Week 15a. Reprise

The lexicon

- The lexicon is where it all begins, where the component parts of a sentence come from.
- A sentence is a number of lexical items, arranged.
- Lexical items have certain properties, or features. Some are nouns, for example. Some are wh-words, some are quantifiers, some are determiners.
- Every head we see in our trees came from the lexicon. So, C, I, v, these are also in the lexicon, components from which we build sentences.

Minimalism

- As we try to determine what the properties of this grammatical system are, we should assume as little as we can get away with.
- Any language-like system that is going to create hierarchical structure is going to need something that takes two (or more, but let’s say that “two is simpler than any other number”) things and puts them together into something eligible for further combinations.
- So, the machine that builds the trees has at least the operation Merge.

Starting over

- Let’s take a tour of the system from the beginning, to help get a better “wide-angle” view of how everything fits together and to try to tie up the loose ends.
- This is the final statement of where we are, what you should take as the end result.

The lexicon

- Since phonological realization and even aspects of meaning can be considered to be properties of lexical items, really what a lexical item is is a bunch of features, bundled together. A thing, with properties.
- Some of the properties lexical items have are in the form of requirements, which need to be satisfied by the time the syntactic structure is finished (the LF tree).

Our model of grammar

- A structure is built by starting with some lexical items on the workbench, that are assembled by using Merge and Adjoin between objects, and Copy/Move to move things inside an object to its edge.
X'-theory

- A phrase is a syntactic object formed by combining (merging) two syntactic objects, with the properties inherited from one of them (the head of the phrase).
- A word is a syntactic object.

θ-theory

- The number of participants that predicates require are at the heart of θ-theory.
- The θ-criterion says that:
  - Every θ-role required by a predicate must be assigned to some argument.
  - No argument can play more than one role.
  - No argument can be inserted superfluously; every argument must get a θ-role.
- θ-roles are assigned by heads to a specifier or a complement. That’s two per head, maximum.

Common thematic relations

- Agent: initiator or doer in the event
- Theme: affected by the event, or undergoes the action
  - Bill kicked the ball.
- Experiencer: feels or perceives the event
  - Bill likes pizza.
- Proposition: a statement, can be true/false.
  - Bill said that he likes pizza.
- Goal:
  - Bill ran to Copley Square.
  - Bill gave the book to Mary. (Recipient)

θ-theory

- Lexical items can be classified in terms of being predicates or arguments.
- Predicates require something else for the computation of their meaning. They might be considered to be relations between the facts of the world (“truth”) and some other entity.
- Arguments are those other entities, that are placed in relations. These are often DPs, like John or the sandwich. Or, they can be propositions, like that John left or John to leave, generally combinations of a predicate and an argument.

θ-theory

- The number (and type) of θ-roles assigned by the predicates are recorded in the lexicon.
- “Weather verbs”: assign no θ-roles, there are no participants (e.g., rain, snow).
- Transitive verbs: assign two θ-roles, often Agent and Theme. These are assumed to assign the external θ-roles through a v component. (e.g., kick, like, see, eat)
- Intransitive verbs: assign one θ-role, can be external (often Agent or Experiencer) (unergative verbs, e.g., run, laugh, dance) or Theme (unaccusative verbs, e.g., melt, sink, trip, fall).
- “Ditransitive verbs”: assign three θ-roles, often Agent, Theme, Goal. These necessarily arise from a combination of v and V. (e.g., put, introduce, give)

Ditransitive verbs

- In order to assign three θ-roles (for ditransitive verbs like introduce), we need two XPs, which we’ve drawn like this.
- The labor of assigning θ-roles is divided between v, the light verb that assigns the Agent or Experiencer θ-role, and V, the main verb that assigns the Theme and Goal θ-roles.
Unaccusatives and transitives

- In *The ice melted* there is no external θ-role. So there is no v.
- In *Bill melted the ice*, we add a causer, an Agent.
  - Bill caused [the ice to melt].
  - "Bill was the agent of an ice-melting."
- So, something like this, where the main verb moves up to the light verb (which we had evidence for in ditransitives).
- In general, Agent and Experiencer are always assigned by a v.

