rules **g** and **z** for pronouns in general (everyone needs an analogue to **z**) - gives us everything: Pied-Piping, functional readings, and their interaction

(b) The interaction of Pied-Piping with Antecedent Contained Deletion: all compatible with direct compositionality (Jacobson, 1998 *SALT* paper)

Background: ACD under direct interpretation (Cormack, 1985, Jacobson, 1992) ACD as TVP Ellipsis

(38) John read every book which Bill did.

- the received wisdom: the semantic composition of a relative clause requires having an open proposition which contains a variable

(39) John read every book which Bill read

- we need a stage in the semantic composition where we have *Bill read x*
- x is then λ-abstracted over to create a property, which intersects with book'

- Therefore, in (38): we also need to have the "missing" stuff be - at the level of the semantic composition - *read x*
- Moreover, since there's no overt VP who can supply this meaning (and/or identical LF), we need to raise out *every book which Bill did* to give *read x* as the antecedent

ACD as TVP Ellipsis:

- given the Steedman-like variable-free account of relative clauses, we are just missing a TVP meaning here
- missing meaning is read', which function composes with type-lifted Bill

- More specifically:
  - assume that VP Ellipsis really is (contra to received wisdom) a matter of "deep anaphora"
  - we just have a missing property which gets contextually supplied
  - but: it is difficult to contextually construct pure functions of type <e,t>

- therefore, it is easiest to do this when the relevant function is the actual meaning of some overt VP

(40) John read *Moby Dick.* Bill will too.

(41) λQ[Bill'(will'(Q))]

(42) John read every book which Bill will

(43) book which Bill will
λR [book' ∩ Bill' o (will' o R)]
λR [book' ∩ λx[Bill'(will'(R(x)))]]

and *read'* can be supplied as the "missing" 2-place relations

TVP Ellipsis is possible across sentences in a discourse:

(44) John read every book, but *Moby Dick* is the only book that Tom did.

(45) Bagels, I like. Donuts, I don't. (Evans, 1988)

TVP Ellipsis without an overt antecedent:

(46) Context: I see you about to grab some cookies.
Not those, you don't.
(47) same context: These, you may. Those, you can't - at least not until they cool down.

ACD and Pied-Piping: (Jacobson, 1998)

(48) Mary voted for every candidate the father of whom John had. (e.g., Bush, ...)

- NOTE: Depending on how one formulates the identity condition on VP Ellipsis, these could be problematic for reconstruction:

- the bottom line: if interpretation requires reconstruction, then the missing/deleted VP won't look anything like its antecedent
John voted for the father of x
Mary voted for x
(but, see Sauerland (1998), Fox (1999), others - no identity of meaning condition)

- if interpretation requires a functional gap with an argument variable (as in the standard semantics) then we have the same problem
John λx[x vote for f(x)]
Mary voted for x

- direct interpretation:

the father of whom John had:

λR[**g**(John' o (had' o R)) (the-father-of')]

in more detail: (note: at each step there is really a λR at the outermost part; I'm simplifying here for readability)

had' o R = λx[had'(R(x))]
λP[P(j)] o λx[had'(R(x))] = λx[had'(R(x))(j)]

teasing out the meaning of *had:* (roughly)
λx[PAST(R(x)(j))]
g on this, so that it wants an ee argument, rather than an e argument: $-->_g$
λf[λy[PAST(R(f(x))(j))]]
this takes the-father-of' as argument: λy[PAST(R(the-father-of-y)(j))]
this intersects with candidate:
λy[candidate'(y) & PAST(R(the-father-of-y)(j))]

at all points: this is actually a function from Rs to this; at the end of the day it is supplied to the contextually salient vote-for function

Part II. *Implications for the semantics of relative pronouns and wh words in other constructions*

• concomitant implications for: how much of the semantics of *wh*-constructions is built into the meaning of lexical items vs. how much arises in some other ways?

• e.g., type-shift rules; general question here relevant to discussion in Bittner on the range of type-shift operations in general; Rullman, and Grosu & Landman on the origin of maximality effects in *wh* -constructions; etc.

