

Lecture notes by Edward Loper

Course: Linguistics 551 (Syntax II)

Professor: David Embick

Institution: University of Pennsylvania

1 Logistics

discuss papers 3-4 assignments

For thursday: re-read 2nd double-object paper

2 Heidi's Paper & Double Object Constructions

Basic fact that prompted the discussions:

- hierarchical asymmetry between the arguments (which arg binds which, etc.)
- Emphasis on binary branching

```
|| give NP1 to NP2      NP1 higher than NP2
|| give NP2 NP1       NP2 higher than NP1
```

Larson's shell structure:

```
|| [VP [V' [V] [VP [DP a letter]
||                          [V' [V send] [PP to Mary]]]]]]
```

This accounts for "NP₁ higher than NP₂"

```
|| [VP [V' [V] [VP [DP Mary]
||                          [V' [V send] [DP a letter]]]]]]
```

- Derivation vs Alternate Projection
 - If alternate projection, what's the content of the lower structure?

2.1 Arguments against derivation approach

Larson: UTAH requires that both sentences have the same deep structure.

Heidi: they use different semantic roles

Idioms: can we apply transformations?

```
|| Mary threw John to the wolves
|| * Mary threw the wolves John
```

Some idioms are preserved under "transformations" (e.g., passivization).

But Larson is using something like passivization to relate double-object forms. Since idioms are not preserved, maybe we want to say that they are not derivationally related. (n.b.: not an airtight argument; some idioms do not survive passivization, etc.)

2.2 Why is Phave a preposition?

- In particular, why not a verb (like in Larson)
- cross-linguistic arguments (languages with be+Phave)
- what does it mean to be a P (distributional definitions?)

2.3 Pesetsky's account

|| [VP [V' [V give] [PP [DP a letter] [P' to [DP mary]]]]]

|| [VP [V' [V give] [PP [DP Mary] [P' G [DP a letter]]]]]

In this argument, give originates in the higher part of the structure. This may violate some locality conditions on idioms? (where do we bound the domain for idiomatic interpretation)

2.4 Idioms

- many idioms of the fact ____ kicked the bucket.. basically

|| [S [DP ____] [VP (idiom)]]

- not many idioms of the form:

|| [S [DP ...] [VP (idiom ____)]]

Keeps idiomatic information within a clause..

|| Mary gave a kick to John

|| Mary gave John a kick

2.5 Locality

For idioms, we want to restrict the domain to a single VP, where possible. Consider expressions like:

|| I sent John to the showers

Really, we would like "sent to the showers" to be a single constituent. That's one advantage of Larson's constructions: in its lower position, we have an idiom..

But using Pesetsky's analysis, we don't get this: "send" takes a PP containing both "John" and "to the showers".

So.. do what Heidi did: define the idiom in terms of the lower PP, and then have it raise up..

But an interesting question:

- Larson allows v+goal idioms, not v+theme idioms
- Heidi seems to allow both v+goal and v+theme

Ok, what about:

|| John gave his all to Linguistics.

But:

|| Linguistics gets my all.

(here, we're using a different verb)

But: is there a systematic relationship between give/get?

enchoative \approx become (e.g., the soup thickened)

We can use this to explain why some idioms exist in one double-object form and not the other:

|| A gave B a kick in the pants

|| * A gave a kick in the pants to B

Heidi:

- But then why do we get both:

|| John gave his all to Linguistics

|| John gave linguistics his all

- there's something funny about "John gave his all to Linguistics". In particular, Heidi claims that you can't have:

|| John gave his all to it

- Claim: both forms are using Phave.. but Heidi doesn't quite explain how that works.

2.6 Get and Give

|| The book got to Sue
 || Sue got the book
 || The book got to France
 || * France got the book

Argument: get & give are related in a specific way: they share structure..

Two types of – verbs:

- unergative (e.g., run): subj comes from spec/vp
- unaccusative (e.g., break): subj comes from arg of vp (no external argument)

Argument:

- "get" is unergative BECOME+Phave/Ploc

|| [VP [V BECOME [PP [DP the book]
 || [P' [P Ploc] [PP to Sue]]]]]
 || [VP [V BECOME [PP [DP Sue]
 || [P' [P Phave] [DP the book]]]]]

Where Phave/Ploc raises to BECOME, and the spec/Pp raises to subject.

This allows us to explain common behavior btwn get and give, e.g., in contexts like "his all."

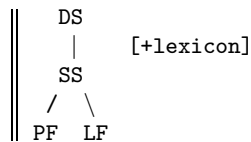
2.7 Get and Give (continued)

What parallels can we make between get and give?

- get and give are not primitive
- get & give relate to features & structure
- get, have, and give are compositional? (subcomponent = Phave)
 - allows us to explain parallellisms between verbs
 - describes double-object verbs correctly

2.8 Model of the Grammar

Old model:



where lexicon provides us with atomic lexical items. Each lexical item contains info on DS, SS, PF, LF..

- Lexicon has some internal compositional rules (e.g., -ed); but these are modularized inside the lexicon

But this model doesn't work with Harley's view.. Basic premise:

- no lexicon

What do we mean by that? There are a set of features, that the syntax works with, and those can be mapped to words..

- accounts for idiosyncratic words better

So really what we have here is a more distributed lexicon..

- mapping btwn features ↔ words (phonological rules?)
- mapping btwn features ↔ semantics?

Idea: universal feature inventory? Ick. I don't like that so much..

- some languages have Phave, others don't.

Analogous to phonology: IAP... But are the phonetic symbols really universal? ASL, etc.. Alternative modes of expression..?

Also, these Phave things have semantics.. at least phonetic symbols are just arbitrary symbols.

Lexical vs functional split in grammar?

Features as compositional parts of speech?

Coersion: you can force an interpretation of a word by putting it in a specific grammatical position.

Features affect the compositionality of our grammar: how can we put things together; there's no reason to expect that these compositionalitys can vary in arbitrary ways.. Posit a set of features, which are universal.. New words just use these..

lexical aspect vs grammatical aspect

What semantic content is carried by features? What isn't?

- e.g., cause is, dog isn't.

Freze & Kayne:

- be+Phave = have
- be+Phave = be+P

Harley:

- (no Phave in the language)

If you can show that goal>theme in double object form of "have"; then you can show that a language has Phave.

