Tanya Reinhart

Strategies of Anaphora Resolution*

The major points of this paper are:

a. The definition of binding in terms of coindexation (or identity of variables) should be replaced by a definition based on the traditional logical-syntax concept of binding. Apart for conceptual reasons, the coindexation view faces empirical problems and disables stating binding generalizations.

b. The traditional distinction between binding and coreference should be modified to distinguish between binding and covaluation. Covaluation is available regardless of the referential status of the antecedent, and shows up also in quantified contexts, as shown in Heim (1993).

c. Covaluation is not governed by considerations of the computational system but by an interface strategy in the spirit proposed in Reinhart (1983). However, although this still requires reference-set computation, there is reason to believe, contrary to my previous view, that this strategy is not an instance of economy, as developed in recent syntactic theory.

1. Two procedures of anaphora resolution

Pronouns are commonly viewed as variables. Thus, (1b) corresponds to (2a), where the predicate contains a free variable. This means that until the pronoun is assigned a value, the predicate is an open property (does not form a set). There are two distinct procedures for pronoun resolution: *binding* and (what I will label) *covaluation*. In the first, we close the property: A common technical implementation is that the variable gets bound by the λ-operator, as in (2b). Here, the predicate denotes the set of individuals who think that they have got the flu, and the sentence asserts that Lili is in this set.

* I would like to thank Danny Fox and Yoad Winter for comments and discussion.

(1) a. Lucie didn't show up today.
b. Lili thinks she's got the flu.

(2) a. Lili (λx (x thinks z has got the flu))
b. *Binding:* Lili (λx (x thinks x has got the flu))
c. *Covaluation:* Lili (λx (x thinks z has got the flu) & *z = Lucie*))

In the second, the free variable is assigned a value, say, from the discourse storage.[1] Suppose (1b) is uttered in the context of (1a). We have stored an entry for *Lucie*, and when the pronoun *she* is encountered, it can be assigned this value. In theory neutral terms, I will represent this assignment as in (2c), where *Lucie* is a discourse entry, and the pronoun is covalued with this entry. (The pronoun can also be covalued with the entry *Lili*, yielding an interpretation equivalent to (2b).)

Let us turn now to the way this basic distinction is captured in the theory of anaphora.

1.1 *The current picture*

The logical concept of binding is trivial: variables are bound by operators. However, in the linguistic theory of anaphora, it turned out useful to talk about the relation between arguments (pronouns and their antecedents), since this enabled the formulation of syntactic conditions on binding, like condition B. The definition of binding assumed since Chomsky (1981) is (3):

(3) *Definition of binding:*
α binds β iff α and β are coindexed, and α c-commands β.

The logical and the syntactic use of the notion 'binding' are, thus, substantially different. Logical binding is a relation between operators and variables, and not between arguments, but syntactic binding is a relation between variables (indices), i.e. between arguments. In the binding construal of (1b), on the syntactic view, *Lili* is said to bind *she*. Technically, then, in the syntactic representation (2b), where indices are replaced with bound variables, one occurrence of the variable x binds the other (which is a meaningless description if one uses the logical concept of binding).

I believe that the idea of capturing binding with restrictions on syntactic coindexation, or identity of variables, has led to several stubborn problems in the theory of anaphora.[2] One set of problems it creates, is in the area of defining the syntactic restrictions on binding. I will only illustrate this briefly here.

If binding is a relation between indices, or variables, it is not trivial to distinguish (4a) from (4b) (a 'strong-cross-over' configuration).

[1] An assumption standard since the eighties is that while processing sentences in context, we build an inventory of discourse entities, which can serve further as antecedents of anaphoric expressions (McCawley 1979, Prince 1981, Heim 1982).

[2] For further — conceptual — problems see Ristad (1992).

(4) a. Who$_i$ e$_i$ said we should invite him$_i$?
b. *Who$_i$ did he$_i$ say we should invite e$_i$?
c. who (λx (x said we should invite x))

Both the pronoun and the *wh*-trace stand for variables. In (4a) both can be (logically) bound by the same operator, yielding a representation like (4c). But in (4b) they cannot, although the relations of the two variable-arguments appear identical. To filter (4b) out, it was necessary to assume a syntactic restriction, known as condition C of the binding theory, which disallows the coindexation in (4b) (hence the interpretation (4c) for it). But how could this condition distinguish (4a) and (4b), given that in both we have coindexed variables? The assumption (since Chomsky (1981) has been that they differ syntactically: *wh*-traces were defined as R-expressions (the same type as referential DPs), while pronouns as 'pronouns.' (condition C prohibits R-expressions from being bound by another argument, in the sense of binding in (3).)

However, Reinhart and Reuland (1993) argue that by all syntactic criteria, pronouns and *wh*-traces pattern alike: They occur in argument position, get full case specification, and pattern the same in other anaphora and chain contexts. Though I cannot enter the details here, the reflexivity frameworks provides evidence for, and rests crucially on the shared typology of pronouns and *wh*-traces as +R (full argument specification, unlike NP-traces and anaphors). If true, the condition C solution to the problem in (4) is not feasible, and the problem remains unsolved, under the present view of binding. No satisfactory alternative account has been offered for the strong cross-over problem. In Grodzinsky and Reinhart (1993), it is treated on a par with weak cross-over, thus leaving unexplained why strong cross-over is so much worse than weak cross-over. The same holds for categorial-grammar accounts such as Jacobson (forthcoming).

Next (and more relevant for the present discussion), problems arise in interpreting the indexing system. The source of the problem is that, in fact, binding and covaluation are both defined in terms of identity of indices, or variables. The only difference, in the syntactic framework, is in the structural configuration: binding is coindexation under c-command. For this reason, it is actually impossible to state, within this framework, the full range of the distinction between binding and covaluation, or between the two procedures of anaphora resolution that we observed. Let us see this with some detail.

Given the two procedures above, the term 'antecedent' is ambiguous (as widely observed): If *Lili* is identified as the antecedent of the pronoun in (1b), the sentence has two anaphora construals. Since *Lili* is also in the discourse storage, (1b) can have, along with (2b), the covaluation construal (5a).

(1) b. Lili thinks she's got the flu.

(2) b. *Binding:*
Lili (λx (x thinks x has got the flu))

(5) a. *Covaluation:*
Lili (λx (x thinks z has got the flu) & *z* =*Lili*))
b. Lili thinks she has got the flu, and Max does too.

Though (2b) and (5a) are equivalent, it was discovered in the seventies (since Keenan 1971) that certain contexts show that there is a real ambiguity here. E.g. assuming that *she* is *Lili*, the elliptic second conjunct of (5b) can mean either that Max thinks that Lili has got the flu (the 'strict' reading), or that Max himself has got it (the 'sloppy' reading). The first is obtained if the elided predicate is construed as in (5a), and the second — if it is the predicate of (2b). Another well known disambiguator is *only*, as in (6a).

(6) a. Only Lucie respects her husband.
 b. *binding*: Only Lucie (λx (x respects x's husband))
 c. *Covaluation:* Only Lucie (λx (x respects her husband) & *her = Lucie*)

(6b) entails that unlike Lucie, other women do not respect their husbands; (6c) entails that other women do not respect Lucie's husband. The two construals are, thus, truth-conditionally distinct.

At earlier stages, a standard assumption, which I shared, was that the distinction at issue is between binding and coreference, i.e. that covaluation is possible only when the antecedent is a referential NP. However, Heim (1993) points out that precisely the same ambiguity illustrated in (6), can be found also when the antecedent is not referential, as in (7).

(7) a. Every wife thinks that only she respects her husband.
 b. Every wife (λx (x thinks that [only x respects x's husband]))
 c. Every wife$_i$ thinks that only she$_i$ respects her$_i$ husband.

(7a) can be construed as entailing either that every wife thinks that other wives do not respect their husbands, or that every wife thinks other wives do not respect her husband.

The most immediate question is what gives rise to the ambiguity of (7a). Since coreference is irrelevant, we are left with the syntactic coindexation in (7c), which is defined (by (3)) as binding, since *she* is coindexed with, and c-commands *her*. Coindexation is just the identity of variables, and (7b), thus, is the representation where the indices are translated into such identical variables. Of course, one could skip the syntactic coindexation (7c), and derive (7b) directly, by letting the top operator bind all free pronouns.[3] The problem remains exactly the same: Looking at the single binding representation — (7b) — that our system associates with (7a), the fact that (7a) is ambiguous is a mystery.

It seems clear that the ambiguity in (6) and (7) should be related somehow. But while coreference could provide another representation for (7), there is nothing in our present theoretical machinery that can provide further distinctions between identical variables.

Heim concludes that the present coindexation system is insufficient. She introduces a

[3] Syntactic coindexation is just a technical device, with no psychological reality. It was never actually necessary for anaphoric binding (as opposed to movement). It was assumed only in order to capture uniform patterns of movement and anaphora. That identical results can be captured by direct translation of unindexed pronouns as variable bound by λ-operators was argued e.g. in Reinhart (1983) p. 159–160.

distinction between two index-types ('inner' and 'outer'). As she points out, the intuition behind her indexation system is similar to that which led Higginbotham (e.g. 1983) to replace indices with a theory of linking (represented by arrows). Fox (1995b) offers the most explicit formulation of this intuition, which is not dependent on index-types (and is also the closest to what I propose below).