- the main moral(s) from the previous section:
  - Pied-Piping semantics comes for free without reconstruction if:
    (a) VFS is right
    (b) relative pronouns are pronouns
  - (a) + (b) allow direct compositionality not only of ordinary Pied-Piping, but also of:
    -- higher order Pied-Piping
    -- Pied-Piping as it interacts with ACD

- crucially: *whom* as $\lambda x[x]$ as with any other ordinary pronoun
- this is what gave the "for free" fact that *the mother of whom/whose mother* is a function of type <e,e> (and, can also be manipulated into <ee,ee> in functional NP case

Note: There are other approaches to the semantics of *wh* words in relative clauses - e.g.,

(1) as an "operator" (presumably this means who_x is λx)
(2) (relatively standard within CG):
$\lambda P[\lambda Q[\lambda x[P(x) \& Q(x)]]]$ (i.e., takes S/NP, takes head, and intersects them)

implications for "radical lexicalism" -

- radical lexicalism: most of the action is built into the meanings of lexical items
- a limited number of combinatory operations available in the syntax
- e.g., intersection is not an available option
- hence, the intersection semantics in relative clauses must be built into the meaning of the relative pronoun, and hence it must have the meaning in (2)

(compare to Bittner: various type-shifters available to allow intersection here without having to build it in to the lexical item)

- *Tentative Conclusion:* It looks like the above story requires *who* to be $\underline{\lambda x[x]}$ (i.e., relative "pronouns" are pronouns after all!). Hence, none of the "fancier meanings" approaches will do. The intersection semantics must thus come in some other way.

- Note: For a fancier account of the meaning of Pied-Piped *wh*-words which is compatible with "radical lexicalism", see Morrill, circa 1993, *L&P* paper

- *A possible problem for the "relative pronouns are pronouns" hypothesis:* (for a related problem, see Fox, 1999)

pronouns in general can be constrastively stressed when their binders contrast (the conditions under which this is possible are explored in Sauerland, dissertation, Sauerland, *SALT 10;* Jacobson, *SALT 10)*

(49) Every third grader$_i$ loves his$_i$ mother. Every FOURTH grader$_j$ HATES his$_j$/HIS$_j$ mother. (Sauerland, 1998)
(50) Every third grader$_i$ loves$_i$ mother. But no third grader$_j$ actually CALLED his$_j$/*HIS$_j$ mother.

the generalization: the pronouns can contrast just in case the binders can

but: relative pronouns never seem to allow this:

(51) Every boy$_i$ who loves his$_i$ teacher does well in school. Every boy$_j$ who HATES HIS$_j$ teacher does badly. (variation on a theme from Jacobson, *SALT 10)*
(52) Every boy$_i$ who loves his$_i$ teacher does well in school. Every boy$_j$ who/*WHO HATES HIS$_j$ teacher does badly.

Jacobson *(SALT 10)* argues for the following account of contrastive stress:
- let a pronoun denote identity function over some contextually salient domain
- if domain supplied differently (as in (44) then stress is possible

tentative hypothesis about the lack of stress in relative pronouns:
- relative pronouns: don't have an open "domain" position - denote identity function over entire domain
- Note, this by itself doesn't answer Fox's problem

moreover: wh-words in other constructions

- Pied-Piping more limited in *wh-* questions and free relatives, but possessive PP'ing possible

(47) Whoever's beer I stole can have it back. (Jacobson, 1995)
(48) I know whose beer you stole.

free relative case: (leaving aside the question of *-ever)*
- Jacobson, 1995: gives it a fancy meaning which includes the maximalizing effect
- But: many more current proposals point out this more general, and derive it in more general ways (e.g. various type-shifting accoutns): Rullman, 1995; Grosu and Landman, 1998; Beck and Rullman, 1999

question case: *wh* words often given "fancy" meanings (e.g., built in as part)
again: implications here: the "fancy" part comes from other sources

- obviously not all *wh* words are just identity function:

(49) I know which man's mother you saw.

which man: identity function on domain of men
Hence *which'* maps a set P into id function on domain P

Part III *Examining two more subtle arguments for reconstruction: Some speculations*

A. Weak (and Strong) Crossover

- "received" position on WCO: it's a constraint on the relationship between two NP/DP/trace/variables

- account of WCO in Jacobson (1992, 1999, to appear):
  - as a constraint on combinatorics:
    z possible - which binds a pronoun slot to a higher argument position
    no **s**: which binds a pronoun slot to a lower argument position
    s: takes a function α of type <e,<a,b>> and maps it to **s**(α) of type <e,<<e,a>,b>>, where **s**(α) = $\lambda x[\lambda f[\alpha(x)(f(x))]]$
    cf., Curry and Feys **S**

(50) *His$_i$ mother loves every man$_i$.