Bill tried PRO to leave

- There is a class of verbs (control verbs) that embed nonfinite clauses that seem to be "missing an argument": try, want, …
- Think about the θ-roles; leave has to assign a θ-role, to the leaver, and try has to assign two θ-roles, one to the proposition (IP) tried, and one to the trier (*Bill*).
- But we only see two of those arguments: the IP and *Bill*.
- The missing argument is PRO.

Subject and object control

- Subject control predicates
  - Bill, is reluctant [PRO to leave]
  - *be reluctant, want, try, ask (no object), …*
- Object control predicates
  - John, persuaded *Bill*, [PRO to leave]
  - *persuade, ask (with object), tell, convince, …*
- PROarb
  - [PROarb to leave now] would be a mistake.
  - [PROarb pontificating] irritates me.

Unergatives, transitives

- Unergatives
  - Just an external θ-role…
  - *Bill lied*.
- There’s an Agent, so there’s a v.
  - *Bill ate the sandwich* looks just like *Bill melted the ice*.
  - v assigns Agent to Bill, V (eat) assigns Theme to the sandwich.

Reluctance…

- *Mary is reluctant (PRO to leave).*
- PRO does not get Case.
  - *Mary is reluctant Bill to leave.*
- In fact, PRO cannot get Case.
  - *Mary is reluctant for to leave*
  - *Mary is reluctant for Bill to leave*
- PRO refers (like a pronoun or an anaphor) to Mary.

Structural uniformity

- The different elements of the structure are each responsible for a certain element of the meaning.
  - C is responsible for the clause type (or illocutionary force) of the clause. It marks clauses as declaratives or as questions (or as imperatives, or as exclamatives).
  - I is responsible for tense interpretation (and also subject agreement).
  - v is responsible for external θ-role assignment (Agent for sure, others like "Experiencer" perhaps [win], or even simply marking as a verb [seen]).
  - D is responsible for definiteness (at least) (a vs. the)
  - N, V, A, P are responsible for lexical content.
Structural uniformity

- As a consequence of structural uniformity:
  - All *wh*-questions have a [+WH, +Q] C.
  - Subject *wh*-questions: Who left?
  - Object *wh*-questions: What did Pat buy?
  - All finite embedded clauses have a CP.
    - I heard [CP that [IP Tracy left]].
    - I heard [CP Ø [IP Tracy left]].

Features

- Lexical items have three kinds of features.
  - **Primary features** (Head features): Interpretable.
    - Fundamental to the meaning, crucial to interpreting the meaning of the structure.
    - [+3sg] on pronouns or D, [+D] on determiners.
  - **Uninterpretable features** (Specifier features): Not part of the meaning, but nevertheless part of the lexical item. Must be eliminated (checked off) by the end of the derivation.
    - [+Nom] on determiners, [+Nom] on I, [3sg] on I, as well as all complement features, all specifier features.
  - **Uninterpretable** (Complement features): Uninterpretable features that must be checked against the features of the complement (at first projecting Merge).

Head features

- **Interpretable** features are fundamental to the meaning, crucial to interpreting the meaning of the structure.
  - [+3sg] on pronouns or D, [+D] on determiners.
- **Uninterpretable** features are not part of the meaning, but nevertheless part of the lexical item. Must be eliminated (checked off) by the end of the derivation.
  - [+Nom] on determiners, [+Nom] on I, [3sg] on I, as well as all complement features, all specifier features.

Complement features (subcategorization)

- Heads can impose requirements on the kind of phrase that they can be combined with (Categorial selection). For example, will requires a bare form of the verb (we'll encode the bare form with the feature [Inf]).
- Finite I (any I except the infinitive to) has a [+Nom] specifier feature (SpecIP is assigned nominative Case).
- Specifier features are always uninterpretable.