However, the basic view of binding remains, in these frameworks, the same as in (3): Binding is a relation between arguments (variables). Hence, the syntactic problems, exemplified here with strong cross over, remain untouched. Indeed, these authors assume the standard condition C, with its arbitrary distinction between pronouns and *wh*-traces. Other complications stemming from this view will be mentioned in section 4.

My first goal is to explore further the intuition that, I believe, underlies these proposals, and to relate it to the actual procedures of anaphora resolution we started with. This should enable addressing also the syntactic problems mentioned above (e.g. eliminating condition C), as well as problems of ellipsis we turn to in section 4.

As our starting point, we need a more explicit definition of the distinction between binding and covaluation.

1.2 *What is binding?*

Rather than examining coindexation, or identity of variables, let us look at the problem posed by (7) from the perspective of the resolution of pronouns that we started with. Let us start with the embedded clause of (7), given in (8a).

(8) a. ...only she (λy(y respects her husband))
b. *Open VP property:* ...only she (λy(y respects x's husband))
c. *Closed VP:* ...only she (λy(y respects y's husband))

The VP in (8a) contains a free variable (*her*). As before, we have precisely two options when encountering a free variable: Either do nothing at this local stage, namely leave the VP as an open property, with a free variable, awaiting a value, as in (8b) (which is just a different notation for (8a)), or bind the variable and close the set, yielding (8c). Here, the VP denotes the set of female entities each respecting her husband. In (8a,b), then, there are two free variable left (*she* and *her*), in (8c) — just one (*she*).

Now we move to the higher clause of (7). Again we have the same two options: leaving the variables free, awaiting a value from the discourse storage, or closing the top VP. Suppose we opt for the second, and bind all variables which are still free at this stage to the top operator (λx). We then obtain the two representations in (9).

(7) Every wife thinks that only she respects her husband.

(9) a. Every wife (λx (x thinks that [only x (λy(y respects x's husband))]))
b. Every wife (λx (x thinks that [only x (λy(y respects y's husband))]))

The sets denoted by the lower λ-predicates remain different. If we take out *only*, this difference would not be noticed, since the two representations would end up equivalent. But with *only* we get different results if for every value of x, x thinks that only x belongs to the set of people respecting x's husband (9a), or that only x belongs to the set of people respecting their own husband (9b). (9a), thus, entails that every wife thinks that other wives do not respect her husband, while (9b) entails that every wife thinks that other wives do not respect their husbands — the distinction we wanted to capture.

As mentioned, from the perspective of syntactic binding (as defined in (3)), the relation of *she* and *her* in (9a) is the binding relation. (9a) is just the translation of the coindexation in the syntactic representation (7c), repeated in (10), into identical variables.

(10) Every wife$_i$ thinks that only she$_i$ respects her$_i$ husband.

However, the reading obtained in (9a) corresponds to the covaluation reading of (6) (*Only Lucie respects her husband*), and not to the bound reading. It is (9b) that corresponds to the bound reading of (6). (This will get clearer as we proceed.)

To capture the binding in (9b), we need to return to the logical concept of binding. Binding is just the procedure of closing a property. Under the widely assumed technical implementation I use here, this is obtained by binding a free variable to a λ operator. But other implementations are certainly conceivable. A similar view of binding as closing a property has been developed, in a variable-free framework, by Jacobson (forthcoming) (her G-function).

However, for stating binding restrictions, it is still convenient to be able to talk about relations of Dps, traditionally described as the relation of a pronoun and its antecedent. Let us look, then, at the relation of *she* and *her* in (9b). The *she* variable (*only x*) is the argument of a λ-predicate whose operator (λy) binds the *her* variable (y). This is the relation which I argue is relevant for the binding theory. To avoid confusion with just the standard logical binding it is based on, let us call this relation A(rgument)-binding, defined in (11). Note that also the term A-binding is used here differently than in the syntactic binding theory, which is based on the definition of binding in (3).[4]

(11) *A-Binding (logical-syntax based definition):*
α A-binds β iff α is the sister of a λ-predicate whose operator binds β.

I use the term sister here, rather than argument, to leave open the interpretation of λ-predication. (In the generalized -quantifiers framework, which I assume, in a formula *DP* $(\lambda x\ (P(x)))$, the DP denotes a set of sets, and thus, although it is the syntactic argument of the predicate, it is not semantically an argument.)

In relating syntactic derivations to logical syntax representations with λ operators, I follow the standard assumption that subjects are obligatorily (syntactic) arguments of a λ-predicate. (This is a standard interpretation of the EPP — the predicate is formed by the raising of the subject from its VP-SPEC position) A sentence like *He likes Lucie*,

[4] In that framework (following Chomsky 1981), α A-binds β iff α binds (coindexed with and c-commands) β and α is in an argument position. E.g. in the LF (ib) the trace A-binds the pronoun by the syntactic definition, while by (11), *every boy* does, if the pronoun is construed as in (ic).

(i) a. Every boy loves his mother
b. Every boy$_i$ [t$_i$ loves his$_i$ mother]
c. Every boy (λx x loves x's mother)

then, corresponds to the representation *He (λx (x likes Lucie))*. Other DP's may become arguments of such λ-predicates by movement.

If α A-binds β, by (11), this entails that α c-commands β, at the given representation (since it is a sister of a node containing β). C-command is relevant for the syntactic conditions under which λ-predicates can be formed (compositionality), but has no independent role in defining binding.

By way of a summary, let us check how this definition works in the examples (8) and (9), repeated.

(8) a. ...only she (λy(y respects her husband))
    b. ...only she (λy(y respects x's husband))
    c. ...only she (λy(y respects y's husband))

(9) Every wife thinks that only she respects her husband.
    a. Every wife (λx (x thinks that [only x (λy(y respects x's husband))]))
    b. Every wife (λx (x thinks that [only x (λy(y respects y's husband))]))

In (8a) *she* is a sister of a λ-predicate. If *her* is bound by the λ-operator, as in (8c), *she* A-binds *her*, under (11). Similarly, in (9b) *she* A-binds *her*. In both (9a) and (9b) *every wife* A-binds *she*. (9a) and (9b) differ in that in (9a) *her* is A-bound by *every wife*, while in (9b), *her* is A-bound by *she*. Given (11), then, the binding relations in (9) are distinct, as desired.

1.3 *Covaluation*

We may check now the relations of *she* and *her* in (9a). Neither A-binds the other. Rather, they are both A-bound by *every wife* ('cobound', in the notation of Heim, 1993). But if we want to talk about their relation to each other, the correct description is that *she* and *her* are covalued, i.e. assigned the same value, which is here the bound variable x.

Given the definition of A-binding (11), then, covaluation can be trivially defined, as in (12).

(12) *Covaluation:*
α and β are covalued iff neither A-binds the other and they are assigned the same value.

The distinction between binding and covaluation holds, thus, regardless of the referential status of the relevant expressions. Binding is the logical relation — a relation between an operator and variables. Covaluation is a relation between arguments — variables, or the indices of discourse entities. A-binding is just the notation introduced to describe the relation of antecedents and pronouns when logical binding holds.

With this, then, we capture the fact that it is the same ambiguity in (9) and in (6), which was our point of departure. For (12), the anaphora relation in (86) or (9a) is precisely the same as that in (6c), where the pronoun is not A-bound by *Lucie*, but is assigned the same (discourse) value as *Lucie*.

(6) a. Only Lucie respects her husband.
b. *binding*: Only Lucie (λx (x respects x's husband))
c. *Covaluation:* Only Lucie (λx (x respects her husband) & *her = Lucie*)

What distinguishes the so-called referential anaphora from quantified anaphora is, thus, not the option of covaluation, which is available for both, but the fact that referential NPs form a discourse entity that the pronoun can directly be covalued with, while in the case of quantified NPs, covaluation is only possible between a pronoun and a variable. Compare the anaphora options in (9), to those in (13).

(13) Lucie thinks that (only) she respects her husband.
Binding:
a. Lucie (λx (x thinks that (only) x (λy(y respects y's husband))))
Covaluation:
b. Lucie (λx (x thinks that (only) x (λy(y respects x's husband))))
c. Lucie (λx (x thinks that (only) x (λy(y respects her husband)))) & *her =Lucie*

In (13), if the pronoun *her* is not bound, it has two possible covaluation construals. In the first, the pronoun is covalued with the variable x, yielding the bound covaluation in (13b), precisely as in the case of (9a). But in the second, the pronoun is covalued with *Lucie*, as in (13c). This option is, obviously, not available in the quantified case of (9), since *every wife* does not have a discourse value that the pronoun can pick up. Of course, the two covaluation representations in (13) are equivalent. By logical syntax, if a pronoun is covalued with the argument (sister) of a λ-predicate, it is also covalued with the variables bound by the λ-operator. But the fact that referential NPs allow also the construal in (13c) explains why the 'strict' reading is available in ellipsis sentences like (5) (*Lili thinks she has got the flu, and Max does too*), and not in the parallel cases with a quantified antecedent.