Bad glosses of Hindi:

```
|| I india-in be      ... locative
|| Room-in man be    ... existential
|| Boy-OBL near dog be ... possessive
```

(Possessive is an existential with an animate in it?)

Freeze:

```
|| [ IP [] [I' [I be] [PP [theme] [P' [P p] [DP location]]]]]
```

- For locative: theme raises
- For others: location raises

Put new info at the beginning of the sentence (thing at the beginning becomes definite)

Indefinite must stay within the VP

But Harley's counter-example:

- "[that boy has my dog]" → "my dog" is definite.

But here, "have" might have a different meaning. So this counter-example may not be air-tight.

Harley's other counter-example:

- some languages (e.g., Scott's Gaelic) treat them all the same..

```
|| [PP [DP I] [P' [PP india-in] [P Ploc]]]
|| [PP [DP Room] [P' [DP man] [P Phave]]]
```

Harley: 3 options for phave:

1. Phave, have
2. Phave, no have
3. No Phave

Irish: no Phave.

```
|| Be the oatmeal in the pot ... locative
|| Be oatmeal in the pot ... existential
|| Be the pen at Mary ... possessive
```

Every boy has his pen:

```
|| Is his pen self at every boy
```

(express ownership with Ploc)

```
|| * Gave Milo his pen to every boy.
```

(since this would require Phave)

3 The minimalist program

3.1 Roadmap

- Conceptual
 - What's necessary
 - What we want to do
 - Arguments & illustrations
 - Technology
 - Derivational system
 - * Derivational locality
 - * Feature system
 - * Language variance

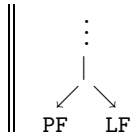
Next paper: tag (includes good summary of derivational system etc.)

3.2 Virtually Conceptually Necessary

What properties would a system have to have to produce a language like the one we hypothesise?

- lexicon
- generative syntactic system
- articulatory/perceptual interface
- conceptual/intentional interface

3.3 Language Faculty



|| $S_0(\text{PLD}) \rightarrow \text{language}$

PLD = primary linguistic data

Language variation:

- limited to what's overtly visible to the learner
 - lexicon (relationship between sign & object)
 - morphology/inflection
 - things like headedness

Strong features vs weak features:

- strong features force movement before the LF/PF split, so we do hear it in surface form.
- weak features force movement after LF/PF split (only on the LF part of the derivation), so we don't hear it in surface form.

Problem:

- no evidence for what happens after PF/LF split

→ we say that it's universal (you don't have to learn it)

Problem:

- what about stuff like optional movement, etc? and if LF is universal, is English a V2 language, etc?

→ say that some of that happens between split & LF?

Non-derivational conditions must be stateable at the interfaces? (E.g., constraints other than derivational locality, conditions at movement). Thus, e.g., case & binding etc. must be conditions at some interface.

3.4 Getting rid of D-structure

What was supposed to hold at d-structure?

- θ criterion: each arg gets exactly 1 θ role
- projection principle: reps at every syntactic level meet lexical requirements.

If we get rid of these, then where do we state them?

- Redundancy with things we want to say at LF anyway?

Tough-movement

Sentences like:

|| John is tough to move

|| John is easy to please

|| John is easy [CP [DP_i WH] [PRO to please t_i]]

→ PRO = arbitrary PRO

|| It is easy to please John.

Movement? I.e., with d-structure, could they both be derived from:?

|| Is easy PRO to please John

No: because movement would violate locality constraints?

Parasitic gaps

|| * John read the book [without understanding]

|| Which book did John read without understanding

So we end up saying that "John" needs to be inserted late; but "John" can be replaced by things that are arbitrarily complex

→ Instead of saying there's a d-structure, focus on the how things are put together.

"Spell-out" = where DF/LF splits

3.5 Getting rid of S-Structure

We want to say that we can't do any constraints at SS.

2 types of evidence:

- difference in spell-out
 - reduce to PF or LF differences (overt differences in inflectional morphology: mediated by the feature system).
 - * strong features vs weak features.
 - * many conditions seem to occur at s-structure? or at least have been assumed to occur there.
 - show that it can/must/can only apply at LF

Example

Here's an example of the type of argument you can give for saying that we don't need SS. Start with something that seems to require SS, and then show that you could get away with just doing it at LF..

he can't be coindexed with John in:

|| You said he liked [the pictures that John took]

but they can be coreferent in:

|| [How many pictures that John_i took] did you say that he_i liked t?

|| Who [t said he liked [α how many of the pictures that J took]]?

Supposed to mean "how many pictures did each person say they liked?"

In theory, at LF, α raises to who.. which should mean that the condition C holds at SS. But we could instead say that just the "how many" raises.

3.6 Derivational System

- We would like to get rid of the "satisfy" operation.
- How do we put things together?
- Locality

X-bar: we can do one of three things with a lexical entry:

|| [x X]
|| [x' [x X]]
|| [xp [x' [x X]]]

Get rid of x-bar, and say that everything has its position by virtue of how it entered the derivation? notions like minimal and maximal are just derivative. we no longer differentiate specifier from adjuncts.

How do we put them together? Generalized Transformation (GT). Basically a substitution operation, with 3 components:

- identify a target K
- add empty element \emptyset .
- L substitutes for \emptyset
- output K'

In general GT is binary. Movement is sort-of a unary GT.

- create trace/chain
- move element

Conditions for GT...

- shortest move: movement goes to the closest position of the right type.
- extend target (extension requirement): you can only add elements to the top of a tree. Enforces some locality, e.g., cyclicity.

4 Conditions on well-formedness

Different elementary objects/rules of compositions lead to different formulations of grammatical rules.

- GB era model
 - elementary object: single phrase of potentially unbounded size.
 - composition: satisfy; then move- α ; and filter.
 - conditions are stated by restricting the single phrase.
 - Chomsky 1993
 - elementary objects: projections of a head
 - composition: GT
 - * binary substitution
 - * unary substitution (=move- α)
 - * adjunction
 - * conditions can be stated as restrictions on derivation; and as restrictions on interfaces (=outputs)
 - * shortest move
 - * extend target
 - * (L)TAG
 - * elementary objects: elementary trees
 - * large enough that we can state grammatical conditions on them. ("privileged objects" = the objects that we state grammatical conditions on.)
 - * composition: substitution and adjunction
 - * conditions follow directly from rules of composition

4.1 GT: Locality in A-Movement

3 components to GT:

- binary substitution
- unary substitution (=move- α)
- adjunction

2 conditions:

- extension requirement (extend target)
- shortest move: landing site of a movement is the closest position of the right type.

