Review from last time…

The structure of the grammar:  
\[ DS \leftarrow \text{phrase structure rules} \]
\[ \leftarrow \text{movement rules} \]

surface structure (abstract) \[\rightarrow\] SS \[\leftarrow\] more movement rules

“phonetic form” \[\rightarrow\] PF \[\leftarrow\] “logical form” (meaning)

Movement comes in four different kinds, each with its own conditions:

**Movement**

- **XP-movement**
  - topicalization
  - *wh*-movement

- **X-movement**
  - passive raising
  - extrapolation heavy NP shift

**Wh-movement**

Move *wh*-XP to Comp when:

1. **Comp is [+Q]** if *wh*-word does not move further.
2. **the *wh*-XP is not inside a [+Q] S’** (except the one containing the Comp).
   - *(wh-island constraint)*
3. **the *wh*-XP is not inside a complex NP.**
   - *(complex noun phrase condition)*
4. **there is no nearer Comp.**
   - *(cyclicity condition, locality condition)*

(1) a. Which book did John buy \( t_1 \) ?  
   b. I know which book John bought \( t_1 \).

(2) a. …the speech that Mary made.
   b. …[\( \text{NP} \) the speech, \( [_{S} \text{Op}, \text{that} \text{[}_{S} \text{Mary made} \ t_1 \text{]} \text{]} \].}
NP-movement
Move NP to an empty subject position, as long as
(i) NP is not contained in a Tensed S (other than the target S).
(ii) NP is not separated from the targeted position by a specified subject.

(3) a. Mary solved the problem.
b. The problem was solved (by Mary). \(\text{passive}\)

(4) a. It seems (that) Mary has solved the problem.
b. Mary seems to have solved the problem. \(\text{raising}\)

Extraposition (a rightward movement rule)
Move XP and attach it as the rightmost constituent of VP.

(5) a. [Details of a secret plan to finance the rebels] have emerged.
b. Details have emerged of a secret plan to finance the rebels. \(\text{extraposition}\)

Heavy NP shift (a subcase of Extraposition)
Move a ‘heavy’ NP and attach it as the rightmost constituent of VP.

(6) a. Mary read \(\text{[NP all the books she had borrowed]}\).
b. ? Mary returned \(\text{[NP all the books she had borrowed]}\) to the library.
c. Mary returned \(t_i\) to the library \(\text{[NP all the books she had borrowed]}\).

(7) a. Mary read Aspects.
b. Mary returned Aspects to the library.
c. * Mary returned \(t_i\) to the library \(\text{[NP Aspects]}\). \(\text{Aspects not “heavy enough”}\)

X-bar theory
We strive for a lean theory—we don’t want redundancy.

(8) a. hit: \([—\text{NP}]\)
b. smile: \([—]\)
c. think: \([—\text{S’}]\)
d. give: \([—\text{NP PP}]\)
e. rely: \([—\text{PP}]\)

And we require that when we insert lexical items into the tree, we only insert them where their subcategorization frames are satisfied. (Lexical Insertion Rule)

For this to work, we also need phrase structure rules which can provide trees which meet the subcategorization frames, so:
(9)  
   a. VP → V NP
   b. VP → V
   c. VP → V S′
   d. VP → V NP PP
   e. VP → V PP

(We can write them with parentheses but that’s really just shorthand)

So we have: VERBS IN GENERAL CAN BE FOLLOWED BY S′
   and: THE VERB think IN PARTICULAR CAN BE FOLLOWED BY S′

If we know the second one, the first one doesn’t really tell us anything about the structure
of a VP where the verb is think.

The goal will be to set down some very general principles about how trees are put
together (eliminating the more complicated PS rules) along with constraints based on the
subcategory frames of lexical items.

Heads

One thing that all our PS rules have in common is that NPs have an obligatory N
constituent, VPs have an obligatory V constituent, APs have an obligatory A
constituent—the head of the phrase. In a sense, how we know it is a verb phrase (a VP) is
that it is centered around a verb.

(10) XP → … X …

So (10) is true of all of our existing PS rules, and moreover, we can restrict our theory by
saying it must be true of any tree built by our grammar.