2. Anaphora Restrictions

2.1 *Restrictions on binding*

Under the present view of binding, we expect it to be sensitive to the standard laws of the relations of operators and variables in logical syntax. Specifically, only free variables can be bound by a given operator.

This can be illustrated with the strong-crossover case of (14).

(14) a. Who did he say we should invite t?
b. who (λx (he said we should invite x))
b′. who (λx (he (λy (y said we should invite x))))
c. *Binding:* *who (λx (he (λx (x said we should invite x))))

Given the definition of A-binding in (11), no special anaphora condition is required to explain why there can be no A-binding relation between the pronoun and the *wh*-trace in (14). The trace is bound by the *wh*-operator, so it cannot be A-bound again by the pronoun: Let us assume that the logical syntax representation for the moved *wh* in (14a) is (14b) (though it does not matter if another question operator is assumed there, rather than the λ-operator.) (14b') is the full representation including also the VP λ-predicate. The same will hold throughout for examples numbered with '. In (14b'), λy cannot bind x, since x is bound. Saying that the pronoun A-binds the trace amounts to assuming some nonsensical logical representation like (14c), where the same variable is bound by two operators. (This does not exclude yet an alternative anaphora construal for (14a), to which I return directly.)

As far as binding goes, then, the present definition eliminates the need to assume a special syntactic restriction, like condition C. Recall (from section 1.1) that for a syntactic restriction to apply here, *wh*-traces had to be defined arbitrarily as R-expressions distinct from pronouns. On the present view, what distinguishes them from pronouns is just the fact that they are bound already. As we shall see directly, the fact that binding is disallowed in (14c) (by logic) will enable the covaluation rule to rule out any alternative anaphora construal for this derivation, with no appeal to any special properties of the expressions involved.

Apart from logic, binding also obeys a condition specific to the Computational System — condition B (or the chain condition). In *Max touched him*, logic does not exclude the binding construal *Max (λx (x touched x))*, but it is excluded by the CS condition B.

2.2 *Restrictions on covaluation*

A question which has been debated is whether there are also syntactic conditions on covaluation. So far, I assumed that covaluation is a free procedure, that can be used everywhere (subject only to discourse conditions). But, in fact, there are well known cases where covaluation is excluded.

Note, first, that the logical restriction on binding does not take us very far in filtering out anaphora in the strong crossover case of (14), repeated.

(14) a. Who did he say we should invite t?
c. *Binding:* *who (λx (he (λx (x said we should invite x)))

(15) *Covaluation (he =x):*
a. #who (λx (x said we should invite x))
a'. #who (λx (x (λy (y said we should invite x))

While the binding construal in (14c) is excluded, nothing so far prevents binding *he* to λx (i.e. A-binding *he* to *who*), as in (15a). In (15a), *he* and the *wh*-trace end up covalued, both bound by the same operation. The problem is that (14a) does not allow this covaluation construal.

Precisely the same problem arises in (16a):

(16) a. She said we should invite Lucie.
b. She (λx (x said we should invite Lucie))

c. *Covaluation:* #She (λx (x said we should invite Lucie & *she = Lucie*))

With QR:

d. Lucie (λy (she said we should invite y))

d′. Lucie (λy (she (λx (x said we should invite y))))

e. *binding:* *Lucie (λx (she λx (x said we should invite x))))

Covaluation:

f. #Lucie (λx (x said we should invite x))

g. #Lucie (λx (she said we should invite x & *she = Lucie*))

Here as well, nothing needs to be stipulated to explain why no A-binding relations are possible between *Lucie* and *she* in (16b): *Lucie* is not the type of object that can be bound by the λ-operator whose sister is *she* (since it is not a free variable). For completeness, let us check whether *Lucie* could A-bind the pronoun here. This would require, first, that *Lucie* undergoes covert movement, forming the λ-predicate in (16d). Let us assume that QR is permitted here, i.e. that (16d) is one of the derivations for (16a). In (16d), the variable-trace of *Lucie* is bound, just as in (14), so it cannot be A-bound by the pronoun. (Binding would yield here some illicit representation like (16e).)

But so far nothing blocks a covaluation interpretation for (16a). This could be obtained most naturally with no QR, as in (16c), and the equivalent covaluation construals with QR are also available, as in (16f,g). The problem is, again, that in practice, the pronoun cannot be construed this way.

How the wrong covaluation interpretations of (16a) are blocked has been a subject of debate. In the seventies (when this was viewed as a coreference problem), it was assumed that there is a special syntactic restriction doing the job (Langacker (1966), Lasnik (1976)). Reinhart (1976) formulated it as the requirement that a pronoun cannot corefer with a full NP it c-commands, which became known as condition C of Chomsky (1981).[5]

Recall that under the present definition of A-binding, condition C is not needed to exclude A-binding, which is independently ruled out by considerations of logical syntax. The question is whether it should be assumed as a condition on covaluation.

The problem is not restricted to condition C environments, but shows equally in condition B contexts. Condition B, or Chain, exclude e.g. the binding construal *Max (λx(x saw x)* for *Max saw him*. But the question is what rules out the covaluation construal of such simple sentences (*Max saw him & him = Max).* Again, this covaluation problem shows up equally in quantification contexts. In (17a), condition B or Chain prohibit the A-binding of *him* by *he*. So (17b) is excluded. However, Heim (1993) points out that in

[5] Condition C states that an R-expression (i.e. any NP which is not a free variable) cannot be bound. This may seem superfluous if binding is defined as in (11), since, as we saw, binding is excluded here anyway, by logical syntax. However, as mentioned above, binding is used differently in that framework: being bound is defined as being coindexed with a c-commanding NP. When one NP c-commands the other, covaluation is an instance of syntactic binding. Condition C, thus, correctly blocks the wrong construals of (8) and (10).

semantic terms, it is still possible to obtain anaphora in (17a) without violating principle B, as in (17c), where *Everyone* A-binds both pronouns, so *he* is covalued with *him*, but does not A-bind it, as can be witnessed in the fuller representation in (17c'). (This recapitulates a problem noted by Higginbotham (1983), under a different notation.)

(17) a. Everyone thinks that he can hear him sing in the bathroom.
b. *Everyone (λx (x thinks that x (λy (y can hear y sing in the bathroom))))
c. Everyone (λx (x thinks that x can hear x sing in the bathroom))))
c′. Everyone (λx (x thinks that x (λy (y can hear x sing in the bathroom))))]

If covaluation is governed by syntactic constraints, we need to modify condition B, so that it excludes both binding and covaluation. We end up then, with two conditions on covaluation: condition C, and half of condition B.

We should note that, under the present assumptions, it is possible at least to unify these two conditions. I suggested two changes in the view of current binding theory, that are needed independently of the questions of the restrictions on covaluation: First binding is defined in traditional logical terms, and, consequently, the distinction between binding and covaluation is indifferent to the referential status of the antecedent. These enabled us to see that condition C is only needed for covaluation (since binding is anyway impossible in condition C contexts, on standard logical grounds). Condition C can be now modified, such that it also handles the covaluation residue of condition B.

Note, first, that both conditions C and B apply only when one of the DPs c-command the other. (This was built into the definition of syntactic binding in (3).) In our terms, the relevant syntactic configuration is not directly determined by c-command, but rather by the question whether one of the DPs is in a configuration enabling it to A-bind the other, namely whether it is an argument (sister) of a λ-predicate containing the other. In all instances of blocked covaluation discussed above, which are summarized in (18a-c), one of the Bold faced DP's is in a configuration to A-bind the other. When neither DP is in a configuration to A-bind the other, as in (18d), there are no sentence-level restrictions on covaluation (as entailed by the classical binding theory).

(18) *Strong crossover (14b'):*
a. who (λx (**he** (λy (y said we should invite **x**))))
Condition C (16b):
b. **She** (λx (x said we should invite **Lucie**))
Condition B:
c. **Max** (λx(x saw **him**))
No restrictions on covaluation:
d. The woman next to **him** (λx (x touched **Max**))
Covaluation permitted in a configuration of A-binding:
e. **Lili** thinks **she** has got the flu, and Max does too. (See (5b).)
f. Only **Lucie** respects **her** husband. (See (6).)
g. Every wife thinks that only **she** respects **her** husband. (See (7).)

It appears, then, that the generalization is that whenever A-binding is possible, covaluation is blocked (clause a of (19) below). However, given our discussion so far, this generalization is too strong. We saw that there are cases where both A-binding and covaluation are permitted. These are illustrated again in (18e-g). Although one of the bold-faced DPs can bind the other, a covaluation construal is also possible. The difference available so far between (18a-c) and (18e-g) is that although in both, one DP is in a configuration to A-bind the other in the first, binding is excluded. Let us, then, state this generalization in (19).

(19) *Modified condition C(ovaluation)*
α cannot be covalued with β if
a. α is in a configuration to A-bind β, and
b. α cannot A-bind β.

