1 Functional structure

Here are some details about the individual heads and the kinds of features we expect to find on them.

1.1 C

C determines the clause type. We differentiate two clause types: declarative and interrogative. (In later syntax, other types would be differentiated: relative clauses, imperatives, etc.)

The C in a declarative clause has a [clause-type:decl] feature, and no other relevant features.

The C in an interrogative clause has a [clause-type:Q] feature and a [uwh*] feature.

There is a special case here: When a main clause C values the [uclause-type:] feature of T as Q, it values it as strong ([uclause-type:Q*]), which causes T to raise to C. (But, note that if a *wh*-word is the subject, the [uclause-type:] feature of T will have already been valued by the time C is merged, so in that case C won't participate, and T won't raise to C.)

1.2 T

T determines the tense of a clause, and is also the facilitator of subject agreement and nominative case assignment.

No matter what kind of clause we're considering, T has a [uD*] feature (referred to as the "EPP"), which will force a DP to occupy the specifier.

When a clause is nonfinite, T has a [tense:inf] feature for normal infinitive clauses. In untensed CP adjuncts (like *After PRO having eaten lunch*), the *ing* is represented as a T with a [tense:ing] feature.

When a clause is finite (not nonfinite), T has a feature that corresponds to the tense ([tense:past] or [tense:pres]), as well as an uninterpretable feature for assignment of nominative case [ucase:nom] and an unvalued uninterpretable feature for agreeing with the subject [uϕ:].

T can value a lower [uInfl:] feature by providing the combined values of its tense feature and its ϕ-feature values. This entails that T's [uϕ:] feature must be valued first, before T can value a lower [uInfl:] feature. The valuation of a lower [uInfl:] feature could result in, e.g., [uInfl:pres3sg] on *v*.

There is a special case when a finite T values a [uInfl:] feature on a head that has an [Aux] feature. In that situation, the [uInfl:] feature is "valued as strong," which is written as, e.g., [uInfl:pres3sg*], and requires that the head whose feature was valued moves up to head-adjoin to T before the feature is checked. Note that this does *not* happen when T is nonfinite (when it has a [tense:inf] feature).

The pronunciation of [uInfl:past] on another inflectible element is the past tense form, e.g., *wrote*, *might*, or *kicked*.

The pronunciation of [uInfl:pres] on another inflectible element is the agreeing present tense form, e.g., *write(s)*, *might*, or *kick(s)*.

The pronunciation of [uInfl:ing] on another inflectible element is the *–ing*-suffixed form, e.g., *writing*, *having*, or *kicking*.

Finite T is responsible for assigning nominative case, so finite T has a [ucase:nom] feature. This will generally value and check the [ucase:] feature on the DP that ultimately moves to the specifier of TP to satisfy the EPP ([uD*]). Nonfinite T does not have a [ucase:] feature, with the exception of the T that

represents *ing* in untensed adjunct CPs (such as *After PRO having eaten lunch*) [*u*case:null] (for checking the case of PRO).

When T is finite, it has a [*u*clause-type:] feature that is valued by a higher C. If it is valued by a main clause interrogative C, it is valued as strong and T will move to head-adjoin to C. Nonfinite T does not have this feature.

1.3   M

M is the category of modals (*can*, *could*, *may*, *might*, *shall*, *should*, *will*, *would*).

Modals are auxiliaries, so they have the feature [Aux].

Modals are considered to be inflected for tense; for example, the modal *may* is the present form, which has *might* as its past form. So, modals have a [*u*Infl:] feature.

The category feature of modals can value a lower [*u*Infl:] feature, which results in a [~~*u*Infl:~~M] feature on, e.g., *v*.

There is no obvious connection between the "past-tense form" (e.g., *would*) and the semantics of the past tense. There is also no future tense in English: *will* is considered to be a present tense modal.

There is a special case with infinitive clauses: The *to* that occurs with the infinitive is considered to be a modal, but is compatible only with a T that has a [tense:inf] feature. Note too that *to* is exceptional with respect to having its [*u*Infl:] feature valued by T—it is not "valued as strong" and does not raise.

The pronunciation of [~~*u*Infl:~~M] on another inflectible element is the bare form of the verb, e.g., *write*, *be*, or *kick*.

1.4   *v*

v can be considered to be what turns a VP into a true verb, there is always a *v* above the VP. There are a couple of different kinds of *v*, depending on properties of the V below it and on whether the *v*P is the complement of Pass.

No matter what kind of *v* we have, there is always a [*u*Infl:] feature (representing the fact that the verb needs to be inflected) and a [*u*V*] feature (causing the V to move up to *v*).