- if there were **s**, then whatever method one uses to get wide-scope objects will allow this
- if there is no **s**, no way to get this

(51) *every man$_i$ who$_i$ his$_i$ mother loves

- Jacobson (1992, 1999) shows that this also could be derivable with **s**
- (apparently) cannot be without **s**: no way to bind gap to the pronoun

But: two caveats:

A. binding out of genitives and complements of relational nouns:

(52) Every man's$_i$ mother loves him$_i$.
(53) The mother of every man$_i$ loves him$_i$.

- above account inherits the problems of a classic c-command account (e.g., Reinhart, 1983)
- no way to get these (without using **s**)
- needs to be tied in to wide scope possibility of this position

B. no account of Strong Crossover except as a special case of WCO

Pied Piping interactions:

(54) *every man whose$_i$ mother he$_i$ loves
Strong and Weak crossover violation (Wasow, 1972(?); Chomsky, 1975)

(55) a. *every man whose$_i$ picture his$_i$ mother liked WCO vs.
(56) b. every man whose$_i$ mother liked his$_i$ picture

- standard wisdom: these argue for reconstruction, which follows if WCO (and SCO) is a constraint between two actual NPs/DPs/traces/variables
- does not argue for a reconstruction account if instead the constraint is built into the combinatorics

But the question remains: do these follow in the account here?

difficult to be sure about (55) vs. (56) - since we first need to get an account of the good binding in (56)

speculation: modulo that problem, this will follow

- basic speculation: there will be no difference between binding with respect to a pronoun and an expression which contains the pronoun: note that they have the same semantics
- however, it's difficult to prove that readings cannot emerge

Tentative conclusion: recasting WCO as a constraint on the combinatorics will entirely remove these kinds of arguments for reconstruction. Right now, though, this is a promissory note rather than a proof. *Nonetheless, the take-away moral:* it's a mistake to take it as axiomatic that WCO is a constraint on the relationship between two actual things (rather than a constraint on the combinatorics which manipulate meanings and "merge" argument slots) - and so a mistake to assume that these facts require reconstruction

B. Principle C effects (in questions; this for convenience)

Basic observation: (originally from Postal, 1970; see also van Riemsdijk and Williams (1981)): No Principle C reconstruction effects in Pied-Piping:

(57) Which of the men who liked $Bill_i$ did he_i visit? (Postal, 1970, "On Coreferential Complement Subject Deletion", *Linguistic Inquiry* ***I*** *)*

1. Claim in literature: Principle C effects show up with arguments (vs. adjuncts) in reconstruction contexts: (Friedin, Lebeaux, and many others)

claimed judgments in recent literature:

(58) a.Which article that mentions $Chomsky_i$ did he_i dislike?
b. *Which article about $Chomsky_i$ did he_i dislike?

note: controlling for length effects, the claimed pattern would also be as follows:
(59) Which article mentioning/*about $Chomsky_i$ did he_i dislike?
(60) Which article mentioning/*about $Chomsky_i$ that appeared in *The New York Times* did he_i dislike?
(61) Which article that appeared in *The Times* mentioning/*about $Chomsky_i$ did he_i dislike?

2. Principle C effects show up on narrow scope readings (Heycock, 1995; Fox, 1998; etc.)

(62) How many books that criticize him_i does $Chomsky_i$ think you should read (in order to be well educated)?
- wide scope and narrow scope ("number") reading

(63) How many books that criticize $Chomsky_i$ does he_i think that you should read (in order to be well educated)?
- narrow scope ("number"/de dicto) reading disappears

Reply:

(1) Principle C not well-understood; why should it be configurational?
- if a constraint on configurational information, needs to be a constraint on something that the syntax has access to: but - not a constraint on indexing (see, e.g., Grodzinsky and Reinhart, 1993)

(though cf. attempt to derive it from partially configurational facts: Reinhart (1983); Grodzinsky and Reinhart (1993), etc.)