DPs

- Even when you can’t see D, we assume it is there. Only a DP (not an NP) can get a 0-role.
- A pronoun (he, she, I, him, …) is just a D, like the.
- Agreement features (e.g., [3sg], [+plural]) and Case features (e.g., [Nom]) are features of D, not N. They need to be able to check features of I.

Specifier features (Case and agreement)

- Finite I (any I except the infinitive to) has a [+Nom] specifier feature (SpecIP is assigned nominative Case).
- Specifier features are always uninterpretable.
**Merge**

- Building a tree up from the lexical items we have available (on the “workbench”) is accomplished by Merge of two objects together.
- When two objects are Merged, one projects. Generally, the one that projects is the one that had an uninterpretable complement-(or specifier-) feature to check.

**Adjoin**

- I generally indicate adjunction with a “double branch” to keep it clear what is adjoined and what is not.
- The concept here is that the VP node has been “stretched out” and the AP has been hooked into it.
- The AP occupies a strange position in the tree. It is not a sister, nor a daughter of VP. It is sort of in-between. It’s not fully dominated by VP, it’s only dominated by part of VP.

**Adverbs**

- Adverbs generally are adjoined to the vP.
  - He quickly ate it.
  - He ate it quickly.

**Adjoin**

- The operations Merge and Adjoin are two different ways to combine two objects from the workbench.
- Merge takes two objects and creates a new object (with the label inherited from one of them).
- Adjoin attaches one object to the top of another one.
- Generally Adjoin is not motivated by the need to check any features. Eat doesn’t need quickly. Quickly doesn’t need a verb even: I want you off the ship quickly.

**Adjunction**

- The main intuitive idea: adjuncts are “loosely connected” and general serve as modifiers.
  - Adjuncts are generally optional (no θ-roles in any θ-grids).
  - They seem to be able to attach either to the right or the left.
  - They seem to attach to maximal projections.
  - The thing that the modifier modifies is the head of the phrase it is adjoined to (important when deciding where John heard a dog bark in the kitchen).

**Adjectives**

- Similarly, adjectives seem to adjoin to the NP.
  - the tasty sandwich.
  - Pat’s tasty sandwich.
PPs serve the same function

- PPs often serve to modify the event like adverbs, and are adjoined in the same way (on the right).
- She ate it on the hill in the rain.

\[
\begin{align*}
\text{IP} & \\
\text{DP} & \text{She} \\
\text{I} & \text{[+Past]} \\
\text{VP} & \text{PP} \\
\text{V} & \text{on the hill} \\
\text{t_i} & \text{if} \\
\end{align*}
\]

- PPs can also modify nouns, like adjectives (again on the right).
- Pat bought the book with the shiny cover.

\[
\begin{align*}
\text{DP} & \\
\text{the} & \text{NP} \\
\text{PP} & \text{with the shiny cover} \\
\end{align*}
\]

N complements

- Not everything that shows up to the right of an N is an adjunct. Some are complements. Generally there can only be one complement, it doesn’t reorder with adjuncts, it defines a fundamental characteristic. Other examples (CP complements of N): the claim that John left, the rumor that John left. One replaces both the N and the complement (the one by Radford, “the one of poems, “the one that John left”)

\[
\text{NP} \quad \text{book} \\
\text{PP} \quad \text{of poems} \\
\text{by Radford} \\
\]

- They dominate H and it dominates YP.
- Does H c-commands YP?
  - Is YP contained in H? No.
  - Does every node that dominates H dominate YP?
  - X? X doesn’t dominate H.
  - Y? X dominates H and it dominates YP.
  - The rest? They dominate H and dominate YP.
  - So, H c-commands YP.

- Is H contained in H? No.
- Does every node that dominates H dominate X?
- X? X’ dominates H and it dominates X.
- The rest? They dominate H and dominate X.
- So, H c-commands X.