We want to avoid "superraising":

|| John seems it is certain to like Pizza.

Extension requirement applies only to overt movement: we need to allow covert movement that violates the extension requirement, in order to allow the object to move to agr-o, etc.

5 Strange things with Chomsky '93

- ER is stipulated to apply before spell-out only
- Adjunction is different from substitution
- Rules out long movement; but there is good long movement:

|| Which_i car did John wonder how to fix t_i?

- How to explain head-movement? In particular, it seems to violate extend-target.

6 TAG

6.1 Basics

Objects

- elementary trees
 - initial trees
 - auxilliary trees

Operations

- Substitution
- Adjoining

Derived structure does not uniquely identify a single derivational structure.

Structure preservation: we're not adding structure as we go; we start out with the structure, and we maintain the relationships.

Objects

- small enough to prevent unbounded relations
- big enough to state all the grammatical conditions we want
 - θ criterion
 - projection principle

6.2 CETM

Elementary trees are a single lexical head plus the extended projection of that head.

- How do we define extended projection? In particular, what about VPs dominated by NPs, etc?

6.3 Fun with Adjunction

|| John seems to like pizza

Start with:

|| [IP [DP John] [I' [I to] [VP like Pizza]]]

Adjoin in:

|| [I' [I (subst)] [VP [V seems] [I' (foot)]]]

What trees should "seems" go in?

|| It seems that John likes pizza.

7 More tag

primitives = privileged objects (objects over which we state constraints).

minimalism: privileged objects are larger than primitives (phases?)

What is an elementary tree?

- CETM: single lexical head plus extended projection
- θ -criterion: license arguments
- projection principle: lexical selectional properties; we need traces.

7.1 Raising/Superraising

|| John seems to like pizza

Start with:

|| [IP [DP John] [I' [I to] [VP like Pizza]]]

Adjoin in:

|| [I' [I (subst)] [VP [V seems] [I' (foot)]]]

This tree is missing spec/IP. Why? Because "seems" doesn't select for a subject.

Scope with raising verbs.. The following has a scope ambiguous: one particular linguist, or any linguist?

|| A linguist is certain to solve this problem.

We can explain this (kinda) with the vp-internal subject hypothesis: "a linguist" moves within the VP; if we interpret the dest vs the trace (after adjunction), we get different scopal readings.

Superraising

|| * John seems it is certain to like pizza

Start with:

|| [IP [DP John] [I' [I to] [VP like Pizza]]]

One option: adjoin in an I' auxiliary tree "seems it is certain." Why can't we do this? Violates CETM; problem with licensing of "it"?

Another option: adjoin in "is certain"; then we would need to adjoin in "seems it." This seems very odd.

derive "it seems that john likes pizza". Start with:

|| [IP [DP John] [I' [VP likes Pizza]]]

Then adjoin/substitute:

|| [CP ... [V' [V seems] [CP (foot/subst)]]]

empirical observation: when you get an expletive subject, you get a cp complement. we could explain this by saying seems needs to be expletive, so the version of "seems" that takes a cp compl needs to extend upwards to cp, so it needs a spec/ip position.

Can we produce "it is certain to like pizza"?

Consider combining:

|| [IP John to like Pizza]
|| [I' seems I']
|| [I' to be certain I']

What happens when we re-order the adjunctions?

8 Movement in TAG

Yesterday: Locality in A-movement

- Adjoining
- Conditions on elementary objects

Today: conditions on other dependencies

- A' movement: relation between non-arg and arg position

We can state some movements within the elementary object. E.g., relationship between a wh-element and a trace.

"Form chain":

- co-indexes an element in spec with a c-commanded θ -position
- can only occur in elementary trees

Apparent successive cyclicity: just adjunction?

|| What do you think that John_i ate t_i

"do you think" is an aux tree at C'. These are "bridge verbs."

conflict in chomsky's notion of economy:

- do we want shortest move or fewer operations?

in tag, there's no conflict: we're just stretching dependencies.

8.1 Long Movement

|| [Which car]_i did Sally wonder how to fix t_i?
|| * How_j did Sally wonder [which car]_i to fix t_i?

Addressing this with Chomsky 1993

Problem with 2nd sentence, according to Chomsky 1993:

- if we move "how" before "which move": violates extension requirement.
- if we move "which move" before "how": violates shortest-move (we're moving "how" over "which move")

But: these conditions also rule out the 1st sentence!

Addressing this with tag

Two options:

- don't posit a co-indexed trace. But the argument needs to be interpreted in a particular position. But maybe say that the adjunct just doesn't need a co-indexed trace.
- multi-component sets: relax the assumption that wh-trace relationship is contained in a single elementary tree.

Multi-component sets: change our notion of elementary sets to multicomponent-sets.

Condition on multi-component set: they must both adjoin simultaneously into the same underived elementary tree. Degenerate element must be c-commanded by the bigger element.

- use multi-component sets for all movement? (i.e., no form-chain?)

How do we explain the (1/2) sentence distinction? Assume that adjuncts and arguments are fundamentally different. :-/

8.2 TAG ECP

If x is empty it must be either:

- properly governed
- head of an athematic auxilliary trace (adjuncts)

|| What_i did John say (t_i) that Bill ate t_i?

That-trace effect: subject-object asymmetries

|| What_i did John say that Bill ate t_i?

|| * Who_i do you think that t_i ate apples?

[CP how to fix [DP *]]

9 Chomsky: Derivation by Phase

Continuity with:

- minimalist program
- desire for smaller domains to state syntactic conditions

To what extent is the human language faculty (FC) an "optimal" solution to certain "design specifications"? What do we need at the bare minimum?

- ways to put things together
- PF and LF interfaces

Minimalist program is a "program" because it tells you what standards to meet, not what exactly to do.