(2) Examples suggesting that it is not configurational (McCray, 1980; Sells, 1987, etc.)

(64) a. *She$_i$ was told that Mary$_i$ would have to step on some people.
b. She$_i$ was told that in order to get anywhere in this dog-eat-dog world, Mary$_i$ would have to step on some people. (McCray, 1980)

(65) a. *He$_i$ heard from us that Walter$_i$ will never be allowed to enter the Lodge.
b. He heard from us that under no circumstances will Walter$_i$ be allowed to enter the Lodge. (Sells, 1987)

see also Kuno (1987)

(3) Argument/Adjunct distinction in "reconstruction": facts are incorrect
- some preliminary "informant work" - 5 informants - 4 linguists, one well-trained non-linguist (all who get the basic run-of-the-mill Principle C cases; none who knew the relevant discussions in the literature)
- 3 informants: no contrast whatsoever; other 2: (58b) slightly better than (58a) (= reverse of claimed judgments)
(all informants then expressed incredulity at the claim that (58b) is significantly worse than (58a))

(4) Narrow scope facts:

- arguably, this has something to do with "point of view" (cf, Kuno, 1987)
- evidence: *the effect disappears if we reverse the subjects of the verbs:*

(66) How many articles that criticize Chomsky$_i$ do you think he$_i$ should/ought to read? (in order for him to be well educated)
(this has been checked with 3 informants besides myself)

Part IV: *A scope puzzle with relative clauses:*

scope ambiguities with Pied-Piped material in relative clauses:

(67) Tom is the man whose sister/ the sister of whom Bill wants to date.

following ideas of Sharvit (1997) (see also Cresti, Rullman and others):

narrow scope: gap of type <s,<e,e>> - *the sister of whom* as <s,<e,e>> - world argument position bound under *want*
wide scope: gap of type <e,e> - *the sister of whom* of type <s,<e,e>> but its world argument position is bound (by world of evaluation): gap of type <e,e>

a promissory note: the full set of combinatorics in Variable-Free approach need to be worked out to get these results

The puzzle: In "normal" relative clauses within a run-of-the-mill extensional position, it is extremely difficult to get the narrow reading:

(68) I invited the man whose sister Bill wants to date.

nb:
(69) a. I invited the man whose sister Bill would like to date.
b. I invited the man whose sister Bill would want to date.
...if she had one

first hypothesis: gap in relative clauses can only be of type <e,e>

But: the missing reading re-emerges in (a) "predicative position" of specificational compular sentences; (b) concealed questions

(70) Tom is the man whose sister Bill wants to date.
narrow reading is perfect
(71) I found out the man whose sister Bill wants to date.
narrow reading is perfect

first attempt: (70) and (71) both involve a type-shift from ordinary individual reading (e.g., (70) via Partee IDENT; concealed questions into something perhaps fancier)
but: how can type-shifting introduce a new meaning internal to the relative clause????

second attempt: both meanings available, but extensionalizing *the man whose sister Bill wants to date* "squashes" one of them
but: why should the interpretation of the NP as a whole care about something internal to the relative clause?

third attempt: not actually a part of the semantic combinatorics, but a combination of function and presuppositions of relative clauses + what happens in extensional positions

Note: the reading is not entirely impossible, if one is clever enough about contexts
Context supplied by Lance Nathan ("Captain Context")

(72) Mary is totally delusional. She thinks that both Bill and Tom are evil emporers. And that each one has a dragon that carries out their orders. She can't stand either of them - she especially hates Bill. She wants to kill them both - but as far as Tom goes, she's happy to kill just him and let his dragon survive. As to Bill, she hates him so much that she wants to kill him, his mother, his dragon, his double, ...

I invited the man whose dragon Mary wants to kill.

- we need a context in general to supply a reason for the restrictor
- this should be true in the wide scope case too - but: if no context supplied, we try to take these as non-restrictives (allowed when relative clause is headed by *the* - cf., relevant discussion in Sells (1984))

(73) I invited every man whose sister Bill wants to date.