Adjunction, c-command

- The main thing this concept of a “stretched” out node affects is what c-commands what in this structure.
- Dominance: A node \( \alpha \) dominates a node \( \beta \) if \( \alpha \) is contained within all of \( \beta \).
  - Under this definition XP does not dominate UP, because part of XP does not contain UP.
- C-command: A node \( \alpha \) c-commands a node \( \beta \) if:
  - \( \beta \) is not contained in \( \alpha \), and
  - every node \( \gamma \) that dominates \( \alpha \) also dominates \( \beta \).
  - By \textit{contained in}, we mean either dominated by or “hanging off of”.

\[
\begin{align*}
\text{UP} & \text{XP} \\
\text{ZP} & \text{XP} \\
\text{H} & \text{YP} \\
\text{X} & \text{ZP} \quad \text{XP} \\
\text{Y} & \text{X} \\
\end{align*}
\]
Adjunction, c-command

- C-command: A node α c-commands a node β if:
  - β is not contained in α, and
  - every node γ that dominates α also dominates β.
- Does UP c-command ZP?
  - Is ZP contained in UP? No.
  - Does every node that dominates UP dominate ZP?
  - Yes, vacuously here, but yes for sure if XP is embedded in any further structure.
  - So, UP c-commands ZP.

Adjunction, c-command

- In practical terms, an adjoined element c-commands what it is adjoined to, and everything that element c-commanded before the adunction.
  - H c-commands X.
  - H c-commands WP.
- The element adjoined to does not c-command the adjoined element—they do not become sisters (which c-command each other).
  - XP doesn’t c-command UP.
  - X doesn’t c-command H.

Three kinds of movement

- Head Movement: Movement of a head to adjoin to the next higher head.
- A-movement: Movement to SpecIP (subjects, passive objects, subject raising), to satisfy the EPP.
  - A-movement ends in a Case location.
- Operator Movement: Movement to SpecCP and other things we’ll talk about later. A.k.a. “A-movement”
  - Operator movement starts in a Case location. So A-movement precedes operator movement.

Movement

- Movement is essentially just Merge/Adjoin but with only a single item from the workbench.
- We find something inside the object, make a Copy, and then Merge or Adjoin that Copy (in)to the object. The newly-added copy must c-command the original (movement is always upwards).
- When pronouncing a tree with two copies of something in it, we pronounce only one copy (the one that c-commands the others).

When V moves to I

- When V moves to I, it will appear before adverbs and negation (in a head-initial language like English or French).
  - Pat is quickly eating a sandwich.
  - Pat is not eating a sandwich.
  - Pat does not eat sandwiches.
- V head-joins (adjoins, head-to-head) to I, forming a complex head, (if it’s an I with a V adjoined to it).
  - English: Auxiliaries (have, be) move to I.
  - French: All verbs move to I.
**Head Movement Constraint**

- Heads can only move to heads.
- The HMC says that a head cannot move past another eligible head to reach its destination. (Economy)
- Specifiers don’t count as eligible (though they contain a head, to be sure).
- The bottom line is: Head movement adjoins a head X to the head of the phrase YP that has XP as its complement.

**Expletive there**

- *There* is another meaningless element without a θ-role (like it) that can satisfy the EPP.
- What differentiates *it* and *there* is the connection between *there* (the expletive) and another DP (the associate). The associate DP is enabled to check its features “by proxy” by its association with *there*.
- *Students* is a DP (has a Case feature needing to be checked). The Case feature (and the [-Plural] feature), can be checked with I across the expletive-associate feature "conduit."

**ECM**

- For example: Bill finds me to be intolerable.
- Bill is the Experiencer of *find*, hence has a DP to assign the Experiencer θ-role.