If correct, the generalization captured in (19) is that when the CS disallows binding, it also disallows covaluation, and it does not matter if binding is blocked by logical syntax (as in the cases of the old condition C), or by binding restrictions specific to natural language (condition B).

In fact, the empirical coverage of (19) is not just identical to that of the original conditions C and B on covaluation. It is also sufficient to capture some anaphora puzzles that cannot be captured by the original binding theory and condition C (the Dahl-cases), to which I return in section 4. But we may note that, if true, (19) has some curious properties. E.g. why should the covaluation option be dependent at all on the option of A-binding? Let us, first, pay more attention to the status of (19).

3. The interface strategy governing covaluation (rule I)

3.1. The empirical problem with (19) is the same as with its predecessors C and B: There are systematic contexts in which it can be violated. Reinhart (1983) argued that this is possible whenever covaluation is not equivalent to binding.

(20) a. [Who is the man with the grey hat? -] He is Ralph Smith.
b. The patient does not remember who he is *t*.
c. Only he (himself) still thinks that Max is a genius.

In (20a,b), it is not easy to imagine a construal of the truth conditions, which will not include covaluation of the pronoun with the NP or the trace. But this covaluation violates condition C. In both cases, however, the covaluation reading is clearly distinct from the bound: What is attributed of the pronoun in (20a,b) is not the property of self identity (λx(x is x), which is what would be obtained by binding. Similarly, believing oneself to be a genius may be true of many people, but what (20c) attributes only to Max, is believing Max to be a genius. (The facts and their interpretation are discussed in Reinhart (1983), Grodzinsky and Reinhart (1993), and, at greater depth, in Heim (1993). So I will not elaborate on them here.)

The same empirical problems surface with the condition B aspects of (19). In the Heim type examples, we may compare (17) to (21a). The later permits anaphora, in violation of condition B, or the modified condition C. But it has only the covaluation interpretation in (21b). (The sentence will be false if, say, Max thinks that someone other than him may have heard him sing in the bathroom.)

(21) a. Everyone thinks that only he can hear him sing in the bathroom.
b. Everyone (λx (x thinks that [only x (λy (y can hear x sing in the bathroom]))))

Other examples are given in (22). If the modified condition C rules out covaluation in all contexts of condition B, this would be right for (22a), but not for (22b,c).[6]

(22) a. The suspect saw him (& *him = the suspect*)
b. The suspect claims that he was in the opera at the time of the murder. But if it is true, only he (himself) saw him there.
c. You are you and she is she. Don't loose your ego!

An alternative view of the restrictions on covaluation was proposed in Reinhart (1983). Stated in terms of current syntactic theory, the view was that covaluation is not directly governed by a condition of the computational system, but by an interface strategy that takes into account the options open for the computational system in generating the given derivation. This general line, I still wish to defend here, though the specific view I took there regarding what this strategy is about, may have been mistaken. I assumed that it fell under familiar economy: The structural generalization is that covaluation is blocked only under c-command, which is the mirror image of where variable binding is always allowed. If we assume that variable binding is the more economical way to capture anaphora, it would follow that avoiding it, when the structure permits it with a different selection from the numeration, is uneconomical. It could only be justified when the covaluation interpretation is distinct from that of binding, so there is a reason to avoid binding. That variable binding is more economical is possibly defendable, in terms of semantic processing. Compare the two interpretations of (23).

(23) Max loves his mother
a. Max (λx (x loves x's mother))
b. Max (λx (x loves z's mother) & (z = Max))

In (a), where the pronoun is bound, the VP forms a set, and we just have to check whether Max is in it. In (23b), the pronoun remains a free variable. The VP remains an open property, and it has to be held open until the pronoun is assigned a value. Only when this happens, assessment can take place. If it turns out that the intended value is,

[6] As noted in Reinhart (1983), covaluation where principle B blocks binding is much harder to find than covaluation in condition C environments. E.g. in both (ia) and (ib), the intended construal of the VP predicate is as in (ii). Still, (ib) is harder than (ia).

(i) a. At the end, only Max voted for Max
b. At the end, only Max voted for him (& *him = m*)

(ii) Only Max (λx (x voted for Max))

The reason suggested there is that (ia) is a more explicit way to express the predicate in (ii). (iib) requires the further task of identifying the value of the pronoun.

anyway, Max, then it is not obvious why we had to go through assignment at all. The economy requirement would be, then, "get rid of free variables — i.e. close open properties — as soon as possible". This view of the economy requirement is developed, under a different terminology, in Fox (1995b).

If this reasoning is correct, then the relevant covaluation strategy can be stated as in (24).

(24) *Covaluation strategy (tentative).*
α and β cannot be covalued if
a. α is in a configuration to A-bind β, and
b. The covaluation interpretation is indistinguishable from what would be obtained if α A-binds β.

(24) is identical in spirit to the generalization proposed in Reinhart (1983), but clause a reflects the changes introduced in this paper regarding what binding is.

However, as plausible as this seems, it is not clear to me that the human processor is indeed sensitive to this type of economy considerations. The problem with this line has always been that, in practice, (23) equally allows both construals of anaphora, as witnessed in the ellipsis context (25a). (The predicate in the second conjunct can be construed as either that of (23a), or of (23b), which can only be obtained if (23) allows both.)

(25) a. Max likes his mother and Felix does too.
b. He likes Max's mother, and Felix does too. (*he ... Max*)

Although ellipsis contexts enable the two construals to be distinct, they crucially do not license covaluation in and of themselves. In (25b), the fact that we want to use the predicate (λx (x likes Max's mother) in the elided conjunct, does not enable covaluation of *Max* and *he* in the first. More generally, evaluating whether the bound reading is distinct from the covaluation reading can be based only on information in the derivation itself (perhaps relative to its previous context), but not on considerations of how it would effect upcoming discourse. (In this sense, this type of economy remains local, as in other instances of economy. See Fox (1995a) for an extensive discussion of this point, in the case of QR in ellipsis structures.)

As stated, (24) will, thus, disable the relevant strict (or 'coreference') interpretation of (25a). We may note that the problem is restricted to VP-ellipsis contexts. In the other contexts, summarized in (18f-g) (like *Only Lucie respects her husband*), the readings obtained by binding and covaluation are distinct locally, so clause (b) of (24) allows covaluation. Fox (1995b) argues that the answer to the problem in (25a) should follow from the theory of ellipsis as PF deletion, and not from the restriction on covaluation. In his approach, the first conjunct in (25a) excludes, indeed, covaluation. However, the relevant strict reading of the second conjunct is still generated, given the way the identity requirements on deletion are defined. At the present, however, this result can be derived only by adding stipulations. I conclude, for now, that this economy view has not been, after all, precisely on the right track.

As mentioned, the intuition behind (24) has been that covaluation is excluded whenever an equivalent binding is possible. The alternative view which emerges is that it is the other way around: covaluation is excluded whenever an equivalent binding is impossible (in a configuration allowing it in principle.) In fact, this was built into the formulation of the covaluation rule I in Grodzinsky and Reinhart (1993, footnote 13), but the account was highly stipulative. Given the analysis in section 2, it is possible now to state the covaluation generalization in (26), which just adds a clause (c) to the covaluation generalization proposed in (19) as the modified condition C. For the sake of continuity, I will continue to refer to the interface strategy (26) as 'rule I'.[7]

(26) *Rule I (-an interface rule)*
α and β cannot be covalued in a derivation D, if
a. α is in a configuration to A-bind β, and
b. α cannot A-bind β in D, and
c. The covaluation interpretation is indistinguishable from what would be obtained if α A-binds β.

If there is a broader strategy behind (26), it seems to be that if a certain interpretation is blocked by the computational system, you would not sneak in precisely the same interpretation for the given derivation, by using machinery available for the systems of

[7] Rule I, of Grodzinsky and Reinhart (1993) is given below, with *covalued* replacing *corefer* in the original formulation.

(i) NP A cannot be covalued with NP B if A could not be bound by B, and replacing A at LF with a variable bound by the trace of B, yields an indistinguishable interpretation. (G&R 1993, footnote 13).

Independently of the question whether it is economy which explains rule I, this formulation had another problem, which is now addressed.

(i) still assumes the syntactic definition of binding: α binds β iff α and β are coindexed, and α c-commands β. (Hence, binding obtains between the trace of the λ-argument and another variable). Under this formulation, rule I could not, in fact, rule out cases of strong crossover, such as (ii) (which were treated there on a par with weak crossover, as in (iii)).

(ii) a. Who did he say that we should invite t?
b. who (λx (x said that we should invite x)

(iii) Who did his mother spoil t?

Recall that in that system there is no condition C. So nothing can rule out independently coindexation of *he* and the trace in (iia). The first condition of (i), thus, is not met in (ii), hence rule I does not apply here. The interpretation (iib) was ruled out by the translation definitions, that entailed that a pronoun is a bound variable iff its binder is in an argument position (A-binder, under the previous definition of binding) (G&R, (15c)). This equally disallows binding of the pronoun in (iia) and (iii). The striking difference in acceptability of weak and strong crossover violations was lost in that system.