9.1 SMT = Strongest Minimalist Thesis

FC is an "optimal" solution to "design specs." To make this an empirical hypothesis, we need to define some things, eg:

- optimal
- interface conditions

We can appeal to SMT as a "standard:" an analysis should conform to SMT. To the extent that an analysis doesn't conform to SMT, it is "descriptive:" it tells us what happens, but not necessarily how..

9.2 Inclusiveness condition

- Work with what you start with (don't add new elements in the course of the derivation). Excludes some things: indices, traces, syntactic categories, bar levels. "category" → lexical/open class categories.
- Applies to lexicon → LF derivation, not necessarily to spellout → PF.

get rid of xbar:

- we don't need it – define positions in terms of their structural position.
- no diff between spec and adjunct?
- Multiple specifiers?

Movement (aka "displacement") is an "imperfection:"

- why would we want it?
- reduce it to independent properties.
- external motivation of an independent nature: scoping, discourse motivations, etc?

Features:

- [PAST], [PLURAL], etc.: justified – they are interpretable → they have some concrete meaning and help us in some direct way
- uninterpretable features: trigger movement; and license relationships (case and agreement).

Functions of agr get re-assigned (they're not justified – uninterpretable features). Reassign AgrSP → TP and AgrOP → vP (little v).

Motivations for little v:

- external arguments are different from internal arguments
- verbalization (esp light verbs).. "blacken" and "fooize" etc.

little-v is what takes "DESTROY" and makes it into a verb..

Distributed morphology:

|| [little-n [DESTROY the city]] ⇒ "destruction of the city"
 || [little-v [DESTROY the city]] ⇒ "destroy the city"

Causitive thing in double-object would be a little-v.

Difference between:

|| the window broke
 || John broke the window

might be that the second one is using a causitive little-v..

ϕ -features: person/number/gender (nominal agreement)

- Tense has uninterpretable, unspecified ϕ -features
- subject has interpretable, specified ϕ -features
- unify to get rid of uninterpretable features (AGREE)

All uninterpretable features must get eliminated.

9.3 Phases

- VPs (transitives)
- CPs
- levels that limit the amount of stuff we need to look at (c.f. tag's elementary trees)
- point in derivation where the object is "complete" in some sense..

10 Argument Structure

| Hale & Keyser

- arguments of verbs
- thematic roles

How to relate thematic roles to verb arguments? What structure is there to the relationships?

Consider "psych-verbs":

- Subject experiencer:

|| John fears apples.

- Object experiencer:

|| Apples frighten John

HK's thesis: the basics of argument structure are syntactic.

- L-syntax: something outside of syntax that looks like syntax "lexical syntax".. But why would we want 2 systems with very similar rules? Maybe part of syntax proper..
- Show that syntactic constraints determine possible argument structures.

10.1 Denominal Verbs

Denominal verbs (e.g., saddle, shelve, calve)

- transparently derived from nouns

Unergative

- no internal argument (John ate.)

Unaccusative

- no external argument (The door opened.)

Denominal Unergatives

Cross-linguistic support for unergatives-as-transitives.

Light verbs:

- do a jig
- make trouble

The complement to the verb gives you the predicate for the external argument.

Or: N incorporates into the verb (head movement), which gives denominal verbs. So our structure is:

|| [V' [V \emptyset laugh_i] [NP [N τ_i]]]

So "laughed" is really a light verb thing like "do a laugh."

Cross-linguistic support:

1. Light-verb verbs
2. N+V verbs

Denominal Transitives

|| John shelved the books.

Maybe it is derived from a double-object-y thing like:

|| John put the books on the shelf

|| [V' [V] [VP [DP the books]
[V' [V put] [PP on the shelf]]]]

|| [V' [V CAUSE shelf_i]

|| [VP [DP the books]
[V' [V \emptyset t_i] [PP [P' [P \emptyset t_i] [DP .. t_i]]]]]]

|| A cow calved.

|| * It cowed a calf.

Why is the second one bad? Because the external argument would need to move down to the verb. Head movement down is weird and bad. :)

Cognate objects:

|| He ran a good run

|| He laughed a hearty laugh

|| He smiled a happy smile

Also think about:

|| John laughed himself silly

|| Mary ran the shoes thin

|| Bill hammered the metal flat

11 More Hale & Keyser

11.1 Thematic Hierarchy

The semantic roles seem to be asymmetrically related to each other. Example:

|| [agent [exp [goal/source/... [theme]]]]

H&K claim: argument structure is syntactic.

- Is the hierarchy a principle of the grammar or is it derivative?
- Why are there so few thematic roles?

H&K response:

- There's really no such thing as a theta-role
- Instead, there are different structural configurations of arguments
 - Thus, there are a limited number of relations (since the trees can only have a limited # of configurations)

What determines the size of a basic configuration? Predication?

Relationships between l-structure elements. 2 Events:

|| [V' [V e₁] [VP [XP] [V' [V e₂] [YP]]]]

- The v₁-v₂ relation is causation?
- Prepositional (e.g., "on")
 - Relationship between wo things

Lower vp for "put" ..

|| [VP [DP the books] [V' [V put] [PP on the table]]]

- "the books" is the subject of the change-of-state

Deadjectival:

|| The cook thinned the gravy.

Lower vp..

|| [VP [DP the gravy] [V' [V BECOME] [AdjP thin]]]

Unergatives usually can't be causitized:

|| * The clown laughed the child.

Why is this excluded? Start off with the unergative..

|| [VP [DP the child] [V' [V] [NP laugh]]]

Why can't we then add a causitive V shell?

(possible counterexample: "I walked the dog")

How do we exclude these? Change rep of unergative, so that the lower VP doesn't have a subject:

|| [VP [DP John] [V' [V laughed_i] [VP [V' [V t_i] [N t_i]]]]]

(instead of:)

|| [VP [DP John] [V' [V laughed_i] [N t_i]]]

When are inner subjects required? Why are they not required for unergatives?

- Nominal isn't a predicate (at l-structure)
 - So it doesn't need a lower subject
 - PP is a predicate
 - So it needs a lower subject
 - Adjective is a predicate
 - So it needs a lower subject

Get

It seems that we can causitivize get only when its argument is a predicate (takes a lower subject):

- We got drunk.
 - We got my friend drunk.
 - We got into the peace corps.
 - We got my friend into the peace corps.
 - We got the measles.
 - We got my friends measles.