**The EPP (driving A-movement)**

- The EPP IP must have a specifier.
- More informally, all clauses have subjects.
- Because *rain* has no arguments (no θ-roles), a special, contentless pronoun (*it*) has to be inserted to in order to have a grammatical sentence. This kind of “empty it” is called an expletive or a pleonastic pronoun. It is not an argument (in this use).
- We stipulate that *it* is not subject to the θ-criterion.

**Government**

- Features can be checked in a local environment (the positions governed by a head).
- The specifier-features of X are checked against DP₁ in its specifier.
- The complement-features of X are checked against YP in its complement, or, failing that, against DP₂ in the specifier of YP.
- This DP₁ position is primarily relevant for checking accusative Case (ECM).

**Small clauses**

- For small clauses (including *I saw her in the garden*), the subject also gets Case via ECM.
- (Note: the meaning represented here is compatible with *me* not being in the garden)
**Movement for EPP/Case: Unaccusatives**

- **Finite I can check Case**
- **Unaccusative V cannot check Case**

**Passive**

- The passive is just like the active, but without the vP.
- The Theme moves into SpecIP, satisfying the EPP (and getting Case).
- Notice that the DP doesn’t get Case in its underlying position (it can’t get Case twice, and it gets Case in SpecIP; “it was eaten the sandwich”). Burzio’s Generalization: No external argument (no little v), no accusative Case.

**Subject raising**

- Subject raising occurs when
  - the subject of a lower clause does not get Case in the lower clause.
  - the main verb in the higher clause has no external θ-role.
- And in the last step, we Move the DP Mary up from the lower SpecIP to the higher SpecIP.

**Operator movement: wh-movement**

- English: One wh-phrase moves to the front.
- What did Bill give to whom?
- Japanese: No wh-words move to the front.
  - Taroo-ga dare-ni nani-o ageta no?
    - T-nom **who-to what-acc** gave Q
    - ‘What did Taroo give to whom?’
- Bulgarian: All wh-words move to the front.
  - Kakvo na kogo Ivan dade?
    - **what to whom** Ivan gave
    - ‘What did Ivan give to whom?’
Spellout

- We handle this kind of variation by supposing that:
  - Wh-words need to move to SpecCP
  - Languages differ in where in the derivation they choose to focus pronunciation (“Spellout”).

Languages differ in where in the derivation they choose to focus pronunciation (“Spellout”).

Fitting all of the *wh*-words in SpecCP

- To get all of the *wh*-words in SpecCP, all of the *wh*-words after the first one move to adjoin to the first one.
- This way, there is still one specifier of SpecCP, but the *wh*-words are still all in the specifier of SpecCP, attached to one another. (Note: right-adjoined, cf. Bulgarian)

Subjacency

- The way Subjacency violations are avoided is through the use of successive-cyclic movement: A moving *wh*-phrase will stop off in each SpecCP on the way from its original case position to its scope position.
- If a SpecCP is full along the way, the *wh*-phrase would have to skip past that SpecCP, which would entail a movement that is too long (*wh*-island violations).

Superiority

- Superiority: The shortest *wh*-movements have to happen first. (*Wh*-movement isn’t possible if there was a shorter one).
- *What* did Bill persuade *who* to buy *what*?

Subjacency

- Not only do movements of *wh*-words need to be as short as they can be (cf. Superiority), they also have an upper bound on how long they can be even if there isn’t a shorter competitor.
- Subjacency: A single movement cannot cross more than one bounding node.
- Bounding nodes (English): IP (if sister to C) and DP.
- Bounding nodes (Italian): CP and DP.

What will they bake?

- We start out with essentially the structure of *They will bake what* as shown here.
- *What* is a DP, but it’s a *wh*-DP, a [+WH] DP.
What will they bake?

For wh-questions, we have an additional item on our workbench, a [+Q,+WH] C.

Two features it needs to check: [+Q], checked by moving I to C. [+WH], checked by moving a [+WH] DP to SpecCP.