For verbs with an Agent, there is v_{AGENT}, and for verbs with an Experiencer, there is v_{EXP}. Both of these cause a DP to be Merged into the specifier, so both have a [*u*D*].

For verbs with neither an Agent nor an Experiencer (unaccusative or passive verbs), *v* does not have [*u*D*] feature.

For verbs with both an Agent and a Theme, *v* has a [*u*case:acc] feature, to assign accusative case to the Theme.

There is a special case when the *v*P is the complement of Pass (in a passive clause). In this configuration, even a verb that would normally have an Agent has the unaccusative *v*—no Agent in the structure and no accusative case assigned, so no [*u*D*] and no [*u*case:acc].

1.5   Pass

Pass appears above *v* in passive clauses. It requires of its complement *v*P that there be no Agent (and therefore no accusative case assignment). So, in a clause with Pass, the *v* has neither a [*u*D*] feature nor a [*u*case:acc] feature.

The pronunciation of Pass is a form of *be*, but it needs to be inflected. So Pass always has a [*u*Infl:] feature.

The category feature of Pass is able to value a [uInfl:] feature below it (leading to a [~~uInfl:~~Pass] feature on v).

Pass is in the class of auxiliaries, so it has an [Aux] feature.

The pronunciation of [~~uInfl:~~Pass] on another inflectible element is the *-en* form (the "past participle" or "passive participle" or "perfect participle" form), e.g., *written*, *been*, or *kicked*.

1.6   Prog, Pref

Prog and Perf are auxiliaries, like Pass, so each has the [Aux] feature. Both need to be inflected, so each has a feature [uInfl:]. Both can value the [uInfl:] feature of a lower item as well, e.g., valuing the feature of v as [~~uInfl:~~Prog] or as [~~uInfl:~~Perf].

The pronunciation of Prog is a form of *be*, and the pronunciation of Perf is a form of *have*.

The pronunciation of [~~uInfl:~~Prog] on another inflectible element is the *-ing* form (the "present participle" or "progressive participle" form), e.g., *writing*, *being*, or *kicking*.

The pronunciation of [~~uInfl:~~Perf] on another inflectible element is the *-en* form (the "past participle" or "passive participle" or "perfect participle" form), e.g., *written*, *been*, or *kicked*.

1.7   D

All Ds have a [$u$case:] feature—all DPs need case.

D also agrees with the lower n in ϕ-features. So D has a [$u\phi$:] feature. The difference between *a(n)* and $\emptyset_{PL}$ (the plural indefinite determiner), *this* and *these*, and *that* and *those* are all considered to be essentially inflections based on number agreement.

There is a special case here: Pronouns are DPs without any internal structure. So, *I*, *me*, *my*, *he*, *him*, *his*, *she*, *her*, *they*, *them*, *their*, *it*, *its*, as well as *who(m)* and *what* are simple Ds. They have intrinsic ϕ-features (e.g., *him* has [ϕ:masc3sg]), rather than an unvalued [$u\phi$:] feature taking its value via agreement.

When there is a possessor or a subject inside, the head of D is $\emptyset_{GEN}$, which has a [$u$case:gen*] feature. This will both assign genitive case to the nearest DP inside its complement, and cause it to move into the specifier of DP.

For gerunds like *John's quickly eating the sandwich*, we hypothesize that the structure has a D with a vP complement (rather than the usual nP). Here we need to assume that $\emptyset_{GEN}$ can value the [uInfl:] feature below as D ([~~uInfl:~~D]), which is pronounced as the *–ing*-suffixed form.

1.8   *n*

n can be considered to be what turns a NP into a true noun, there is always an n above the NP. The n is the nominal analog to v. There are a couple of different kinds of n, depending on properties of the N below it, and on certain properties of the D in whose complement the nP is.

There is a [uN*] feature on every kind of n, which causes N to move up to head-adjoin to n.

There are two basic cases for n: one is where the NP is essentially a VP, with θ-roles to assign (these are the deverbal nouns, like *destruction*), and the other is the more basic common nouns (like *hat*).

For deverbal nouns, most of the same kinds of considerations apply to n as they do for v. If there is an Agent, there is n_{AGENT}, and for verbs with an Experiencer, there is n_{EXP}. Both of these cause a DP to be Merged into the specifier, so both have a [uD*]. For verbs with neither an Agent nor an Experiencer (unaccusative or passive verbs), n does not have [uD*] feature.

For deverbal nouns with both an Agent and a Theme, n will have a [$u$case:of] feature, to assign "of"-case to the Theme.