11.2 Alternations

Middle Alternation

```
|| John cut the bread.
|| * The bread cut.
|| The bread cut easily.
|| John shelved the books
|| * The books shelved
|| These books shelve easily
```

c.f.:

```
|| The gravy thinned.
|| This gravy thinned easily.
```

To explain, consider the difference:

```
|| The pigs got mud on the wall.
|| Mud got on the wall.
|| The pigs put mud on the wall.
|| * Mud put on the wall.
```

Get patterns with "splash, drip, dribble, etc." Put patterns with "smear, rub, wipe, etc."

- For put etc., manner component applies to the agent (it is the agent that smears, rubs, etc).
- For get etc., manner applies to theme (it is the substance that splashes, drips, dribbles, etc).

12 Hale & Keyser 98

Predicates and relations..

12.1 Unergatives

- analytic
 - e.g., have puppies
 - synthetic
 - e.g., laugh

Structure:

|| [VP [V have] [NP puppies]]

|| [VP [V DO] [NP laugh]]

putting things together to give verbs is similar to syntax. constraints:

- head-movement
- incorporation

call these constraints "conflation."

12.2 Bidirectional or Dyatic verbs

|| put the books on the shelf.

Relate two things (book and shelf).

The P is relating two things: put one in the specifier of the PP?

|| [VP [V put] [PP [DP the books]
[P' [P on] [DP the shelf]]]]

12.3 Verbal Adjectival

|| The leaves turn red.

Adjective requires a subject, but can't license it. Parasitic on the verbal component to license it.

Would we want "the leaves" to originate as a complement of red, and move out? Probably not..? Why?
There's not evidence for it.. But why would adjectives need to be parasitic on verbs, but for PPs it's not..?

in bulgarian:

|| book-the
big-the book
very big-the book
big-the red book

in standard swedish:

|| book-DEF
the big book-DEF
bigbook-DEF
bigblackbook-DEF

Explain the last 2 with head movement of adjectives..

|| the leaves turned red
|| the wind turned the leaves red

For the passive, "turn" is moving to a higher (new) V position. c.f.:

|| John broke the vase.
|| The vase broke.

Syntactically: when do we get transitivity alternation? Morphologically: what's the structure?

12.4 Cross-language studies

how universal are the head categories?

Notional categories: intuitive idea of what kind of thing a category expresses.

Morphosyntactic categories: structures defining a category..

"all adjectives are verbs" → "all adjectives/notional are verbs/morphosyntactic."

How much variation is there in the surface categories?

cf:

|| (1) * the clown laughed the child.
|| (2) The wind turned the leaves red.

For (2), we start with something that has an inner specifier. To transitivize, add a new V ([CAUSE]), move "turn" up.

For (1), we try adding a new V, but there's no place for "the child."

English	Mishitu	Navajo
boil/boil	pya-w/pya-k	béézh/l-béézh

In Navajo: associate higher V head with "I"

|| [V' [V I] [VP [DP ...] [V' [V] [AdjP ts'il]]]]

- Lexical causitive: e.g., break/break. there's a causitive item (john broke the vase) and a non-causitive one (the vase broke)?
- syntactic causitive: e.g., "john made bill eat the lima beans"... make+VP causitivizes..

Variation

1. morphology
2. categories
3. conceptualization of eventualities

various alternations patterns of derivational morphology and arguments that feed into the verbal system take a language, and experiment with how h&k's system works in it.

work with sergio...

go through verbs, categorize, etc.

go through several types of verbal structures.

morphoogical analysis

13 Argument Structure in Hungarian

- Free word order
- Sentence structure:
 - topic positions
 - wh positions
 - focus positions
 - vp

Almost anything can move into focus

|| [VP [DP the liquid] [V' [V freeze] [AP solid]]]

|| [F [AP_j solid] [... [V [V freeze] [VP
[DP the liquid] [V' [V t_i] [AP t_j]]]]]]]

Preverbs and adjectives have complementary distribution.

preverbs make the verb perfective (i.e., completed action)

c.f.:

|| John ate his way to fame.

|| * John ate up his way to fame.

|| John ate up.

The "up" and "his way to fame" are complementary – shared position?

So maybe preverbs and the adjectives share a common generation position?

13.1 Deadjectivals

Intransitive:

|| [V [DP the leaves] [V [V -od] [A piros]]]

Transitive:

|| [F [] [... [] [V [V] [V [DP the leaves] [V [V -it] [A piros]]]]]]]

13.2 Denominals

|| "to beer" → drink

|| "to tree" → plant trees

Equivalent of "the clown laughed the children" is ok:

|| the clown preV- laugh- CAUSE- past3s the child- plural- ACC

Hungarian equivalent of "I saddled the horse":

|| [V' [V] [VP [NP horse] [V' [V -e1] [PP [P up] [NP saddle]]]]]]]

But with this structure, "up"/"horse" end up in the wrong order; and how does "saddle" move past "up"?
Etc..

14 Verbal Alternations in Korean

- Agglutinative
- SOV unmarked word order

14.1 Zero-alternation

e.g., "car moved"/"I moved car". only 10-12 zero-alternation verbs in Korean. Example:

```
|| memchu = stop
|| car-NOM stop-Past-Decl
|| police-NOM car-ACC stop-Past-Decl
```

14.2 Adding Arguments

Intransitive → Transitive

Transitivizing morphemes: 'i-' 'hi-' 'li-' etc. Example:

```
|| cuk- = die
|| cuk-i- = kill
|| kkae- = awake
|| kkae-u- = wake up
|| john-nom die-past-dec
|| tom-nom john-acc kill-past-dec
```

Transitive → Causative

Same morphemes: 'i-' 'hi-' 'li-' etc. Example:

```
|| mek- = eat
|| mek-i- = have someone eat
|| ip- = put on
|| ip-hi- = have someone put on
|| emma-ka ai-eykey siliel-ul meki-ess-ta
|| mother-nom child-dat cereal-acc eat-cause-past-decl
|| "the mother fed the child with cereal"
```

key ha cuasation

Add "-key ha-". Very productive. Example:

```
|| emma-ka ai-eykey siliel-ul mek-ke-ha-ess-ta
|| mother-nom child-dat cereal-acc eat-cause-past-dec
|| "the mother made the child eat cereal"
```