Successive cyclic wh-movement

When a wh-word moves, it has to move to the closest SpecCP. It can't skip a SpecCP (or it would have to cross two IPs).

What did you hear that they bought?
Successive cyclic *wh*-movement

- When a *wh*-word moves, it has to move to the closest SpecCP. It can’t skip a SpecCP (or it would have to cross two IPs).

What did you hear that they bought?

Successive-cyclic movement

- Then, the *wh*-phrase moves from the intermediate SpecCP to the main clause SpecCP.

What did you hear that they bought?
Successive-cyclic movement

- Then, the wh-phrase moves from the intermediate SpecCP to the main clause SpecCP.

What did you hear that they bought?

Wh-islands

- Now, suppose we have an embedded wh-question.
- You wonder what they bought.
- And try to question the subject.

Wh-islands

- Now, suppose we have an embedded wh-question.
- You wonder what they bought.
- And try to question the subject.

Wh-islands

- Too far—
- Wh-movement can’t go past the middle CP without “stopping off”
Complex NP islands

- IP and DP are both bounding nodes, so you can’t move a wh-word out of a DP.

Op

- In relative clauses, we sometimes find Op, the silent wh-word.
- That is, *the book which Mary read* and *the book Mary read* are really exactly the same except that in one case you pronounce the wh-word, and in the other, you don’t.
  - the book \([_{CP} \text{which} \_C \text{Mary read} \_t \_i]\)
  - the book \([_{CP} \text{Op} \_i \text{(that) Mary read} \_t \_i]\)

Relative clauses

- The structure of a relative clause is like this.
- A [+Q, +WH] CP is adjoined to the NP, like an adjective, or a PP modifier.
- The meaning is essentially “the man with the property of being the answer to ‘Who did I meet?’”

Op, DFC, & Recoverability

- The Doubly-Filled COMP filter is the traditional “explanation” for why *the book which that Mary read* is bad.
- Doubly-Filled COMP filter: *\([_{CP} \text{wh-word if/that/for…}]\)*
- Recoverability condition: The content of a null category must be recoverable.
  - the place \([_{Op} \_i \text{(that) Mary bought that book} \_t \_i]\)
  - the day \([_{Op} \_i \text{(that) Mary bought that book} \_t \_i]\)
  - the reason \([_{Op} \_i \text{(that) Mary bought that book} \_t \_i]\)
  - the way \([_{Op} \_i \text{(that) Mary bought that book} \_t \_i]\)
- This is why you can’t just ask a regular wh-question with Op.
Yes/no questions

- There is also reason to think that there is an Op occupying SpecCP in yes-no questions as well.
  - *Who did you wonder if John met?*

Quantifiers

- These phrases which don’t refer to specific people/things in the world but rather seem to do things to sets of people/things are quantifiers. Examples include:
  - most students
  - twelve angry men
  - fewer than half of the members
  - some custodian
  - nobody in their right mind

QR: Covert adjunction to IP

- QR adjoins the quantifier to IP.
- Moving a quantifier (QR) is required because the quantifier needs to get out of the IP (for interpretation). IP itself has no need for quantifiers.
- Moving to SpecIP or moving to SpecCP is motivated by some need of I (EPP: SpecIP must be filled) or C ([+WH] C needs a [+WH] in its specifier).

Subjacency for overt movement

- Who believed the rumor that John bought what?
- Who remembers where we bought what?
- These sentences would suggest that covert wh-movement is not sensitive to wh-islands. A very widely adopted assumption about Subjacency is made to explain this:
  - Subjacency only holds for overt movement.
  - Thus:
    - All overt wh-movement is successive-cyclic.
    - Covert wh-movement can move directly to SpecCP.

QR

- Sue read every book.
  - For every book \( x \), Sue read \( x \).
- After Spellout, the quantifier moves to a position above the sentence, so there is then a direct mapping between the structure and the logical form.
- \([\text{every book}]_i. \{\text{IP} \text{ Sue read } t_1\} \).  