Under the present definition, for *he* to bind the trace means that the trace should be bound by the VP λ-operator whose sister is *he*, which is impossible, since the trace is already bound. So (ii) is subject to rule I, and the special status of strong crossover is restored.

Note, again, that this is independent of the issue of economy. Replacing the definition of binding would give the right results here also under the economy view of rule I.

use. It is easy to see why such a strategy could be useful at the interface. The problem for users of linguistic derivations is how to minimize the set of possible interpretations of a given PF. The more options there are, the more mysterious is the fact that speakers manage to understand each other. In the specific case of anaphora resolution, the problem is how to restrict the set of potential antecedents for a given pronoun (i.e. the set of potential values). If the computational system provides a restriction of that set, it is not cooperative for users to overrule that, even if they have the machinery to do so.

I will leave open here the question whether there is a broader strategy behind (26), and assume it as a specific strategy for anaphora resolution. In any case, answering this question requires getting fully explicit about what type of limitations of the CS cannot be bypassed at the interface. There may be various kinds of legitimate interface procedures employed to enrich expressive power,[8] as well as optional steps in the derivation which enable certain discourse uses. If a broader strategy exists here, it only restricts bypassing an actual prohibition, i.e. it rules out a derivation only if it (or a member in its reference set) involves a violation of principles of the CS, as, in our case, condition B, or logical syntax prohibition against binding a variable doubly. Technically, what limits (26) to apply just in these cases is its clause (a). This is needed to exclude sentences like (27) from the scope of (26).

(27) a. Max's mother loves him (*he* = *Max*).
b. His mother loves Max (*he* = *Max*).
c. The lady next to him kissed Max (*he* = *Max*).

Neither *Max* nor the pronoun are in a configuration to bind the other in (27). Hence, (a) of (26) does not hold, and no further aspects of rule I need to be checked. This means that covaluation is free in such structures, as far as rule I is concerned.[9]

[8] Ariel Cohen pointed out that presupposition accommodation may be an example for such procedure.

[9] In the contexts of (27) binding is impossible (e.g.**His mother loves everyone)*. This is attributed to a 'weak crossover' generalization, which (as before) does not follow from anything discussed in this paper. Without clause (a), it may seem that clause (b) of rule I would rule out the covaluation construal in (27).

In the long run, clause (a) may be found just a reflex of a more semantic property: When a variable is A-bound by a c-commanding argument, this always reduces the number of open properties. A-binding by a non-c-commanding antecedent (which is created by QR) is vacuous, in the sense that it does not reduce open properties. To see this, let us compare (i) and (ii).

(i) a. He loves his mother
b. $[_{ip}$ x $[_{vp}$ λy (y loves z's mother)]]]
c. $[_{ip}$ x $[_{vp}$ λy (y loves y's mother)]]]

In the IP of (ib), there are two open properties: The lower VP, and the IP itself, which both contain a free variable. If we bind the free variable z, as in (ic), the VP can be closed (i.e. form a set), and we are left with just one property open — the IP.

(ii) a. His mother loves him.
b. $[_{ip}$ x's mother $[_{vp}$ λy (y loves z)]]]
c. $[_{ip}$ x (λz (z's mother $[_{vp}$ λy (y loves z)]]]

3.2. Under either of the formulations of the covaluation strategy ((26) or (24)), if we get precise about the way it applies, it must involve reference-set computation: Computing clause (c) requires constructing a reference-set, which includes the current derivation under the covaluation interpretation and another member with the binding interpretation. If the two members are equivalent, the covaluation interpretation is blocked. (For (24), this is so because covaluation is less economical; for (26) — because it enables bypassing a prohibition of the CS.) Let us specify the procedure of constructing the reference-set for (26) as in (26′).

(26′) To check clause (c) of (26) construct a comparison-representation by replacing β, with a variable A-bound by α.

Let us now check in more detail how rule I works in assessing whether covaluation should be permitted in a given derivation. In (23), repeated, the derivation D that we are considering is (23b), where the pronoun remained a free variable. The question is whether *his* (z) can be covalued with *Max*. Since *Max* is in a configuration to bind *his* (i.e (a) of (26) holds), the next clauses of rule I must be considered.

(23) Max loves his mother
a. Max (λx (x loves x's mother))
b. Max (λx (x loves z's mother) & (z = *Max*))

However, (b) of (26) does not hold here. Since the pronoun is a free variable, in the scope of the λ-operator, further application of binding to the given derivation would have enabled it to be bound (— the same operation that applied in (23a)). Hence, assessing is completed, and (c) of (26) need not be checked.

Next, consider the 'strong cross over' case of (14), repeated in (28). The derivation on which anaphora assessment is computed is (28b). The question is whether *he* could be covalued with x, which would lead to the interpretation in (28c). (I assume that *who* can A-bind *he* here, but if we choose to do that, we obtain a covaluation of *he* and *x*, which needs to be checked.)

(28) a. Who did he say we should invite t?
b. who (λx (he said we should invite x))
Covaluation:

In (iia), there are two open properties, as before. But if we A-bind z to x, or x to z (technically obtainable by QR), we get (iic), which has precisely the same number of open properties: The VP is still open, and so is the IP.

So, possibly, rule I can be stated to disallow covaluation of α and β if α cannot A-bind β non-vacuously, and covaluation is nevertheless equivalent to non-vacuous binding. Since no equivalent non-vacuous binding exists here, clause c never has to be checked in such (weak crossover) structures, i.e. covaluation is always allowed.

On this view, it may turn out that the generalization behind the weak-crossover restriction is something like 'avoid vacuous binding'.

c. who (λx (x said we should invite x))
c′. who (λx (x (λz (z said we should invite **x**))))
d. *Binding- comparison:*
who (λx (x (λz (z said we should invite **z**))))
(c) ↔ (d), hence *he* ≠ x; (c) ruled out.

he is in a configuration to A-bind x, so we turn to clause (b) of rule I. Here, as we saw, no further operation on the derivation (28b) could allow *he* to A-bind x, since x is A-bound already by *who*. We are now considering the covaluation derivation in (28c), fully specified in (28c')). To decide whether this is a possible construal, we have to check, clause (c) of rule I, namely, check whether the result of covaluation is distinguishable from what we would have obtained by binding. This requires, first, constructing the comparison-representation which would have been derived if binding was not excluded here. The procedure for constructing this binding-comparison, is (26'), repeated:

(26′) To check clause (c) of (26) construct a comparison-representation by replacing β, with a variable A-bound by α.

In (28c′) the c-commanding x (*he*) is α of rule I, and the lower x (the trace) is β. (For convenience, the β element is printed in bold-face in the examples below.) The binding comparison is obtained by replacing the trace x with a variable A-bound by *he*, as in (28d). Next, we check whether the two representations are semantically distinguishable. We find that they are equivalent. So, the verdict of rule I is that *he* cannot be construed as x, i.e. (28c) is not an appropriate interpretation of (28b).

As we saw, the same reasoning is involved in (16), repeated in (29).

(29) a. She said we should invite Lucie.
b. She (λx (x said we should invite Lucie))
c. *covaluation:*
She (λx (x said we should invite **Lucie**) & (*she* = *Lucie*))
d. *binding-comparison:*
She (λx (x said we should invite **x**) & (*she* = *Lucie*))
(c) ↔ (d), hence *she* ≠ *Lucie*. ((d) ruled out).

In (29b) — the derivation we are considering — *she* is in a configuration to A-bind *Lucie* (i.e. *she* is α and *Lucie* is β, of rule I in (26)). We need to decide whether the covaluation in (29c) is a possible construal. *she* cannot A-bind *Lucie* (since *Lucie* is not a free variable). Hence clause (c) must be checked. To check clause (c), a comparison-representation should be constructed by replacing β (*Lucie*) with a variable A-bound by α. This is done in (29d). Since the binding in (29d) is equivalent to the covaluation in (29c), the later is ruled out. In (16d-g), we checked, for completeness, the option that *Lucie* undergoes QR. I will return to this option shortly, in section 3.3.

In (30a), *Max* is in a configuration to A-bind *him*. Binding is excluded by condition B, and no further operation could change that. We are considering the covaluation interpretation of the derivation, in (30b), which requires checking clause (c).

(30) a. Max admires him.
b. *Covaluation*: Max (λx (x admires **him**) & (*him* = *Max*))
c. *Binding-comparison:* Max (λx (x admires **x**))

The binding comparison is (30c), which is the interpretation the sentence would have received, had binding been permitted. Since (30c) is equivalent to (30b), (30b) is excluded. The same holds when two bound pronouns are covalued, as in (17), repeated in (31).

