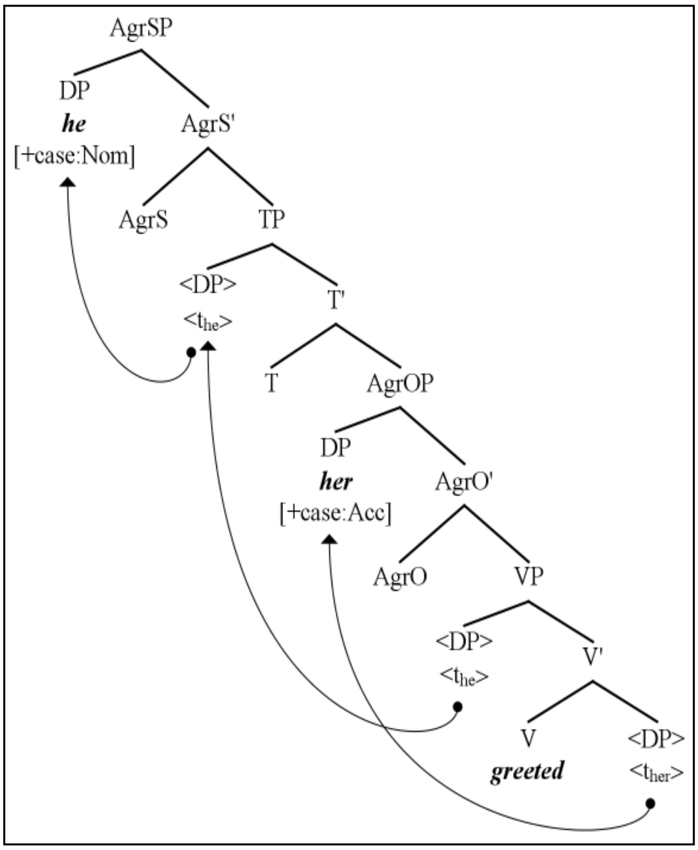
NOTES ON SESSION 3 (10.11.2008)

This was a somewhat difficult session, to say the least. I don't know whether I am just slow or whether I just like maximal formal explicitness, some of the things just were pretty hard to swallow. The technicality of the notions such as **domain**, and **chain** is rigid and if you forget how these terms work at the bottom line, it becomes very difficult to follow minimalist argumentation. So I will here mainly try to rely on **Hornstein et al. (2005)**.[1]

What is the fuss about **domains** and **chains**? Let me take a look at the tree representation below: This is a more or less explicit tree depiction of the algorithm described on p. 146 (10a-e). This should be the



sentence on its way to becoming: *He greeted her.* The problem with this is that the "moving paths", so to speak, of the two arguments he and her cross each other. For minimality reasons this is not allowed. Hornstein et al. say:

> "However, such interventions go against the standard GB-wisdom that movement is restricted by minimality considerations, which, roughly speaking, prevent a given element from moving across another element 'of the same type'". (p. 142)

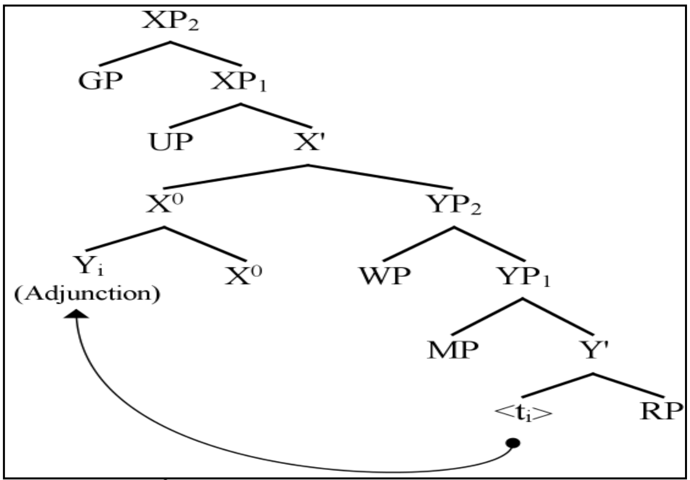
So what we have to do is look for some alternative notions and come up with general ideas on how to allow this **Argument-over-Argument** movement. On p. 144, some examples are given that support the assumption of minimality proposed in the general MP.

> "In short, minimality seems like a conceptually congenial condition on grammatical operations from a minimalist perspective as it encodes the kind of least effort sentiments that minimalism is exploring". (p. 144)

The problem is just that an intransitive sentence such as *He greeted her.* would already violate the "do not cross requirement".

Containment, Domination, Minimal Domain, Extended Minimal Domain

The tree below is p. 149 (19):



Alright now[2] let me tease out the essential truths about these notions – placing focal emphasis on **domination** and **domain**.

> **Def: Minimal Domain (MinD):**
> The Minimal Domain of α, or MinD(α), is the set of categories **immediately contained** or **immediately dominated** by projections of the head α, excluding projections of α.

It follows:
MinD[X^0, X^0]={GP, UP, YP_2, YP_1, WP}.
MinD[$Y_{\text{(prior to movement)}}$]={WP, MP, RP}.

Head movement of Y_i extends its minimal domain while forming the chain CH=[Y_i, t_i]. Thus:

MinD[CH=[Y_i, t_i]]=
MinD[$Y_{\text{(prior to movement)}}$]∩MinD[$X^0$, X^0] but not
{$_{\text{(projections of Y)}}$ YP_2, YP_1, Y} it follows
{GP, UP, $WP_{\text{(two times)}}$, MP, RP}.