Specifiers and complements

In something like

(11) This solution to the problem

(12) \[ \text{NP}_1 \quad \text{NP}_2 \quad \text{N} \quad \text{PP} \]

This solution to the problem
The NP₂, N, and PP are each sisters of each other—there is no *structural* (hierarchical) distinction between them. The only constituent containing two contains all three.

(13)  
   a. I like [this [solution to the problem]], but not [that [one]].  
   b. * I like [[this solution] to problem #1], but not [[one] to problem #2].

It looks like *one* can be used in place of the …N PP… sequence, but not the …NP₂ N… sequence—suggesting it is a constituent.

Plus, the relation between *solution* and *to the problem* is different from the relation between *solution* and *This* (or *Mary’s* or *The*)—*to the problem* tells you what kind of *solution* it is, *This* specifies which one. We can’t read that off the structure in (12). We can solve both problems at once, though:

(14)  

```
       NP
           SPEC
             Det
               This
               N  PP
               solution  to the problem
```

The *complement* is the sister to the head—here, the PP. The *specifier* is sister to the new node N’ (N-bar), a daughter of the maximal projection NP.

(15)  

```
       NP
           specifier
             N’
               N  complement
```

**one-replacement**

Replace N’ with *one*.

(16)  

The big red book of poems
(17) NP
    SPEC
    Det
    the
    AP
    big
    A
    AP
    red
    N
    book
    PP
    of poems

(18) a. I bought this \[N^\prime \text{big red book of poems}\], not that \[N^\prime \text{one}\].
b. I bought the big \[N^\prime \text{red book of poems}\], not the little \[N^\prime \text{one}\].
c. I bought the big red \[N^\prime \text{book of poems}\], not the big blue \[N^\prime \text{one}\].
d. ?* I bought the big red book of poems, not the little red one of cartoons.

It looks like \(N^\prime\) can iterate. And it doesn’t have to be to the left, either.

(19) The book of poems on the table by the window

(20) NP
    SPEC
    the
    N\prime
    PP
    …by the window
    N\prime
    PP
    …on the table
    N
    book
    PP
    of poems

(21) I saw this \[N^\prime \text{book of poems on the table by the window}\], not that \[\text{one}\].
(22) I saw the \[N^\prime \text{book of poems on the table}\] by the window, not the \[\text{one}\] by the door.
(23) I saw the \[N^\prime \text{book of poems}\] on the table by the window, not the \[\text{one}\] on the floor.

(24) The big ugly red book of poems on the table by the window in the cabin

So we can write the rules of NP construction like this:

(25) NP \rightarrow (Spec) N\prime \hspace{1cm} \textit{SPECIFIER} (daughter of NP and sister of N\prime)
    N\prime \rightarrow N (PP) \hspace{1cm} \textit{COMPLEMENT} (sister of N)
    N\prime \rightarrow N\prime PP or XP N\prime \hspace{1cm} \textit{ADJUNCT} (sister of N\prime)

where XP can be either AP or PP at least.
The main claim of X-bar theory is that this structure (specifiers, heads, complements, adjuncts) is sufficient to characterize not only NP but XP—any kind of phrase.

(26) \[ \text{XP} \rightarrow \text{Spec* X'} \]
\[ \text{X'} \rightarrow \text{X ZP*} \]
\[ \text{X'} \rightarrow \text{YP X'} \text{ (or X' YP)} \]

(27)
```
XP
   Spec
   X'
   specifier
   YP
   X'
   adjunct
       X
   ZP
   head complement
```

(28) a. I ate lunch at Burger King.
b. I ate lunch and drank cola at Burger King.
c. I ate lunch at Burger King and Bill [did (so)] at McDonald’s.
d. I ate lunch at the Burger King on Comm. Ave. in Boston.

(29) a. \[ \text{NP} \] [the suggestion \[ S' \] that John should resign] is absurd.
b. \[ \text{NP} \] [the suggestion \[ S' \] \( Op_i \) that John made \( t_i \)] is absurd.
c. [That [one]] is absurd.
d. [The [one] that Bill made] is absurd.