There is a special case here, akin to the passive: When the D that takes the *n*P as its complement is overt (e.g., *the*), then the *n* has neither a [*u*D*] feature nor a [*u*case:of] feature.

In order to subsume both deverbal and common nouns, we need to assume that the number feature at least (therefore the ϕ-features) are features of *n*. So, the difference between *book* and *books* is not in the N, but rather in the *n*. *n* might have the feature [ϕ:sg], for example. For deverbal nouns, the same holds—if the DP refers to several events (*the permutations of the set*), then *n* will have a plural feature ([ϕ:pl]).

1.9   Poss

Poss introduces a possession relationship, and is found inside DPs. The Possessor θ-role is assigned to its specifier and the Possessee θ-role is assigned to its complement.

2   The Hierarchy of Projections

The Hierarchy of Projections tells the computational system what things should be Merged next when there is no strong feature left to check. You are allowed to do XP-adjunction (Adjoin) prior to doing the next Merge called for by the hierarchy. In the Hierarchy of Projections, things listed in parentheses are optional—for example, if there is a Pass on the workbench, then Pass and *v*P are Merged when *v*P is finished. If there are no auxiliaries, then T and *v*P are Merged when *v*P is finished.

Hierarchy of Projections
(C >) T > (Neg >) (M >) (Perf >) (Prog >) (Pass >) *v* > V
D > (Poss >) *n* > N

When the motivation for Merge is the Hierarchy of Projections, feature checking between the two things Merged does not occur. For example, *v* is next thing in the Hierarchy after V, so once the VP is finished, it is Merged with *v*. *v* always has a [*u*V*] feature, but this is not checked at this step. A second step must be performed in order to check the [*u*V*] feature, which is to move V up to head-adjoin to *v* (close enough to allow the [*u*V*] to be checked).

3   UTAH

The **Uniformity of Theta Assignment Hypothesis** says that θ-roles are determined by the structural position in which an argument is initially Merged into the structure. The concise statement of the UTAH is below, along with examples demonstrating all of the cases.

Uniformity of Theta Assignment Hypothesis (UTAH)	
AGENT	DP daughter of v_{AGENT}P/n_{AGENT}P
EXPERIENCER	DP daughter of v_{EXP}P/n_{EXP}P
THEME	DP daughter of VP/NP
GOAL	PP sister of V/N
PROPOSITION	CP/TP sister of V/N
POSSESSEE	*n*P sister of Poss
POSSESSOR	DP daughter of PossP
EXPERIENCER	PP daughter of VP

4 Some tree fragments for specific cases

Unergative verb
v does not assign case
dance, jog, shout

```
        vP
       /  \
     DP     v′
    AGENT   / \
         V+v   <VP>
```
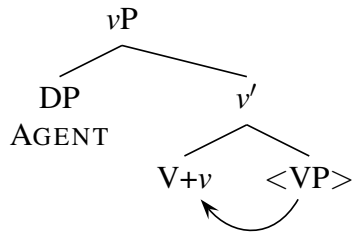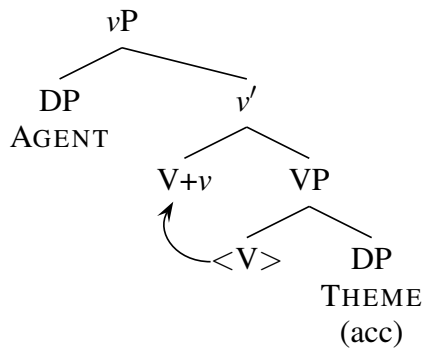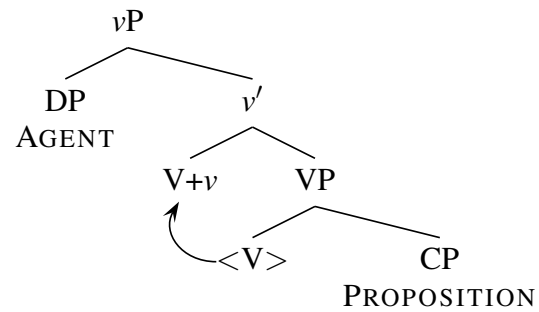
Unaccusative verb
v does not assign case
fall, sink, melt

```
        vP
       /  \
    V+v     VP
          /    \
       <V>      DP
               THEME
```
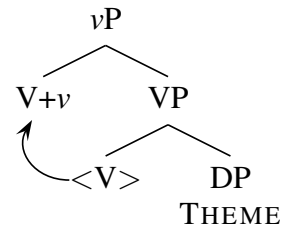
Transitive verb (with THEME)
v assigns ACC to Theme.
kick, eat, see

```
        vP
       /  \
     DP     v′
    AGENT  /  \
        V+v    VP
            /     \
         <V>       DP
                 THEME
                  (acc)
```

