Some working definitions (second version)

A grammar G consists of a pair of a set of lexical elements L and a set of operations O:

G = <L, O>

A derivation on a numeration D_N is a pair:

a set of lexical elements drawn from L, called the Numeration N, and
an ordered n-tuple of phrase markers PM:

D_N = <N, <PM_1, … , PM_n>>

A derivation D_N is said to *converge* iff

1. PM_n contains no unchecked uninterpretable (*u*) features
2. PM_n contains no unchecked strong (*) features
3. PM_n contains no unvalued (:__) features
4. All elements in the Numeration have been Merged
5. For each adjacent pair of phrase markers <PM_k, PM_{k+1}> in D_N, there is an operation Ω such that Ω applied to PM_k yields PM_{k+1}.

Feature structures:

A lexical item *LI* has the following feature structure, given in three equivalent notations:

	category features	*inflectional features*	*selectional features*
LI	[ ]	[ ]	[ ]

LI [CAT […] ; INFL […] ; SEL […]]

LI [… ; … ; …]

Some examples:

see	V [-aux]	[ ]	*u*D | <TH>

v_{trans}	CAT	*v*		
		[-aux]		
	INFL	Infl:__		
		φ __		
		V*		
		*u*Case:ACC		
	SEL	*u*V, *u*D		
		|		
		<AG>		

T_{pres}	éT	ù	éClause-type:__	ù	[*u*v, *u*D*]
	ê[-aux]	ú	ë ϕ __	û	
	ëInfl: Pres	û			
dog	éN	ù	[Case:__]		[ ]
	ë ϕ 3sm	û			

Operations:

Merge(α, β)

For any syntactic objects α, β, where α bears an unchecked selectional feature F, and β bears a matching categorial feature F,

call α the *head* and
let α = { γ, { α, β}}
call γ the *label* (or *projection*) and
let F be checked (written ~~F~~), and
let γ = α ∩ , where is the set of all unchecked non-inflectional features

---

Adjoin(α, β)

For any syntactic objects α, β, where neither α nor β has any unchecked selectional feature,

call α the *host*, and
let α = { γ, { α, β}}
call γ the *label* (or *projection*) and
let γ = α

Move$_{\textbf{head}}$(X*, Y) (F* on probe) (read: 'X moves to Y')

If X is a head with a strong inflectional feature F*, Y a head with a matching feature F, and X c-commands Y, then

let X = {X, {Y, X}} and
let F* = ~~F*~~, and
let Y = <Y>

Move$_{\textbf{head}}$(X, Y*) (F* on goal) (read: 'X moves to Y')

If Y is a head with a strong inflectional feature F*, X a head with a matching feature F, and X c-commands Y, then

let X = {X, {Y, X}} and
let F* = ~~F*~~, and
let Y = <Y>

Agree(X,Y;F) (read: ‘X triggers agreement on Y in F’ or ‘Y agrees with X in F’)

For any syntactic objects X and Y, where X bears a feature F with value Val(F) and Y bears a matching unvalued inflectional feature F´: , and either X c-commands Y or Y c-commands X,

let Val(F´) = Val(F) and
if F is uninterpretable, let F = ~~F~~

Move$_{\textbf{phrase}}$(Y, X*) (F* on probe) (read: ‘Y moves to specXP’)

If X is a projection with a strong feature F*, Y a maximal projection with a matching feature F, and X contains Y, then

let X = {X, {Y, X}} and
let F* = ~~F*~~, and
let Y = <Y>

Move$_{\textbf{phrase}}$(Y*, X) (F* on goal) (read: ‘Y moves to specXP’)

If Y is a maximal projection with a strong feature F*, X a projection with a matching feature F, and X contains Y, then

let X = {X, {Y, X}} and
let F* = ~~F*~~, and
let Y = <Y>

Move$_{\textbf{phrase}}$(Y*, X*) (F* on both probe and goal) (read: ‘Y moves to specXP’)

If X is a projection with a strong feature F*, Y a maximal projection with a matching feature F*, and X contains Y, then

let X = {X, {Y, X}} and
let F* = ~~F*~~, and
let Y = <Y>

to$_{raising}$	éT	ù	éıClause-type:__	ù	[*u*v, *u*D*]
	ê[-aux]	ú	ëıϕ: __	û	
	ëInfl: Inf	û			