(30) a. \[ \text{NP} \]
```
   Det
   the
   N'
   suggestion
   that
   John
   should resign
```
b. \[ \text{NP} \]
```
   Det
   the
   N'
   suggestion
   N
   \( Op_i \) that
   John
   made \( t_i \)
```

(31)
```
VP
   V'
   V'
   PP
   at
   Burger King
   eat
   NP
   lunch
```
(32)  a. Mary quickly fixed the car.
b. Mary fixed the car quickly.

(33)  a.  
   \[
   \text{VP} \\
   \text{V'} \\
   \text{Adv} \quad \text{V'} \\
   \text{quickly} \quad \text{fixed} \\
   \text{NP} \quad \text{the car}
   \]

   b.  
   \[
   \text{VP} \\
   \text{V'} \\
   \text{Adv} \quad \text{V'} \\
   \text{quickly} \quad \text{fixed} \\
   \text{NP} \quad \text{the car}
   \]

(34)  a.  I am [AP very [A tired of reruns]] but Bill is [AP less [A so]].
b. John knocked it [PP right [P off the table] and [P into the trash] ].

---

**S vs. IP**

Headedness says this can’t be right:

(35)  S → NP Aux VP

Neither NP nor VP can be the head—they’re maximal projections. The head must be Aux.

We are going to call it INFL (for inflection), since it is the home of Tense.

“S” is now “InflP” or “IP”, headed by the INFL node; sentences now look like this:

(36)  
   \[
   \text{IP} \\
   \text{Spec} \\
   \text{I'} \\
   \text{subject} \\
   \text{I} \\
   \text{tense} \\
   \text{VP} \\
   \ldots
   \]

English has a limited amount of subject agreement—

(37)  a.  I **watch** television.  
     first person, singular (1sg)
b.  You **watch** television.  
     second person, singular (2sg)
c.  We **watch** television.  
     first person, plural (1pl)
d.  You (all) **watch** television.  
     second person, plural (2pl)
e.  He **watches** television.  
     third person, singular (3sg)
f.  They **watch** television.  
     third person, plural (3pl)
A special suffix appears on the verb when the subject is 3sg. In many languages, this is more dramatic, with a different suffix for each person/number combination. We will refer to this as agreement. Infl will be responsible not only for Tense but for Agreement as well. That I will have tense features and agreement features.

Agreement features often go by the obscure name of “ϕ-features”.

The way we ensure that the subject agrees with the verb is by imposing this condition on our trees:

**Spec-Head Agreement**
A head (X) and its specifier (SpecXP) must agree in the relevant features.

So, for IP, where the subject has ϕ-features (e.g., John has [3] and [sg] features), I must also have [3] and [sg] features (and will also have a tense feature like [present] as well).

(38)

```
(38) IP
     \ Spec
      \ John [3sg]
       \ I [3sg]
            \ VP [present] ...
```

We can also see Spec-Head agreement in the NP system:

(39) a. NP b. NP

```
(39) a. NP
     \ Det
      \ these [pl]
       \ N books [pl]

(39) b. NP
     \ Det
      \ this [sg]
       \ N book [sg]
```
Affix hopping.

The problem: Tense, a feature of Infl, ends up being realized together V (e.g., solved).

Two options: Move solve to attach to Aux.
Move Tense to attach to V.

Looking at adverbs, generated at the edge of VP, tells us that the second is right.

(40) a. John [vp cleverly avoided Bill ].
b. John [vp rarely visited Bill ].
c. * John avoided [vp cleverly Bill ].
d. * John visited [vp rarely Bill ].

(41)

If Tense is not the only thing in Infl, this movement does not happen:

(42) a. John should avoid Bill.
b. John might (cleverly) avoid Bill.
c. John could avoid Bill.
d. * John could avoided Bill.
Affix Hopping
Move Infl to V provided Infl does not dominate a modal.

Do-support

John did not avoid Bill.
With negation, Affix Hopping does not occur, but past tense is realized on did.
Two things we need to do:  
• Amend the Affix Hopping rule
  • Generate do.

not intervenes between [+Past] and V, and so Affix Hopping cannot merge them.