14.3 Reducing Arguments

Transitive → passive

detransitivizing morphemes: 'i-', 'hi-', 'li-', 'ki-'. Example:

```
|| cap = catch
|| cap-hi = be caught
|| yel = open
|| yel-li = be opened
|| at last police-nom that thief-acc catch-past-dec
|| at last that thief-nom (police-dat) catch-passive-past-decl
```

14.4 deadjectivals

```
suffix: -eci-
|| i wultari-ka noph-ta
|| this fence-nom BeHigh
|| i wultari-ka noph-eci-ess-ta
|| this fence-nom BecomeHigher-past-decl
```

14.5 denominals

```
suffix: -ha-
very productive. e.g., with 'study'
```

15 Syntacs/Semantics Interface

Ideas originating in the 1980's.. LF (logical form). A syntactic level of rep where some semantics goes on. E.g., scope..

|| [DS [SS PF LF]]

There were claims that some things occurred at LF..

- thematic interpretation (θ roles)
- quantifier raising
- verb movement (minimalism)
- case (minimalism)

The first 2 have to do with interpretation; the second 2 with uniformity/invariance of language..

Why should all of these things be located in the same part of the grammar? (The reason we actually put them there – because they have effects that we don't see in PF; is that a good reason?)

15.1 May (1985): Logical Form

What belongs in LF? May: Whatever aspects of syntax are relevant to interpretation.

iSyntax constrains possible interpretations. (c.f. arguments in arg structure for why we can't get "the cow calved a child" or whatever).

How much semantic structure is manifest in the syntax?

- What are the representations?
- How are they derived?
- What constraints are there?

Framework:

- LF is transformationally derived from SS
- move- α , ECP & constraints on output

15.2 Quantifiers

|| [IP [DP John] [VP saw everyone]]

Use movement to bring "everyone" higher, so it has correct scope.

|| [XP everyone_i [IP [DP John] [VP saw t_i]]]

Weak Crossover

|| * [CP Who_i [IP does [VP [DP his_i mother] [V' love t_i]]]]

(Where the intended reading is "who is loved by his mother"?)

|| [CP Who_i [IP is [VP loved t_i] [PP by his_i mother]]]

Explanation: trace movement can't cross over coindexed pronominal element.

|| 1) His_i mother saw John_i
|| 2) *His_i mother saw everyone_i

The explanation is that in (2), "everyone" actually raises at LF, so we get the same weak-crossover violation (trace movement can't cross over coindexed prenominal).

ACD: Antecedent contained deletion

|| Dulles suspected everyone who Angleton did.

Assume that "did" is a copied VP.

|| [IP [DP Dulles]
|| [VP [V' [V suspected] [DP everyone [CP who Angleton did]]]]]

So we're substituting the VP into the place where "did" is..

|| [IP [DP Dulles]
|| [VP [V suspected] [DP everyone [CP who Angleton
|| [VP [V suspected] [DP everyone [CP who Angleton ...]]]]]]]

Infinite regress.

Reason for the name: the ellipsis antecedent itself contains the deletion (=ellipsis site). This is a problem, because when we plug the antecedent in, we get a new ellipsis site. So we get infinite regress.

c.f.:

|| [ConjP [IP Dulles [VP suspected Phillby]]
|| and [IP Angleton did too.]]

Where plugging the ellipsis antecedent for "did" in is fine..

|| * Dulles suspected Phillby, who Angleton did.

Again we have antecedent contained deletion. But now the antecedent doesn't have a quantifier. The quantifier can undergo QR, but the name can't.

|| [. [QP_i everyone [CP who Angleton did]] [IP [DP Dulles] [VP [V' [V
|| suspected] [DP t_i]]]]]

So now when we copy "Dulles suspected t_i" for "did," we don't get infinite regress:

|| [. [QP_i everyone [CP who Angleton
|| [VP [V' [V suspected] [DP t_i]]]]] [IP [DP Dulles]
|| [VP [V' [V suspected] [DP t_i]]]]]

Spys and Russians

|| Every spy suspects some russians

∃ y∀ x suspects(x,y)

|| [XP [QP_i some russian] [XP [QP_j everyone]
|| [IP [DP t_i] [VP suspects [DP t_j]]]]]

∀ x∃ y suspects(x,y)

|| [XP [QP_j everyone] [XP [QP_i some russian]
|| [IP [DP t_i] [VP suspects [DP t_j]]]]]

Where scope ≈ c-command.

16 ACD and LF

LF: The syntactic representation that's relevant for interpretation.

16.1 Quantifier raising

Are aspects of meaning affected by movement constraints?

- weak crossover
- scope eliminations
- acd (antecedent conditioned deletion)

how does it interact with other aspects of the grammar? (Binding, etc.)

Our QR account relies on ordering:

1. do quantifier raising
2. do copy

How do we do indexation? E.g., when we move/copy for acd, what are the various indices etc? In particular, is the "who" generated next to everyone? Or does it move up from somewhere? If it moves up, it should be moving up from within "did"..?? And how does it get coindexed with the everyone DP.

Binding theory

Condition c: r-expressions must be free

Fiengo & May (1994): Effects of QR are invisible to Binding Theory. So we would say that binding theory applies before QR.

|| * [IP You [VP bought him_i [DP every picture that John_i liked]]]

QR gives us:

|| [. [DP₂ Every picture that John_i liked]
|| [IP you [VP bought him_i t₂]]]

Which is ok with binding theory. But the sentence is bad. So it seems that the effects of QR are invisible to binding theory. (If binding theory applied after QR, then we would expect this sentence to be good). Old theory account: binding at SS, and QR at LF.

|| * You bought him_i every picture that John_i liked.

|| You bought him_i every picture that John_i thought you would.

|| You sent him_i the letter that John_i expected you would.

|| * You sent him_i the letter that John_i expected you would write.

So it seems that sentences that contain ACD get interaction with the binding theory.

Interim generalization: binding theory is affected by QR in ACD environments.

But: sometimes in ACD environments, there's no QR/binding interaction..

|| * He_i bought you every picture that John_i wanted to.

c.f.:

|| You [VP bought him_i every picture that John_i thought you would]

|| * He_i [VP bought you every picture that you thought John_i would]

Modified interim generalization: QR is affected by binding when both (1) ACD applies and (2) binder of the r-expression has a copy within the elided VP.