Question: Is it a problem that MinD(X) and MinD(Y) have one element (WP) in common? I think there

---

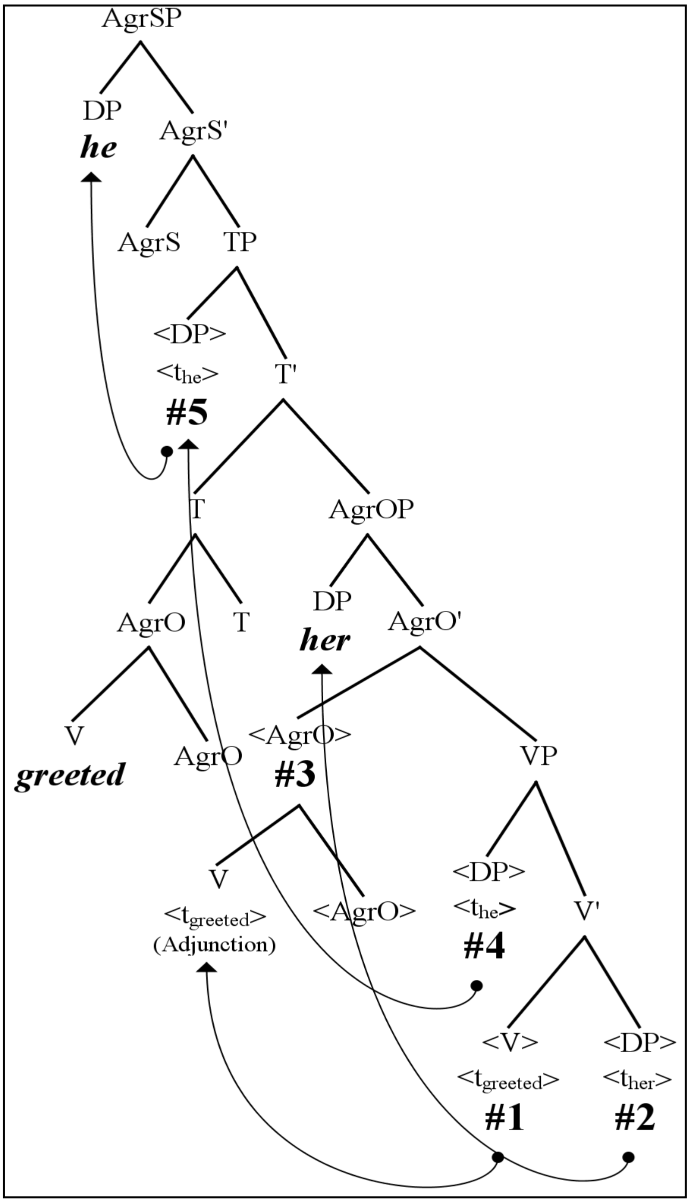
[1] See Neven's bibliography.

[2] Runnin's been done for today…

should be no problem – it could just be due to coincidence.

Remember that we originally wanted to account for some movements which would (under minimality considerations) be ruled out. The thing is now that you can move an element γ to either an element α or β if α and β are *in the same minimal domain* (p. 151). α and β are then said to be **equidistant** form the element that wants to move, i.e. γ.

We are now finally in a position to account for at least a simple transitive construction such as *He greeted her.* The solution strictly based on Hornstein et al. (p. 151-3) is given below:



This is a most stepwise fashion tree. Let me make clear what seems to happen - at least for me.

#1: V adjoins to AgrO, forming the chain CH=[greeted, $t_{greeted}$], with the minimal domain being MinD(greeted)∩MinD(AgrO)=MinD(Chain)= {DP, DP, DP}
Since all this is in the same minimal domain, it makes DP-over-DP movement possible – due to a *"flattened structure"* provided by the minmal domain. Note that Hornstein et al. think that <t_{her}> is an element of the set MinD(Chain), but can this be? It can only be formed *after* moving the DP *her*. I will therefore assume here that the elements in MinD(Chain) are the *categories* of the potential moving sites and *not* the actual lexical items.
#2: DP *her* crosses DP *he* – in minimal domain.
#3: Since DP he cannot move out its minimal domain, move the whole complex AgrO (including the adjoined element V), forming the chain CH=[AgrO, T]. Now, since AgrO is complex (it contains an adjunct), we seem to move a complex element creating an even more complex one: [[$V_{Adjunct}$, AgrO] T], the chain being CH=[[$V_{Adjunct}$, AgrO] t_{AgrO}]
I am not sure whether one could code it like this, but the MinD($CH_{complex}$)={DP, DP, VP}, providing the landing site for step number four.
#4: The DP he can now move and cross the DP *her* – being in the same minimal domain.
#5: Has actually nothing to do with minimality effects or domains of some sort. The DP *he* just has to receive its proper Nominative case at [Spec, AgrSP].
Q.E.D.

Is it not a good idea to label the steps you are doing in a tree? I simply don't know whether it would save you so much space if you did it via long lists of bracketed representations.

To me, all this seems to boil down to a couple of very basic operations:

(1) A-over-A movement only possible in MinDs.
(2) Check for A movement, check for MinDs.
(3) Create MinDs via moving elements, providing moving (landing) sites.

Questions:

(1) Are there elements that do not form domains or chains?
(2) If I merely have to head-adjoin elements from below in the tree in order to form a domain and chain, then I could almost license every type of A movement.
(3) What are the constraints on domain and chain formation?