Affix Hopping (second statement)
Move Infl to V provided Infl does not dominate a Modal or have a NegP complement.

Do-support
Insert do to support stranded Tense.

Both Affix Hopping and Do-support provide “morphological support” for Tense.
Tense is an affix, it can’t stand alone.
V-raising

Auxiliary verbs *have and be*:

(46) a. John is not happy.
    b. John has not visited Bill.
    c. * John did not be happy.
    d. * John did not have visited Bill.

*Have and be* are unique among English verbs, the verb raises to Infl (instead of AH).

To distinguish *have and be* from other verbs, we will say they are [+AUX], whereas other verbs are [–AUX].

If Tense needs support, a [+AUX] verb will raise—but won’t if not needed.

(47) a. John must be happy.
    b. John might have visited Bill.

**V-raising**

Raise V to Infl, provided:

i) V has the feature specification [+AUX]

ii) Infl does not dominate a modal.

**Affix Hopping**

Move Tense (from Infl to V), provided:

i) V has the feature specification [–AUX]

ii) Infl does not dominate a modal or have a NegP complement.

iii) VP does not dominate a V with the feature specification [+AUX].

(48) a. * Mary be worked at home.
This is getting complicated. V-raising and Affix Hopping have almost complementary conditions. We can simplify this a little bit:

**Goal**: Support Tense.

**Preferences**: Prefer V-raising to Affix Hopping

Prefer Affix Hopping to Do-support.

- If Tense is already supported (if there is a modal in Infl), goal is met.
- If Tense is still unsupported and V-raising can be applied (there is a [+AUX] V), apply it; goal is met.
- If Tense is still unsupported and Affix Hopping can be applied (Infl doesn’t have a NegP complement), apply it; goal is met.
- If Tense is still unsupported, apply Do-support (as a “last resort”).

**V-Raising**

Raise V to Infl provided V has the feature specification [+AUX].

**Affix Hopping**

Move Infl to V, provided Infl does not have a NegP complement.

\[(49)\]

<table>
<thead>
<tr>
<th>a.</th>
<th>* John did avoid Bill.</th>
</tr>
</thead>
<tbody>
<tr>
<td>b.</td>
<td>John <em>did</em> avoid Bill.</td>
</tr>
</tbody>
</table>

**S’ vs. CP**

The other part of our phrase structure rules that doesn’t fit into X-bar structures yet is S’

\[(50)\]  
S’ → Comp IP

It is fairly clear that Comp must be the head of S’, so we will now call S’ “CP” (Complementizer Phrase).

Headedness guarantees that Comp is not optional—if it is not pronounced it is because it is null.

\[(51)\]  
```
    CP
   /   |
Spec C’
   |
  C  IP
  |
  \ ...
```
This gives us two “Comp positions”: SpecCP and C.
We have seen many cases where complex phrases (XPs) have been found in Spec.
Wh-movement moves wh-XPs to “Comp”—
   If this is moving to C, it is the only case we know of where we move an XP
to attach to a head.

But moving it to SpecCP would make it like the other specifiers.
In fact, we will constrain our theory of structure even further by adding:

**Only maximal projections (XPs) are found in Spec position**

So wh-movement moves a wh-XP to SpecCP (where C is [+Q]).

In wh-questions, there are two things we have to move
   (Infl to Comp, and wh-XP to Comp).

Infl (Aux) is not an XP, so it can’t be moving to SpecCP; moreover, SpecCP is “filled”
by the wh-XP. But we still have C…

---

**I-raising (‘Aux-raising’)**

(52)  a. Mary will solve the problem.
      b. Will Mary solve the problem?

(53)  a. You can drive my car.
      b. Can you drive my car?

To make a question, the Infl element is moved over the subject—to C.
I-raising
Move Infl to C, when C has the feature specification [+Q].