QR: multiple quantifiers

- QR adjoins the quantifier to the IP.
- QR must happen for every quantifier.
- A quantifier is interpreted with its c-command domain in its scope.
- Detail: For multiple adjunction structures, we need to assume that \( \text{QP}_1 \) c-commands \( \text{QP}_2 \), but \( \text{QP}_2 \) does not c-command \( \text{QP}_1 \).
Weak Crossover

- Who, does his, roommate like t_i?
- Who, t_i, likes his, roommate?

- Weak Crossover (WCO): A coindexed pronoun cannot intervene between an operator and its variable.

- [Every girl], [p, her, roommate likes t_i].
- For every girl x, x’s roommate likes x.
- [Every girl], [p, t_i, likes her, roommate].
- For every girl x, x likes x’s roommate.

Binding Theory

- Principle A. An anaphor must be bound in its binding domain.
- Principle B. A pronoun must be free in its binding domain.
- Principle C. An r-expression must be free.

The binding domain for an anaphor is the smallest of (i) An IP that dominates it, (ii) A DP, with a specifier, that dominates it.

Bound: coindexed with a c-commanding antecedent (Free: not bound).

Weak Crossover

Pronouncing at Spellout

- Lexical items come with some information about how to pronounce them. That is, cat is pronounced [kæt].
- Some lexical items can be pronounced alone.
- Some lexical items are affixes that attach to other kinds of lexical items.
- English Tense/Agr (I), for example, is a suffix that is pronounced together with (usually at the end of) a verb.
- Occasionally PF will be faced with the task of pronouncing a suffix without a host nearby to attach it to.

PF: do-support

- When a verbal suffix is “stranded” like this, the only way to pronounce it is to pronounce a verb along with it.
- The “default” verb in English is do.
- So, “stranded tense” affixes get pronounced attached to do: do-support.
- Does John eat constantly?
- John does not eat constantly.
- Note: do is not in the tree. It is inserted as we try to pronounce the tree. It therefore also doesn’t (and couldn’t) have any effect on the meaning.

A few things to look for

- θ-criterion: Are all θ-roles assigned to exactly one argument?
- Is there an Experiencer or Agent? (Then there’s a vP)
- Do all DPs get their Case features checked?
- Is the EPP satisfied everywhere (all IPs have a specifier)?
- Have all of the quantifiers adjoined to IP by LF?
- Are all wh-words in SpecCP by LF?
- Did any wh-movement cross two or more bounding nodes (for movement before Spellout)? (Subjacency violation)
- Are all anaphors bound in their binding domain? (Principle A)
- Are all pronouns free in their binding domain? (Principle B)
- Are all r-expressions (completely) free? (Principle C)
- Did an Operator movement cross a coindexed pronoun? (WCO violation)
- Have the auxiliaries moved to I?
- Has I moved to C (in main clause questions)?
Some sentences from previous finals/practices

- 2002F:
  - What does every agent suspect Jack gave to Nina?
  - Jack successfully convinced Vaughn to fire Will.
  - Which memo is likely to have been dropped behind Leo’s desk?
- 2001PF:
  - Every father wants to know what the children are watching.
  - What had Bert’s mother said was stolen from the living room?
  - Ralph’s puppy seems to like to chew the sofa.
- 2001F:
  - What had Bill expected to buy at Wal-Mart?
  - Every serious linguist will eventually need to know what Chomsky has written.
  - My tape of Benton’s last episode appears to have been misplaced.

Some sentences from previous finals/practices

- 2000PF:
  - Who do you think bought the laptop which Mary said she sold?
  - Which student will Mary say took every prerequisite?
  - Mary said that John’s mother was chosen.
- 2000F:
  - Which test will Mary say that every student took?
  - Which senator said that Congress will pass which bill?
  - The pen which Larry’s assistant thought that Artie lost was found under the table.