(31) a. Everyone thinks that he can hear him sing in the bathroom.
Covaluation:
b. Everyone (λx (x thinks that x can hear x sing in the bathroom))
b′. Everyone (λx (x thinks that x (λy (y can hear **x** sing in the bathroom))))
c. *Binding comparison:*
Everyone (λx (x thinks that x (λy (y can hear **y** sing in the bathroom))))

In the derivation (31b) both pronouns are A-bound by *everyone*. Thus they end up covalued, though neither A-binds the other. ((31b) is more fully spelled out in (31b'), where it is obvious that there is no binding.) One occurrence of x is in a configuration to A-bind the other, and condition B prohibits binding, so this covaluation needs checking with clause (c) of rule I. The binding comparison is obtained by replacing the bold-face variable with a variable A-bound by x. Since (31b) is equivalent to the binding comparison in (31c), the derivation is filtered out.[10]

In (28)–(31) covaluation is ruled out by clause (c) of rule I, since it is indistinguishable from illegitimate A-binding. But we saw some examples where the two were distinct, like (20c), repeated in (32).

(32) a. Only he (himself) still thinks that Max is a genius.
b. *Covaluation:* Only he (λy (y thinks **Max** is a genius) & (*he* =*Max*))
c. *Binding comparison:* Only he (λy (y thinks **y** is a genius) & (*he* =*Max*))
(c) is not equivalent to (b), hence (b) is allowed.

As before, the covaluation construal in (32b) is subject to rule I, since *he* is in a configuration to A-bind *Max* (or its trace, if QR applies), and A-binding is excluded. The binding comparison (32c) is, as before, the interpretation the derivation would have had, if binding was permitted. However, (32c) is not equivalent to (32b): The properties

[10] Technically, this specific derivation is also ruled out because the lower x is bound, and cannot be bound again. This result always obtains when two bound variables are covalued under c-command. Thus, rule I happens to rule out (ib) as a possible anaphora construal of (ia).

(i) a. Everyone/Max said that he loves his mother.
b. Everyone/Max (λx (x said that x (λz (z loves x's mother))))
c. Everyone/Max (λx (x said that x (λz (z loves z's mother))))

The difference between (ia) and (31a) is that (ia) still allows the bound anaphora construal in (ic), which in (31a) will be ruled out by condition B.

attributed only to *he* are different, hence the representations have different truth conditions. ((b) could be true if everyone considers himself a genius, as long as no one but Max considers Max a genius. (c) will be false in this situation.) Clause (c) of rule I blocks covaluation only if it is indistinguishable from binding. Hence it does not block it here, and (32b) is allowed. The same reasoning applies in (20), repeated in (33).

(33) a. He is Ralph.
b. *covaluation:* He (λx (x is **Ralph**) & (*he* = *Ralph*))
c. *binding comparison:* He (λx (x is **x**) & (*he* = *Ralph*))
(c) is not equivalent to (b), hence (b) is allowed.

(33b) and (33c) are not equivalent, the second being a tautology. Hence (33b) is allowed. In these examples, the representations are distinguishable since their truth conditions are distinct. In other cases, they may be distinguishable because only one of the properties, but not the other is relevant to previous context. (For discussion, see Reinhart (1983) and Heim (1993).)

3.3. Sentences like (16), repeated, which were traditionally governed by condition C, deserve further attention. We have ruled out already the covaluation in (16c) (in the discussion of (23)). However, as mentioned, it is necessary to check also their derivation with QR, as in (16d), since covaluation in sentences with this PF should be blocked under any derivation. Although the discussion may seem tedious, if just this problem is concerned, it is, in fact, necessary also for more substantial problems I return to in section 4.

(16) a. She said we should invite Lucie.
b. She (λx (x said we should invite Lucie))
c. *Covaluation:* #She (λx (x said we should invite Lucie)) & (*she* = *Lucie*)
With QR:
d. Lucie (λy (she said we should invite y))
d′. Lucie (λy (she (λx (x said we should invite y))))
e. *binding:* *Lucie (λx (she (λx (x said we should invite x))))
Covaluation:
f. #Lucie (λx (x said we should invite x))
g. #Lucie (λx (she said we should invite x & *she* = *Lucie*))
g′. #Lucie (λx (she (λz (z said we should invite x)) & *she* = *Lucie*))

(34) *binding-comparison:* Lucie (λx (x (λz (z said we should invite z))))

With this derivation, there are two covaluation construals that need to be ruled out: (16f,g). (Recall, from the discussion of (13), that although these construals are equivalent, it is necessary to assume that a pronoun can also be covalued directly with the argument (sister) of the λ-predicate, for the contexts of ellipsis.) (16f) is straight forward: *she* is covalued here with the variable x, which it cannot A-bind (since x is bound already). This is, then, the standard strong-over configuration. Since the result is equivalent to the binding-comparison (34), covaluation is ruled out. The question, however, is how (16g) is ruled out. At first glance, the covaluation rule faces a problem here: Technically, *she* is not covalued with the variable *x*, but with *Lucie*. Hence, α and β of this rule must be *Lucie* and *she*. Since *Lucie* is in a configuration to A-bind *she*, clause (b) of rule I requires checking if *Lucie* can A-bind *she* in this derivation. If it can, then covaluation will be wrongly permitted. Note that this problem is not related to the interface aspects of rule I (27), but it arises in the same way for the modified condition C (19). The question for both is whether *Lucie* can A-bind *she* here.

We assumed two circumstances where binding is excluded: when prohibited by logical syntax, or by condition B. Neither of these prevent *Lucie* from A-binding *she* in (16g). However, if binding applies, we obtain the representation (16f), where *she* ends up covalued with *x*. This covaluation is illicit, as we saw already. So, in fact, *Lucie* cannot A-bind *she* here, since no licit interpretation can be obtained for this derivation, if it does. Admittedly, the reasoning involved here is somewhat complex: considering whether a given pronoun can be bound by a given λ-operator (or its sister) requires considering the effect this would have on the rest of the derivation within that λ-predicate. However, in terms of semantic processing, this does not amount to reopening closed constituents: The computation is done within a predicate which is still open (The top λ-predicate contains the free variable *she*, and the lower contains the variable *x*, which is free in that predicate, as can be checked in the fuller representation (16g').)

In the case of (16d), the whole issue could be possibly dismissed, if we assume that QR cannot apply arbitrarily in a given derivation, where it has no effect on the interface, as in the case of (16d) (Chomsky (1995)). However, the same would surface in other derivations where it is obvious that movement has applied, as in (35).

(35) a. #Lucie, she insisted we should invite e (& *she =Lucie*).
b. #She insisted that we should invite Lucie, last week, and not Lili (& *she =Lucie*).

In (35a), topicalization applies overtly. In the elliptic conjunction (35b), QR of *Lucie* must apply for the conjunction to be interpretable. The computation of covaluation would work here precisely as illustrated in (16d). Of course, one could always resort to various formulations of reconstruction in such cases. (On such lines, the covaluation rule checks the relations of *she* and the copy of *Lucie* left in situ, and the derivation is ruled out just as (16).) But in the next section we will see an instance of the same form of computation, where no movement has applied at all.[11]

[11] As I mentioned, the discussion in this section (3.3) is independent of whether we assume the interface rule I, or the modified condition C (19). The two would differ only in cases where covaluation and binding yield distinct interpretations. One such instance is the difference between (i) and (ii).

(i) Max, only he (himself) can stand e. & (*he = Max*)

(ii) #Only Max (himself), he can stand e. & (*he = Max*)

The modified condition C rules (ii) out successfully: As in (16), or (35a), *Max* cannot bind *he* here, (since this leads to illicit covaluation of *he* and the trace). But it would rule out (i), in

4. Further problems of covaluation

Ellipsis contexts provide an anaphora problem which has regained much attention lately, most notably in Fiengo and May (1994) and Fox (1995b).

(36) Max said that he likes his paper, and Lucie did too.

Taking all pronouns in the first conjunct to be anaphoric to *Max*, the second allows three construals. Two of those are the familiar ones: the 'strict' reading, where both pronouns are covalued with *Max*, and then Lucie too said that Max likes Max's paper, and the bound or 'sloppy' reading (under which Lucie said that Lucie likes Lucie's paper). Apart from these, there are two logical options, only one of which can in practice be realized. Since judging the construals requires some processing, it may be easier to view them in their non-elided form below. (The presence of *too* requires usually the same sort of parallelism as required for ellipsis.)

(37) Max said that he likes his paper, and Lucie too said that she likes his paper.

(38) #Max said that he likes his paper, and Lucie too said that he likes her paper.

(37) poses no problem, but (38) is funny, (as long as the destressing and intonation pattern required by *too* is kept). In any case, (36) can be construed as in (37), but not as in (38). For the construal (37) to be generated, the first conjunct must be analyzed as in (39a). The predicates in the two conjuncts are, then, identical (with *Lucie* as the argument in the second conjunct, (39b)).

(36) a. Max said that he likes his paper
b. and Lucie did too.

(39) a. Max (λx (x said that x likes his paper) & (*his* = *Max*))
b. and Lucie (λx (x said that x likes his paper) & (*his* = *Max*))

(40) a. Max (λx (x said that he likes x's paper) & (*he* = *Max*))
b. and Lucie (λx (x said that he likes x's paper) & (*he* = *Max*))

precisely the same way. For rule I, (i) is permitted for the same reason as in previous examples with *only*. In (iii), if *Max* binds *he*, we get the covaluation (iii-b). The sets denoted by the lower λ-predicate in (iii-b) and its binding comparison (iii-c) are different. Hence, we get different interpretations if only x belongs to these sets.