Transitive verb (with finite CP PROPOSITION)
v does not assign case.
say, hear, claim

```
        vP
       /  \
     DP     v′
    AGENT  /  \
        V+v    VP
            /     \
         <V>       CP
                PROPOSITION
```

Transitive verb (with TP PROPOSITION)
v assigns ACC to subject of Proposition.
consider, want (ECM; TP is non-finite)

```
        vP
       /  \
     DP     v′
    AGENT  /  \
        V+v    VP
            /     \
         <V>    TP PROPOSITION
                  /    \
                DP      T′
               (acc)
```
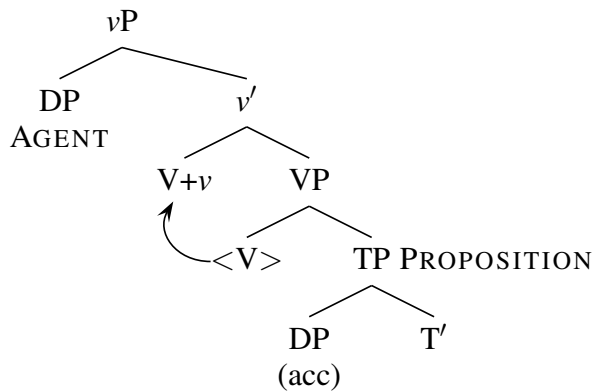
Subject control verb (with PROPOSITION)
C is $\emptyset_{\text{NULL}}$, assigns null case to lower PRO.
want, promise, ask

```
        vP
       /  \
     DP     v′
    AGENT  /  \
        V+v    VP
            /     \
         <V>    CP PROPOSITION
                  /    \
                 C      TP
             ∅NULL     /  \
                     DP    T′
                    PRO
                   (null)
```
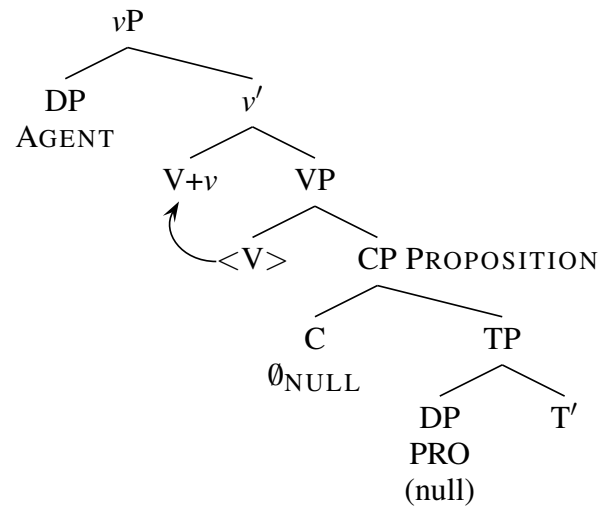
Ditransitive verb (with GOAL)
v assigns ACC to Theme
put, send, give

```
                vP
          ┌─────┴─────┐
         DP           v′
        AGENT    ┌─────┴─────┐
               V+v          VP
                ▲      ┌─────┴─────┐
                │     DP           V′
                │    THEME    ┌─────┴─────┐
                │    (acc)   <V>          PP
                └──────────────┘         GOAL
```
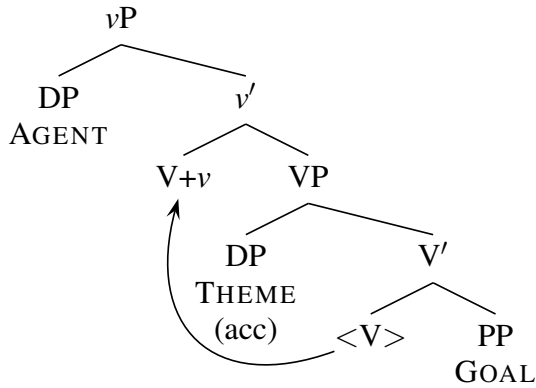
Ditransitive verb (with PROPOSITION)
v assigns ACC to Theme
tell, persuade, promise

```
                vP
          ┌─────┴─────┐
         DP           v′
        AGENT    ┌─────┴─────┐
               V+v          VP
                ▲      ┌─────┴─────┐