This happens in *wh*-questions too…

(55)  a. Which problem can Bill solve?

   b. DS: \([CP [+Q] [IP \text{Bill} [I \text{can}] [VP \text{solve which problem}] ]]\)

   c. SS: \([CP [NP \text{which problem}] [I \text{can}] [IP \text{Bill} [I \text{can}] [VP \text{solve which problem}] ]]\)

But, I-raising doesn’t apply in embedded questions:

(56)  a. I wonder which problem Bill can solve.

(b. I wonder if Bill can solve this problem.

I-raising (second statement)
Move I to C, provided
i) C has the feature specification [+Q]
   ii) C is situated in a root clause.

Some examples where several things occur…
(57)  a.  Was John at home?
    b.  DS:  \([\text{CP} \ [+Q] \ [\text{IP} \ \text{John} \ [\text{I} \ \text{Tense} \ [\text{VP} \ \text{be at home}]]] ]\)
    c.  VR:  \([\text{CP} \ [+Q] \ [\text{IP} \ \text{John} \ [\text{I} \ [\text{V} \ \text{be}] \ [\text{I} \ \text{Tense} \ [\text{VP} \ \text{t at home}]]] ]\)
        \[\[\text{t}\]\]
    d.  IR:  \([\text{CP} \ [\text{C} \ [+Q] \ [\text{I} \ [\text{V} \ \text{be}] \ [\text{I} \ \text{I} \ \text{Tense} \ [\text{I} \ \text{S} \ \text{John} \ [\text{I} \ \text{I} \ \text{VP} \ \text{t at home}]]] ]\]
        \[\[\text{t}\]\]

(58)  a.  Did John solve the problem?
    b.  DS:  \([\text{CP} \ [+Q] \ [\text{IP} \ \text{John} \ [\text{I} \ \text{Tense} \ [\text{VP} \ \text{solve the problem}]]] ]\)
    c.  IR:  \([\text{CP} \ [\text{C} \ [+Q] \ [\text{I} \ [\text{Tense} \ [\text{I} \ \text{I} \ \text{IP} \ \text{John} \ [\text{I} \ \text{I} \ \text{VP} \ \text{solve the problem}]]] ]\]
        \[\[\text{t}\]\]
    d.  do:  \([\text{CP} \ [\text{C} \ [+Q] \ [\text{Tense} \ [\text{I} \ \text{do} \ [\text{I} \ \text{I} \ \text{IP} \ \text{John} \ [\text{I} \ \text{I} \ \text{VP} \ \text{solve the problem}]]] ]\]

*Note:  Affix Hopping apparently couldn’t apply here, even though there was no Neg…*

**Affix Hopping**

Move Infl to V, provided

i)  Infl does not dominate a Modal or have a NegP complement

ii)  V has the feature specification \([-\text{AUX}])

iii) VP does not dominate a V with the feature specification \([+\text{AUX}])

iv)  Infl is adjacent to VP which dominates the targeted V

or

**Affix Hopping**

Move Infl to V, provided

i)  Infl does not have a NegP complement.

ii)  Infl is adjacent to VP which dominates the targeted V

(We could even conflate these two if NegP counts as making Infl “non-adjacent to VP”).

*Notice:  Affix Hopping can only apply after I-raising.*

The generally accepted structure of *wh*-questions these days is something like this:
It is obligatory in English to move the *wh*-XP into SpecCP. We can encode this thusly:

**[+Q]-CP Principle**
A [+Q]-CP must have a [+wh] specifier.

(60)  
\[
\begin{align*}
&\text{a. I wonder who Mary saw.} \\
&\text{b. } * \text{ I wonder Mary saw who.}
\end{align*}
\]

(61)  
\[
\begin{align*}
&\text{a. I wonder } [\text{CP} \text{ whether C } [\text{IP} \text{ Mary left}]]. \\
&\text{b. I wonder } [\text{CP} \text{ Op if } [\text{IP} \text{ Mary left}]]. \\
&\text{c. } * \text{ I wonder } [\text{CP} \text{ whether if } [\text{IP} \text{ Mary left}]]. \quad *\text{DFC}
\end{align*}
\]

Like relative clauses (except here *Op is really like silent whether rather than silent which*)

Structure Preservation and types of movement

**Structure-Preserving Hypothesis**
Transformations are structure-preserving (the results still conform to \(X'\)-theory).