(iii) a. Max (λx (only he (λy (y can stand x)) & *he* = *Max*))
b. Max (λx (only x (λy (y can stand x))))
c. Max (λx (only x (λy (y can stand y))))

(iv) a. Only Max (λx (he (λy (y can stand x) & *he* = *Max*)))
b. Only Max (λx (x (λy (y can stand x))))
c. Only Max (λx (x (λy (y can stand y))))

In (ii), *only* occurs with the top argument — (ii) asserts that only Max is in the λx set. In this case, the sets denoted by the λx-predicate are identical in (iv-b) and the binding comparison (iv-c), i.e. (iv-b) and (iv-c) are equivalent. Hence covaluation is disallowed.

Similarly, to generate the construal (38), the predicate must be construed as in (40). Since (38) is an impossible construal (of 36), this means that something went wrong in (40a). The problem is, then, why (40a) is blocked as an anaphora construal of (36a), while (39a) is permitted.

When anaphora is viewed as a sequence of coindexation of arguments, this is a serious puzzle, since from that perspective, there is only one representation of anaphora for (36a). (This is, apparently, what initiated the study of this problem originally.) But given the analysis here, (40a) turns out an instance of strong-crossover. It is ruled out in the same way as the derivations just discussed in section 3.3.

In the derivation (40a), the lower pronoun, *his*, got bound to the top λx operator, but the higher *he* was left as a free variable. When *Max* is encountered, the question is whether this free pronoun can now be covalued with *Max*. The fuller representation of (40a) is given in (41a). Since *Max* is in a configuration to A-bind *he*, Rule I (or the modified condition C) has to check first whether *Max* can A-bind this pronoun.

(41) a. Max (λx (x said that he (λy (y likes **x**'s paper)) & *he* = *Max*))
b. Max (λx (x said that x (λy (y likes x's paper))))
c. Max (λx (x said that x (λy (y likes y's paper))))

The reasoning proceeds as in (16d) of section 3.3. In principle, *Max* could A-bind the pronoun. But then covaluation is obtained between *he* and *x*, as in (41b). This covaluation is illicit: *he* cannot A-bind *his*, since *his* is already bound. In this case, it is not movement that created the binding of *his*, but an optional step taken in the derivation, namely, the choice to bind it. Nevertheless, *his* is a bound variable, and cannot be A-bound again by *he*. In other words, if the derivation has started as in (41a), no licit operation could apply to derive (41c) from (41a). Since (41c) and (41b) are equivalent, (41b) is ruled out. Hence, in fact, *Max* cannot A-bind *he* and covaluation is ruled out.

In the derivation (39a), by contrast, clause (b) of rule I does not hold, (Both *Max* and the variable *x* can A-bind *his*.) So nothing blocks covaluation. The fuller representation of (39a) is given in (42a). In this derivation, the lower pronoun *his* remained free, while the top one is bound. Since *his* is a free variable it can be A-bound by any c-commanding NP. If *Max* A-binds *his*, covaluation is obtained also between *x* and *his*. But this covaluation is permissible, since *x* could A-bind *his*, as in (42b).

(42) a. Max (λx (x said that x (λy (y likes his paper & *his* = *Max*))))
b. Max (λx (x said that x (λy (y likes y's paper))))

(43) Max likes his paper & (*his* = *Max*).

The situation in (42a) is, thus, analogous to the covaluation construal of (43). As we saw in section 3.2, covaluation is always permitted in such derivations, precisely because binding is not excluded.

Thus, rule I (or the modified condition C in (19)) allows the covaluation derivation (39a) for (36a), but excludes (40a). Probably, the reason why the correlation between (40b) and cross-over configurations could not be formulated in previous approaches is that this configuration is created here by a choice of a particular non-obligatory step in the derivation. The difference between (36a) and the other instances of strong cross-over is that for (36a) it was possible to take different steps in the earlier derivation that would have allowed covaluation (as in (39a)), while in the standard cross-over cases, once the selection from the numeration was made, there is no way to derive a configuration allowing covaluation.

In a way of summary, we may observe that anaphora in sentences like (36) may be obtained under several construals, two of which are ruled out by rule I (as well as by the modified condition C, for such examples):

(44) a. Max said that he likes his paper
b. Max (λx (x said that x (λz (z likes z's paper))))
c. Max (λx (said that he likes his paper & *he =Max* & *his =Max*))
d. Max (λx (x said that he (λz (z likes z's paper)) & *he =Max*))
e. Max (λx (x said that x likes his paper & *his = Max*))
f. #Max (λx (x said that he likes x's paper & *he = Max*))
g. #Max (λx (x said that x likes x's paper))

(44b) is the bound reading, which will lead to the 'sloppy' construal of the ellipsis in (36). In (44c) both pronouns are covalued with *Max*. Both (44c) and (44d) lead to the 'strict' construal of the ellipsis in (36). (44e) is what we discussed in (39a), which leads to the construal (37) of the ellipsis. The next two construals are excluded by rule I: In both, clause (b) or rule I holds, since the lower x is bound. Hence clause (c) rules them out. (Nothing empirical hinges on (44g) being excluded, since in ellipsis contexts it would have led to the same interpretation as (44b). Nevertheless, this is an entailment of rule I).

Fox (1995b) offers a different perspective on why (44f,g) are ruled out. On this view, the problem here is not directly with the covaluation of free variables, but with the binding of variables. This follows the spirit of a partial reformulation of rule I, which was proposed by Heim (1993), and which Fox reformulates again as 'rule H', in (45). This is intended to replace only the condition B aspects of rule I. Heim and Fox assume that along with (45), one needs assume also the traditional condition C of the binding theory. However, Fox argues that once (45) is introduced, it accounts also for the problem under consideration here.

(45) *Rule H*
A variable x cannot be bound by an antecedent α, if a more local antecedent β could bind x yielding an indistinguishable interpretation.

(44) f. #Max (λx (x said that he likes x's paper & *he = Max*))
g. #Max (λx (x said that x likes x's paper))

(Note that (45) assumes the standard syntactic definition of binding as identity of variables). On this formulation, the reason why (44f,g) are ruled out is that the lowest variable x (*his*) is long-distance bound by the top variable x, while the semantic representation that is obtained could have been obtained also if it was bound by a closer antecedent (*he*).[12] Rule H is stated within the economy view of rule I, which, as mentioned, I also shared in the past. As explained in section 3.2, the economy principle that could be behind it, is "get rid of free variables (i.e. close properties) as soon as possible".

Note, first, that for (45) to apply more broadly (and not just to the configurations of type (44)), one has to assume a specific view of discourse anaphora, developed in DRT. As stated, it appears that (45) has nothing to say on why anaphora is blocked in (46b), since there are no bound variables in this representation.

(46) a. [Max is happy....] He really likes him.
b. #He (λx (x likes him) & (*he* = *Max*, *him* = *Max*))

However, Heim assumes that discourse anaphora (covaluation) is also a form of variable binding. Covalued pronouns are bound by a discourse λ- operator, whose argument (sister) is some discourse entity. On this view, *him* in (46) is bound to the discourse entry *Max*, which is more remote than the local antecedent (*he*) that could have bound it with indistinguishable semantic results. Hence it is ruled out. (Since the pronoun is also prohibited by condition B from being bound locally, this derivation has no anaphora construal.)

Once this is assumed, we are back to the problem with the economy view, that I mentioned in section 3. (44e), repeated, turns out to violate rule H, as well.

(44) e. Max (λx (x said that x likes his paper & *his* = *Max*))

The pronoun *his* here is discourse-bound, while the representation is indistinguishable from what we would have obtained had we bound it to the more local x. So the derivation is ruled out. But we saw that, in fact, this construal is allowed. More generally, as stated, rule H does not allow covaluation in sentences like *Max likes his paper*. This is why I concluded that although the rational behind the economy, or locality, view seems reasonable, it is not the type of consideration that the human processor takes into account.

In addition, rule H cannot be extended to capture the full range of condition C effects, for which Heim and Fox assume additional constraints. On the other hand, (44f,g), where rule H seems most successful, are ruled out anyway, by the version of rule I presented here. The locality effects illustrated in (44f,g) are, thus, an entailment of rule I, rather than an independent condition.

5. The psychological reality of rule I

Let us return now to the question whether covaluation is governed by syntactic principles, or by an interface strategy like that formulated in rule I. Recall, first, that with the

[12] The locality solution to the problem of this section has also been proposed under different formulations by Ben-Shalom (1996) and, according to Fox, by Kehler (1993).

present definitions of binding and covaluation, it is possible to unify condition C and the covaluation residue of condition B, while also improving their empirical coverage. The modified condition C and rule I are repeated for the comparison.

(19) *Modified condition C(ovaluation).*
α and β cannot be covalued in a derivation D, if
a. α is in a configuration to A-bind β, and
b. α cannot A-bind β in D.