                │     DP           V′
                │    THEME    ┌─────┴─────┐
                │    (acc)   <V>        CP/TP
                └──────────────┘      PROPOSITION
```
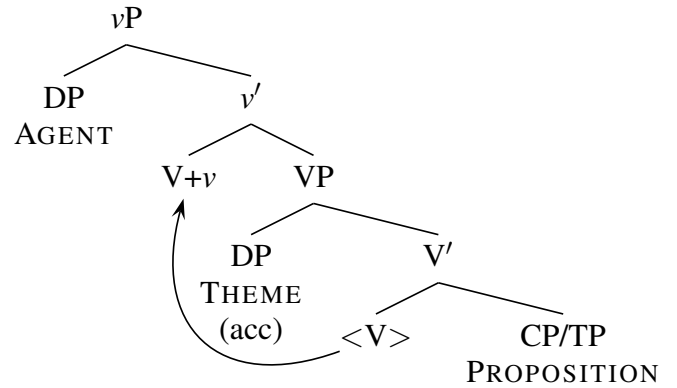
Raising verb
v does not assign case
seem, appear

```
           vP
      ┌────┴────┐
    V+v        VP
     ▲    ┌────┴────┐
     │   <V>      CP/TP
     └────┘     PROPOSITION
```
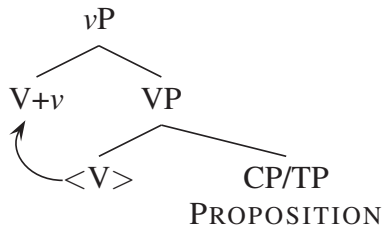
Raising verb (with EXPERIENCER)
v does not assign case
seem to me

```
              vP
        ┌─────┴─────┐
      V+v           VP
       ▲      ┌─────┴─────┐
       │     PP           V′
       │  EXPERIENCER ┌───┴───┐
       │             <V>     CP/TP
       └──────────────┘    PROPOSITION
```
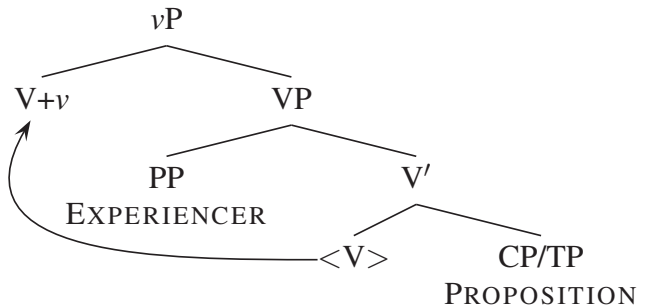
DP (with POSSESSOR, POSSESSEE)
∅GEN assigns genitive case to SpecDP
my hat

```
           DP
      ┌────┴────┐
   DPpsr        D′
   (gen)   ┌────┴────┐
     ▲     D        PossP
     │   ∅GEN   ┌────┴────┐
     │       <DPpsr>     Poss′
     │       POSSESSOR ┌──┴──┐
     │               Poss   nP
     └────────────┘    ┌────┴────┐
                      N+n      <NP>
                       ▲     POSSESSEE
                       └───────┘
```
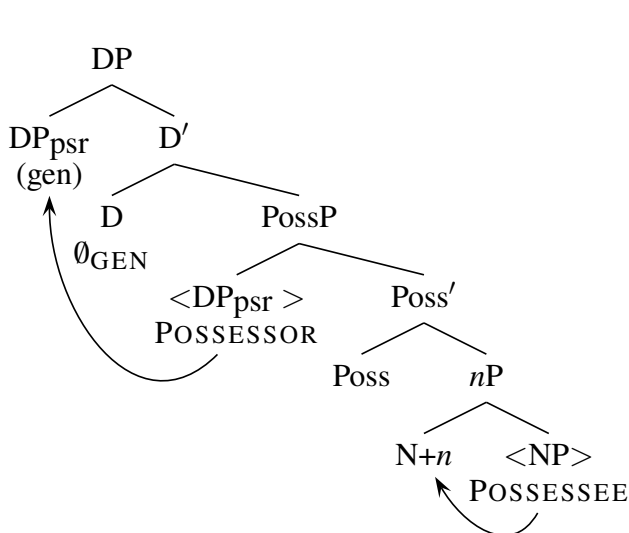
Deverbal DP (with AGENT, THEME)
∅GEN assigns genitive case to SpecDP
n assigns *of*-case to Theme
my destruction of it

```
           DP
      ┌────┴────┐
    DPag        D′
   (gen)   ┌────┴────┐
     ▲     D         nP
     │   ∅GEN   ┌────┴────┐
     │       <DPag>       n′
     │       AGENT   ┌────┴────┐
     │             N+n        NP
     └────────┘     ▲    ┌────┴────┐
                    │   <N>        DP
                    └────┘       THEME
                                  (of)
```