Binding theory applies to indices: \rightarrow it applies to i only when all instances of i are present.

"globality"

Assume an LF copying operation.

1) Binding theory fails to apply when an index is missing 2) So QR can affect BT if QR is required for the entry of an index 3) Then binding theory will wait for QR 4) ACD involves LF copying 5) If you're copying an indexed elt, you'll get a new index

|| I expected him_i to buy anything that John_i thought I did.

What chunk will the elided "did" be resolved to? a. <expected him_i to buy t> b. <bought>

Claim: a is bad, b is good.

- a: I expected him_i [everything that J_i thought I did] to buy t_i → Bad because J_i (r-expr) is bound
- b: [everything that J_i thought I did] I expected him_i to buy t_i → Ok because J_i (r-expr) is free

16.2 Economy

- If two applications of QR yield the same interpretation, the longer one is blocked.

I watched him buy everything that John thought I did

- <watched him buy>
- <bought>

16.3 Chomsky 93

|| a. *Which argument that John_i is a genius did he_i believe
b. Which argument that John_i made did he_i believe

- movement: A' always leaves a copy
- copies present at LF
- binding theory applies at LF
- adjuncts can be added late in the derivation
- interpret the restrictions at the tail of the chain, unless this is overridden.

Restrictor in {brackets}:

|| Dulles suspected [everyone {Angleton did}]
→ Dulles [everyone {Angleton did}] suspected t
→ Dulles [everyone] [suspected [t {Angleton did}]]

17 Schedule

- thurs & next tues: LF stuff (survey)
- next thurs: guest lecture
- then: get final, discuss, do other stuff

18 Moral

- We can rule out interpretations using syntax

19 Motivation

- Sentences with quantifiers and wh- words have "special" semantic properties.
- These "special" properties reflect syntactic generalizations.
- The machinery needed to derive LF is needed anyway (for overt movement).

- contrast
- condition

20 Issues in Quantifier Scope

| Guest speaker: Kimiko Nakanishi

Why use movement to capture scope? Because of parallels with wh-movement. E.g., island constraints preventing some readings, etc.

But, indefinites act somewhat different: they can take wide scope in places where other quantifiers can't. E.g., "a woman" can take high scope but "each woman" can't in:

|| if a/each woman comes to the party, John will be happy.

So indefinites can sometimes take scope over islands. But maybe these indefinites are just referential elements, so they don't have scope. (Use feature +specific/-specific, where +specific = referential).

At first, it seems that [+specific] indefinites must take the widest scope (they can't be between QR'ed elements. But not always. Explanations?

1. say that indefinites have island-insensitive QR
2. use choice functions

21 Remarks on Nominalization (Chomsky)

This paper introduces

- Lexicalist hypothesis
- binary branching
- X' theory

Basic question: what is the division of labor between various parts of the grammar?

- lexical vs syntactic

The grammar

1. The base
 - lexicon (words, features, etc)
 - Categorical component (CFG rewrite rules)
 - Transformations

Building a sentence:

1. expand CFG rules to terminal categories (=lexical categories)
2. replace terminal categories with lexical items
3. apply transformations

|| John is eager to please

- Gerundive: John's being eager to please
 - productive
 - transparent semantics/interpretation
 - case: internal structure typical of VP
- Derived: John's eagerness to please
 - not always productive
 - semantics not always transparent
 - no case: internal structure typical of NP

So gerundive seems transformational (from VP). What about derived? lexical or transformational?

- Lexical: extend the base
- Transformational: extend the syntactic apparatus (Lees 1960 gave a transformational approach; at that time, lexical approach wasn't an option.)

|| John is easy to please

- Gerundive: John's being easy to please
- Derived: *John's easiness to please

What's the diff between easiness to please and eagerness to please? Non-transparent semantics.

Case/modification:

- John's unmotivated criticism of the book
- * John's unmotivated criticizing the book

Previously, generalizations were captured by transformations.

Similarities between verbal and nominal forms.

- Mary refused to come to the party
- Mary refused the invitation

Nominalize:

- Mary's refusal to come to the party
- Mary's refusal of the invitation

The distribution of arguments for refuse as V and N are the same. Say that refuse has properties that are "category neutral". In particular, say that its selectional/subcategorial properties are category neutral. The lexicon can have category-free roots. n.b., certain semantic features may be associated with particular environments or categories or features.

Easy vs eager:

- have different complement properties
- eager takes an S complement
- does easy take an S complement?
 - John is easy to please
 - It is easy to please John
 - To please John is easy
 - Claim: last is base form, so easy doesn't take an S compl, so we can't get "john's easiness to please."

Failure of some nominalizations follows from category-neutral lexical properties.

1. John grew tomatoes.
2. Tomatoes grew.
3. The growth of tomatoes.
4. *John's growth of tomatoes. (for reading: john's growing of tomatoes)

(nominalized) grow doesn't take an agent. Just an internal argument. If we want "John grew tomatoes," we need to use a causative element.

We can't get the intended reading for "John's growth of tomatoes," since xformations (e.g., +caus) apply after lexical processes.

21.1 Questions

- || 1. John looked up the information.
- || 1'. John looked the information up.
- || 2. John's looking up of the information.
- || 2'. * John's looking of the information up.

Transform takes us from $1 \rightarrow 1'$. But derived nominal is done in the lexicon, so it has to happen before this transformation. Assume that the transformation only applies to verbs. Then we don't get 2'.

But...

- || 1. The Roman's destroyed the city.
- || 1'. The city was derived by the Romans.
- || 2. The Roman's destruction of the city
- || 2'. The city's destruction by the Romans.

If lexical xform (\rightarrow derived) has to come before passive, then we can't say that we first passivize ($1 \rightarrow 1'$) and then derive nominalization ($1' \rightarrow 2'$). But we could first nominalize, and then apply some transformation that is common to $1 \rightarrow 1'$ and $2 \rightarrow 2'$.

Reminder: what questions do we want to answer?

- Why is this "lexicalist"
- Why does x-bar notation get introduced?

