Some Governing Category Computations

Here's the core of the LGB binding theory: (Unless stated otherwise, the example numbers reference the Overview of Chomsky's Binding Theory HO.)

The LGB Binding Theory: [All of this demands that every referential type NP has an index (as already suggested in Chomsky (1965), and that a trace has the same index as the NP it is a trace of.]

(52) α is bound by β if and only if α and β are coindexed and β c-commands α.
[X c-commands Y iff every Z dominating X also dominates Y (and X ≠ Y).]

(53)a α is A-bound by β iff β binds α and β is in an A-position
b α is (A-)free if and only if it is not (A-)bound.

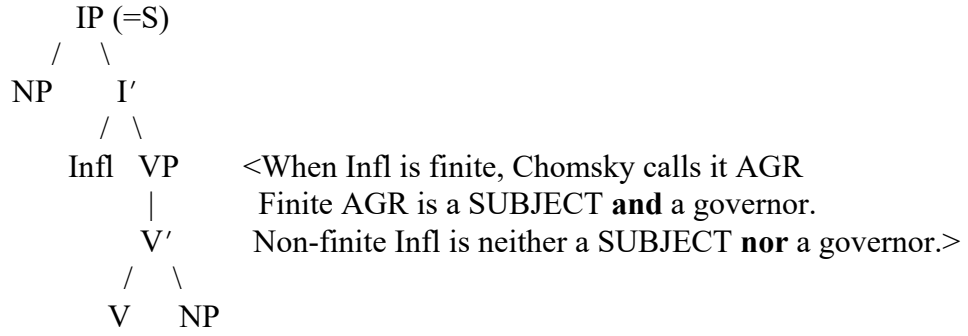
(54)A An anaphor is (A-)bound in its GC. 'Condition A'
B A pronominal is (A-)free in its GC. 'Condition B' (A descendent of RI)
C An R-expression (fully lexical NP, or variable) is (A-) free. 'Condition C' (A descendent of the noncoreference rule of Lasnik (1976))

(55) α is a governing category for β if and only if α is the minimal category [i.e., XP] containing β, a governor of β, and a SUBJECT accessible to β.

(56)a SUBJECT = AGR in a finite clause (i.e., finite Infl); NP of S in an infinitival; NP of NP in an NP.
b 'Accessible' (preliminary version). X is accessible to Y only if X m-commands Y (and X ≠ Y)
c X m-commands Y iff every ZP dominating X also dominates Y

<Anaphors: reflexives; reciprocals; traces of A-movement; PRO
Pronominals: standard personal pronouns like *she* and *him*; PRO
Thus, PRO must obey **both** Cond. A **and** Cond. B (almost a contradiction)

Basic clausal structure:



<When Infl is finite, Chomsky calls it AGR
Finite AGR is a SUBJECT **and** a governor.
Non-finite Infl is neither a SUBJECT **nor** a governor.>

Some examples:

(63) *John$_i$ AGR believes [$_{IP}$ Mary to like himself$_i$] Cond. A is violated.
<The GC for *himself* is the embedded clause, with *like* the governor of *himself* and *Mary* the accessible SUBJECT, since it m-commands *himself*. (Note that the matrix clause also contains all of these things (since it contains the embedded clause and 'contain' is transitive), but the GC is the **minimal** XP that does.) There is no A-binder for *himself* **in the embedded IP.**>

(64) John$_i$ AGR believes [$_{IP}$ Mary to like him$_i$] Cond. B is satisfied
<Exactly the same GC for *him* in (64) as for *himself* in (63). This satisfies Cond. B, because, even though *him* is A-bound, it is **not** A-bound in its GC.>

(58)a John$_i$ AGR believes [$_{IP}$ himself$_i$ to be clever] Cond. A is satisfied
<The GC for *himself* is the entire matrix IP, with *believe* the governor of *himself* and AGR the accessible SUBJECT, since it m-commands *himself*. There is no smaller XP that contains all those things. There is no governor of *himself* in the embedded IP, nor is there a SUBJECT accessible to *himself*. (*himself* is a SUBJECT but is not accessible to itself since m-command is irreflexive.>

(59) *John$_i$ AGR believes [him$_i$ to be clever] Cond. B is violated
<The GC for *him* is the entire matrix IP, with *believe* the governor of *him* and AGR the accessible SUBJECT, since it m-commands *him*. There is no smaller XP that contains all those things. There is no governor of *him* in the embedded IP, nor is there a SUBJECT accessible to *him*. (*him* is a SUBJECT but is not accessible to itself since m-command is irreflexive.>

(58)a′ John$_i$ AGR injured himself$_i$ Cond. A is satisfied
<The GC of *himself* is essentially just the same as in (58), which is, of course, the whole point of the approach. For BT, ECM subjects behave as if they are in the matrix clause. The GC for *himself* is the entire IP with *injured* the governor of *himself* and AGR the accessible SUBJECT, since it m-commands *himself*.>

(59)′ John$_i$ AGR injured him$_i$ Cond. B is violated
<The GC of *him* is essentially just the same as in (59), which is, of course, the whole point of the approach. For BT, ECM subjects behave as if they are in the matrix clause. The GC for *him* is the entire IP, with *injured* the governor of *him* and AGR the accessible SUBJECT, since it m-commands *him*.>

(57)a *John$_i$ AGR believes [$_{CP}$ (that) [$_{IP}$ himself$_i$ AGR is clever]] Cond. A is violated
<The GC for *himself* is the embedded IP, with embedded AGR the governor of *himself* and also the accessible SUBJECT, since it m-commands *himself*. (Note that the matrix clause also contains all of these things (since it contains the embedded clause and 'contain' is transitive), but the GC is the **minimal** XP that does.) There is no A-binder for *himself* **in the embedded IP.**>

(60) John$_i$ AGR believes [$_{CP}$ (that) [$_{IP}$ he$_i$ AGR is clever]] Cond. B is satisfied

<The GC for *him* is the embedded IP, with embedded AGR the governor of *him* and also the accessible SUBJECT, since it m-commands *him*. (Note that the matrix clause also contains all of these things (since it contains the embedded clause and 'contain' is transitive), but the GC is the **minimal** XP that does.) There is no A-binder for *him* **in the embedded IP.**>

(82)a *I AGR like PRO

*I$_i$ AGR like PRO$_i$ Cond. B is violated

<The GC for PRO is the entire IP, with *like* the governor of PRO and AGR the accessible SUBJECT, since it m-commands PRO.>

*I$_i$ AGR like PRO$_j$ Cond. A is violated

(82)c *John AGR believes [$_{IP}$ PRO to be intelligent]

<The GC for PRO is the entire matrix IP, with *believes* the governor of PRO and AGR the accessible SUBJECT, since it m-commands PRO.>

*John$_i$ ..... PRO$_i$ Cond. B is violated. PRO is A-bound in its GC.

*John$_i$ ..... PRO$_j$ Cond. A is violated. PRO is **not** A-bound in its GC.

(81) John AGR tried [$_{CP}$ [$_{IP}$ PRO to leave]]

<PRO in (81) has no GC. It has an accessible SUBJECT, AGR, which m-commands it. But it has no governor, because, by stipulation, neither non-finite Infl nor null C is a governor. By virtue of having no GC, it vacuously satisfies **both** Cond. A and Cond. B, no matter how it is indexed. Conversely, whenever it has a GC, it will necessarily violate Cond. A **or** Cond. B (not both!), depending on how it is indexed. This conclusion is called the PRO Theorem.>