*Substitution movements* are movements to a position which is provided by \(X'\) structure but generated empty (specifically, specifiers).

Passives:

(62)  
\[
\begin{align*}
&\text{a. } [\text{IP} e [t, I [\text{VP} \text{ was } [\text{VP} \text{ solved the problem }]]]] \\
&\text{b. } [\text{IP} \text{ the problem }, t, I [\text{VP} \text{ was } [\text{VP} \text{ solved } t_i]]]
\end{align*}
\]
**Wh-questions:**

(63) a. I wonder… \[CP e [\_C C[+_Q] [IP John bought what ]]\]

   b. I wonder… \[CP what [\_C C[+_Q] [IP John bought t_i ]]\]

Some other things we’ve looked at are trickier—

**Topicalization?**

(64) \[I know… CP\]

\[Spec C'\]

\[C (that)\]

\[NP_i this problem\]

\[Spec NP I\]

\[I'\]

\[VP\]

\[V'\]

\[V solve NP t_i\]

This is not a possible X-bar structure. NP_i needs to be attached to IP in a way which is different from the way Spec is.

We can extend our treatment of I’ a little bit to IP—The idea being that we kind of “stretch out” the IP node so we can attach the moved NP to it.

(65) \[I know… CP\]

\[Spec C'\]

\[C (that)\]

\[NP_i this problem\]

\[Spec NP I\]

\[I'\]

\[VP\]

\[V'\]

\[V solve NP t_i\]

We say that here *this problem* is still dominated by IP.
However, *this problem* is not a sister to *I’* (like SpecIP is).

Movement that attaches to a “stretched out category” like this is *adjunction movement*.

Extraposition and Heavy NP shift are also adjunction movements, right- adjoining to VP.

**Head movement**

We will also consider *head movement* (movement of terminal categories) to be adjunction movement, where the moving head adjoins to the target head.

So, where *I* moves to *C*, we have a structure like:

(66)

```
  C
 / \   /
C  IP
  /  \
I   Spec
     /  \ 
    t   VP
     \  
      ... 
```

**The Projection Principle**

We should impose another requirement on our trees in order to make sure they obey the subcategorization requirements. Like Structure Preservation, we will also require that this holds all the time.

**Projection Principle**

Representations at each syntactic level (LF, DS, SS) are projected from the lexicon, in that they observe the subcategorization properties of lexical items.

Actually, this isn’t really *all the time*, because it is only “checked” at three points—DS, before movement starts; SS, after the overt movement is over; and LF, after all the movement is over. In this way it is different from Structure Preservation, which constrains every transformation in a derivation.

**Small clauses**

(67) a. John considers [IP [NP Bill] [I’ to [VP be [AP incompetent]]]].
    b. John considers Bill incompetent.

What is the structure of the second one?

There’s no *to*, and there can be no *be*, suggesting that there is no Infl under *considers*.
The current analysis of this is that Bill is in this case in the specifier of AP, serving as “the subject of the adjective phrase”

(68) \[
\begin{array}{c}
\text{Spec} \\
\text{NP} \\
\text{Bill}
\end{array}
\xrightarrow{\text{A}}
\begin{array}{c}
\text{A'} \\
\text{incompetent}
\end{array}
\] AP

This seems to be possible with other phrases as well—

(69) a. The captain expects the drunken sailor off the ship (immediately).
b. John made Bill read the whole book.

(70) \[
\begin{array}{c}
\text{Spec} \\
\text{NP} \\
\text{the drunken sailors}
\end{array}
\xrightarrow{\text{P'}}
\begin{array}{c}
\text{P} \\
\text{off} \\
\text{NP} \\
\text{the ship}
\end{array}
\] PP

(71) \[
\begin{array}{c}
\text{Spec} \\
\text{NP} \\
\text{Bill}
\end{array}
\xrightarrow{\text{V'}}
\begin{array}{c}
\text{V} \\
\text{read} \\
\text{NP} \\
\text{the whole book}
\end{array}
\] VP

Subjects, IP, and the Extended Projection Principle

Like CP in questions, it turns out that IP seems to require a specifier (unlike most of the other categories). This is encoded as follows:

**Extended Projection Principle (EPP)**

Clauses must have a subject (that is, ‘The specifier of IP must be filled.’).