(26) *Rule I*
α and β cannot be covalued in a derivation D, if
a. α is in a configuration to A-bind β, and
b. α cannot A-bind β in D, and
c. The covaluation interpretation is indistinguishable from what would be obtained if α A-binds β.

On the empirical side, we saw that rule I is closer to capturing the facts: When covaluation under c-command (i.e. in A-binding configuration) is distinct from binding, it is allowed, correctly, by rule I. However, this empirical gain appears to come at a heavy cost. This is visible already in their formulation. Rule I, at least as presented here, just adds a further clause to condition C. But the major cost is in terms of processing: The computation involved in assessing covaluation by rule I is highly complex, as we saw. It requires constructing a reference-set which includes a comparison-representation, and then assessing the semantic relations between the two representations. The syntactic condition, by contrast, requires much simpler computation. Furthermore, the empirical gap between the two approaches is not gigantic. In most cases, they yield, by definition, the same result, differing only in the complex contexts we observed. Under the present formulation, condition C captures also the anaphora in ellipsis contexts discussed in section 4. The difference will show up again only in the cases where covaluation is distinct from binding, discussed for these contexts in Fox (1995b).

Even in contexts where covaluation is distinct from binding, the use of covaluation which goes against condition C is not the most common choice. Except for identity contexts (like *He is Max*), speakers often prefer using an ambiguous derivation, rather than using this option. E.g. though (32a) is possible, as we saw, (47) may be, in practice, the preferred option for expressing the same content. This is so despite the fact that (47) is ambiguous between the two readings observed in (32c,d), while (32a) has only the desired reading (32c).

(32) a. Only he (himself) still thinks that Max is a genius.
Covaluation:
b. Max (λx (only he (λy (y thinks x is a genius)) & (*he =Max*))
c. Max (λx (only x (λy (y thinks x is a genius) (*he = x*)
d. *Binding comparison:* Max (λx (only x (λy (y thinks y is a genius))))
e. (d) is not equivalent to (b,c), hence (b,c) are allowed.

(47) Only Max (himself) still thinks that he is a genius.

On theoretical grounds, it would not be unreasonable, under these circumstances, to dismiss the empirical differences between the two approaches, at least for the time being, and opt for the syntactic approach, which looks less costly. This, indeed, is the line taken by many researchers.

Nevertheless, there is much stronger empirical evidence that a complex strategy, rather than a mechanical syntactic rule is at work here. As just noted, rule I entails that computational complexity is involved in derivations of the type we have been considering, hence one may expect a visible processing cost. (Reinhart (1995) argues that, more generally, whenever an interface strategy based on reference-set computation is at work, there should be evidence for a processing cost.)[13] This got unexpectedly confirmed in studies of the acquisition of anaphora.

Let us, first, recall which steps in rule I involve a processing complexity. If either clause (a) or clause (b) do not hold, the assessment ends here, with nothing complex about it. E.g. we saw that in (48) (=(27)), clause (a) is not met, hence nothing could preclude covaluation. In (49) (=(23)), clause (b) does not hold, since binding could have applied. Hence, anaphora is permitted, whether it is construed as binding, or as covaluation.

(48) Max's mother loves him (*he = Max*).

(49) Max loves his mother.

But if both (a) and (b) hold, assessment must go through clause (c), which is the costly step. These are all and only the derivations that violate the syntactic condition C, as modified in (19).

(50) She said we should invite Lucie.

(51) Max admires him.

(32) Only he (himself) still thinks that Max is a genius.

In these cases, a comparison representation must be constructed and compared to the intended covaluation representation. In terms of processing, it does not matter whether the final verdict of rule I is 'allow', as in (32), or 'disallow' as in (50, 51). In both cases, the decision requires a complex procedure.

This, then, is the crucial difference between condition C and rule I. Condition C requires precisely the same steps in computing covaluation in (48)–(49) and in (50)–(51), while for rule I, computing the second is a much more complex enterprise than computing the first. If there is empirical evidence that the second involves indeed a processing difficulty absent from the first, this is evidence for rule I.

Returning to acquisition, it turns out that children consistently fail on tasks involving step (c) of rule I, and only on those tasks. Many repeated studies have shown that

[13] Note that the shift I made here from viewing rule I as an economy principle to a different sort of (cooperation) strategy does not effect the processing complexity of the procedure. Under either view, two representations must be compared.

children's performance on anaphora in (48) and (49) parallels more or less that of adults. But in the cases of (50–51), they perform at chance level, i.e. guessing, even when the results of individual children are considered. (For a survey of the findings, and references, see Grodzinsky and Reinhart (1993).) Sentences like (32) were not studied. However, the theoretical expectation, if rule I is at work, is that children would have the same difficulties with (32). More generally, that they will have the same difficulty with interpretations ruled in or ruled out by clause (c) of rule I. Other studies (starting with Wexler and Chien (1991)) show that the problem arises only with covaluation construals. When anaphora could only mean binding (e.g. with quantified antecedents), children do not make these mistakes.

Grodzinky and Reinhart (1993) argue that it is impossible to explain these results in the syntactic approach, particularly not the individual guessing pattern, which is rarely found in other cases, and which cannot be expected when a child does not know a rule. But these findings follow if rule I is at work. The processing load posed by clause (c) of rule I is greater than children can execute. Assuming that rule I is innate, they know exactly what they have to do, and try to comply. When the execution gets stuck, they resort to a guess. In personal communication, researchers often report that children are visibly busy with computing in these cases, which is witnessed by facial expressions and a longer response time.

The processing cost of clause (c) of rule I may explain also why, in practice, adults as well don't opt for it too easily. (47) is often preferred over (32), since constructing and comparing representations in a reference-set is a high cost to pay for avoiding ambiguity.

References

Ariel, M. (1990), *Accessing Noun Phrase Antecedents*, London and New York: Routledge.

Ben Shalom, D. (1996) "Dependent and Independent pronouns", Chapter 2 of a UCLA dissertation.

Chomsky, N. (1981) *Lectures on Government and Binding* Foris, Dordrecht.

Chomsky, N. (1995) *The Minimalist Program*, MIT Press, Cambridge, Mass.

Fiengo, R. and R. May (1994) *Indices and Identity*, MIT Press, Cambridge, Mass

Fox, Danny (1995a) "Economy and scope", *Natural Language Semantics*

Fox, Danny (1995b), "Locality in variable binding" to appear in P. Barbosa, D. Fox, P. Hagstom, M. Mcginnis and D. Pesetsky (eds) *Is the best good enough?* MIT Press, and MITWPL.

Grodzinsky, Y. and T. Reinhart (1993) "The innateness of binding and coreference" *Linguistic Inquiry*

Heim, I. (1982), "File change semantics and the familiarity theory of definiteness" in R. Bauerle et al., eds. *Meaning, use and the interpretation of language*. Berlin/New York: de Gruyter, p. 164–189.

Heim, I. (1993) "Anaphora and semantic interpretation: A reinterpretation of Reinharts Approach" SfS-Report-07–93, University of Tubingen. Reprinted in Uli Sauerland and O. Percus, Eds., *The interpretative Tract, MIT working papers in Linguistics* Vol. 25. Cambridge, Mass: MITWPL, 1998.

Higginbotham, J. (1983) "Logical Form, binding and nominals" *Linguistic Inquiry*, 14, 395–420.

Keenan, E. (1971) "Names, quantifiers and a solution to the sloppy identity problem", *Papers in Linguistics*, vol. 4, no. 2.

Kehler, A. (1993) "A discourse copying algorithm for ellipsis and anaphora" proceedings of EACL.

Lasnik, H. (1976) "Remarks on coreference" *Linguistic Analysis*, vol 2, no. 1.

Langacker, R. (1966) "On pronominalization and the chain of command" in W. Reibel and S. Schane (eds) *Modern Studies in English*, Prentice Hall, Englewood Cliffs, New Jersey.

McCawley, J. (1979) "Presuppositions and discourse structure" in Oh, C.K and D.A. Dinneen (Eds) *Presuppositions. Syntax and Semantics vol 11*, Academic Press, New York.

Prince, E. (1981), "Towards a taxonomy of Given-New information", in P. Cole (ed.) *Radical Pragmatics*, New York: Academic Press, p. 233–255

Reinhart, T. (1976), *The syntactic domain of anaphora*, PhD dissertation, MIT, Cambridge, Mass. Distributed by MITWPL.

Reinhart, T. (1983), *Anaphora and semantic interpretation* Croom-Helm; Chicago University press.

Reinhart, T. (1995), *Interface Strategies*, OTS Working Papers in Linguistics, University of Utrecht.

Reinhart, T. and E. Reuland (1993) "Reflexivity" *Linguistic Inquiry* 24.4 p. 26183–321.

Ristad, Eric (1992) *Computational structure of natural language*, Ph.D. dissertation, MIT, distributed by MITWPL.

Wexler, K. and Y. C. Chien (1991) "Children's knowledge of locality conditions n binding as evidence for the modularity of syntax and pragmatics" *Language Acquisition*, 1, 225–295