21.2 Continuing from last time

1. John grew tomatoes.
2. Tomatoes grew.
3. The growth of tomatoes.
4. *John's growth of tomatoes. (for reading: john's growing of tomatoes)

In (1):

- || John cause [tomatoes grow] \rightarrow John's grows tomatoes.

But we can't use that transformation to produce (4), since lexical processes come before transformations.

Why is this "lexicalist"? Because we're modifying the base to produce predictions that were previously done with transformations.

So what's done by the lexicon? And what by the base?

We want to represent "refuse" in a category neutral way, with the possibility of taking an S complement:

- || refused [to leave]
- || refusal [to leave]

In previous categorial system, each category has a unique distribution. We can define the categories distributionally. But to encode information in a category neutral way, we can't use the traditional categorial system. So abandon the old definition of the distributional categorial system?

Basic rules for (some subpart of) the base:

- $VP \rightarrow V$ Compl (compl can be NP, S, etc.)
- $AP \rightarrow A$ Compl
- ...
- $NP \rightarrow N$ Compl

Since nominals are base-generated, rather than derived via transformations, it is no longer distributionally different from verb. So just define:

- $XP \rightarrow X \text{ Compl}$

Refine to:

- $XP \rightarrow \text{Spec } X'$
- $X' \rightarrow X \text{ Compl}$

|| [XP Spec [X' X compl]]

We can use features like [+N] and [+V], but they're not part of the categorial system.

21.3 Other stuff

3 criteria:

- productive
- transparent semantics/interpretation
- internal structure (NP vs VP)

are the same/similar to conditions for derivational vs inflectional morphology.

Lexicon:

- generative
- generates words
- outputs are the axioms of syntax

Lexicalist hypothesis:

- lexicon is separate from syntax
- (syntax has nothing to do with the internal structure of words)

After this paper: people use those 3 criteria to decide what to put in the lexicon/syntax.

Different notions of lexical include . . .

- "Lexical" \rightarrow pertaining to open-class categories.
- "Lexical" \rightarrow idiosyncratic (bloomfield, idioms)
- "Lexical" \rightarrow syntactic primitives

In "lexicalist" theories, these 3 things are hypothesized to be related, with a single lexicon that generates the primitives of syntax.

n.b.: there's no a priori reason to believe that these 3 things are all handled by the same system.

21.4 Looking forward to next time (Marantz)

|| [Syntax [. PF LF]]

Get rid of the lexicon.

Lexicalist hypothesis: reanalyze nominals etc \rightarrow modify the base (categorial component) \rightarrow everything has the same distribution \rightarrow x-bar schema.

Further developments after chomsky's paper: lexicon has ideosyncratic elements; etc.

Marantz goes back and says that if you actually look at remarks on nominalization, in the context of current theory, you see that you should use a different structure & get rid of the "lexicon" as it existed in current syntax theory.

22 Marantz & the death of lexicalism

Point of departure:

|| [Lexicon [DF [SS PF LF]]]

Lexicalism:

- words are created in the lexicon, by processes that are distinct from syntax.
 - some phonology is lexical
 - some structure/meaning correspondances are lexical
 - some phonology & structure/meaning correspondances are syntactic

All of these can be phrased as questions about modularity: how to divide things into modules (syntax & lexicon). How do we justify the distinction? If their rules for putting things together are really distinct. If they're not, then collapse them?

22.1 A non-lexicalist model

Try taking the basic functions of the lexicon & splitting them apart. Basic functions of the lexicon:

- provides syntactic primitives
- keep track of idiosyncratic properties

Replace lexicon with non-generative lists.

- list 1: primitives for the syntax = morphemes = bundles of abstract features.
- list 2: vocabulary = pairings of phonology with conditions
- list 3: encyclopedia = pairings of semantics with conditions

22.2 Excursis

Comparatives:

- smart → smarter
- intelligent → more intelligent

(rule: >1 syllable: more)

What side is it on? [-er smart] [more intelligent] (adjectives are on left?) So how does "-er" get to the right?

Try adding adverbs and see what happens:

- rich
- richer

Compare:

- more amazingly rich
- amazingly richer

Different meaning: where did the [-er] originate?

22.3 Does the lexicalist model work?

Lexicalist model predicts:

- word-sized objects: lexicon, potentially idiosyncratic
- larger objects: syntax, compositional & transparent

Are idioms in the lexicon or the syntax? Is the idiosyncrasy of idioms different (qualitatively) from the idiosyncrasy of words? What about idiosyncratic compounds?

- There's a syntactic domain for special interpretation
- It's the same for words & for idioms

22.4 Remarks Revisited

Why is "Remarks on Nominalization" against lexicalism?

- It was lexicalist, in so far as it was not transformational. It modified the categorial component.

Consider the tomato growth argument:

- Tomatoes grow
- John grows tomatoes
- The tomatoes' growth
- * John's growth of tomatoes

Before, we said the 4th was allowed by transformational. Now we see that 4th might be allowed by lexicalist approach, too.

Different approach: height of attachment

- \exists different points where we could apply the nominalization

Lexicalist system:

- 2 generative modules
 - lexicon
 - syntax

Do the modularity arguments for this division hold?

Alternative:

- divide the stuff that the lexicon did into 3 non-generative lists:
 - list of atoms
 - vocabulary
 - encyclopedia

cf:

- Tomatoes grow
 - *The city destroyed
 - John grows tomatoes
 - The Romans destroyed the city
 - The tomatoes' growth
 - The Roman's destruction
 - * John's growth of tomatoes
 - The city's destruction

Different roots represent different kinds of things:

- grow: internally caused change of state
- destroy: external change of state, implies agent/cause
- break: result of change of state

Different kinds of verbs..

- v1 = licenses agent
- v2 = does not license agent

So grow uses v2 and destroy uses v1. We assume that agentive verbs can't go with v2. That gives us the right judgements for the verb examples.

Then explain the nominalization thing by saying that we can only nominalize things that don't contain v1 (for derived nominalization)?

Different height at which the nominalizing thing attaches gives you differences in properties. (derived vs gerundive)

In general, we can attach possessors to NPs. The verb wants an agent, and it can take over the possessor & make it a possessor? n.b. that in:

|| **John's performance of the play**

We *can* interpret John as agent, but we can also interpret him as being connected in some other way (e.g., the manager of the play).