(The name really has very little to do with the conceptual content—you might be better off just forgetting about the words and refer to this as the “EPP”).
One bit of evidence comes from the raising cases; where we can’t raise the embedded subject (for example, when it violates the TSC), we have to put a meaningless element (expletive) *it* in subject position.

(72)  
  a. *(It) seems that Mary has solved the problem.
  b. Mary seems *t_1* to have solved the problem.

A type of sentence of this sort which has gotten a great deal of attention are *there*-constructions like:

(73)  
  a. A unicorn is in the garden.
  b. *(There) is a unicorn in the garden.

In (73b), the meaningless element *there* is required. What makes it required is the EPP.

Where is *a unicorn* in (73a)? A good guess, given the small clauses discussed earlier, is that it is the subject of a PP:

(74) 

And, for (73b), perhaps *there* fills SpecIP, satisfying the EPP, which allows *a unicorn* to stay in SpecPP.

One other thing: in (75), there are two IPs. Where is the subject of the lower one?

(75)  

[IP Mary seems [IP *t_1* to have solved the problem]].

*Answer:* The trace counts—further evidence that we need traces. SpecIP is filled, but with the *trace* of *Mary.*
Nonfinite clauses and PRO

(76)  
  a. John tried \([_{\text{CP}} \  {_{\text{IP}}} \ e \  \text{to leave}]}\).
  b. John persuaded Bill \([_{\text{CP}} \  {_{\text{IP}}} \ e \  \text{to leave}]}\).
  c. It is difficult \([_{\text{CP}} \  {_{\text{IP}}} \ e \  \text{to leave}]}\).

Here there appears to be no subject (*EPP) in the lower (nonfinite) clause.
The subject is not a trace. *Try, persuade, be difficult* are not raising verbs (like *seem*).
Moreover, the embedded agent (the one leaving) is *John* in (76a), *Bill* in (76b), and *someone/anyone* in (76c).

So:
  • There must be something there (by the EPP).
  • We can’t hear it.
  • It can’t be a trace of movement.

→ It must be an unpronounced element (and like a pronoun).

(77)  
  a. John, tried \([_{\text{CP}} \  {_{\text{IP}}} \ PRO \ i \ \text{to leave}]}\).
  b. John persuaded Bill, \([_{\text{CP}} \  {_{\text{IP}}} \ PRO \ i \ \text{to leave}]}\).
  c. It is difficult \([_{\text{CP}} \  {_{\text{IP}}} \ PRO_{arb} \ \text{to leave}]}\).

**Subject control**  
**Object control**  
**Arbitrary control**

PRO only appears in nonfinite clauses.
Nonfinite clauses lack tense and agreement features in Infl:

(78)  
  IP
  /\  
  Spec  I'  
  \ /  
  I  VP
  /   (to) \  
  \   \  

Not all nonfinite clauses have *to*—gerundive clauses are nonfinite but lack *to*:

(79)  
  a. John dislikes \([_{\text{CP}} \  {_{\text{IP}}} \ PRO \ \text{eating in public}]}\).
  b. \([_{\text{CP}} \  {_{\text{IP}}} \ PRO \ \text{reading detective stories}]}\] is fun.

Binary branching and the VP shell (kind of a preview)

As a result of the changes we’ve made, most of the trees have only binary branches.
We can take this a step further and *require* that our trees have no more than binary branches:

**Binary Branching**
A node can dominate at most two branches.

If we do this, there is a particular “problem” we have with ditransitive verbs like *put*. 
These verbs subcategorize for two arguments—both are required. But how can a subcategorization frame like this ever be satisfied if we only allow binary branching?

Perhaps Binary Branching is wrong… But if we hold on to it, it will force us to a different analysis of ditransitive verbs like put.

This makes a number of claims which we aren’t in a position to confront; the object for example is not the sister of anything—it’s in SpecVP.