Bare Bones Syntax

An insane idea

Adam Szczegielniak
adam.s@post.harvard.edu
LingLunch MIT Linguistics, April 6 2006

Syntactic computations

- Syntax computations are reduced to a minimum
  - Set formation:
    - Recursive
    - Hierarchical
    - Unordered
    - Merological

Lexical Features

- Lexical Items posses sets of features:
  - Phonological
  - Morphophonological
  - Semantic
  - Syntactic
    - C-selection
    - Long distance dependency

What syntax operates on?

- Syntactic features
  - C selection
  - Long Distance Dependency (EPP type feature)
- There is no a-priori reason to assume that syntax can read, or operate on phonological, morphological or semantic features.

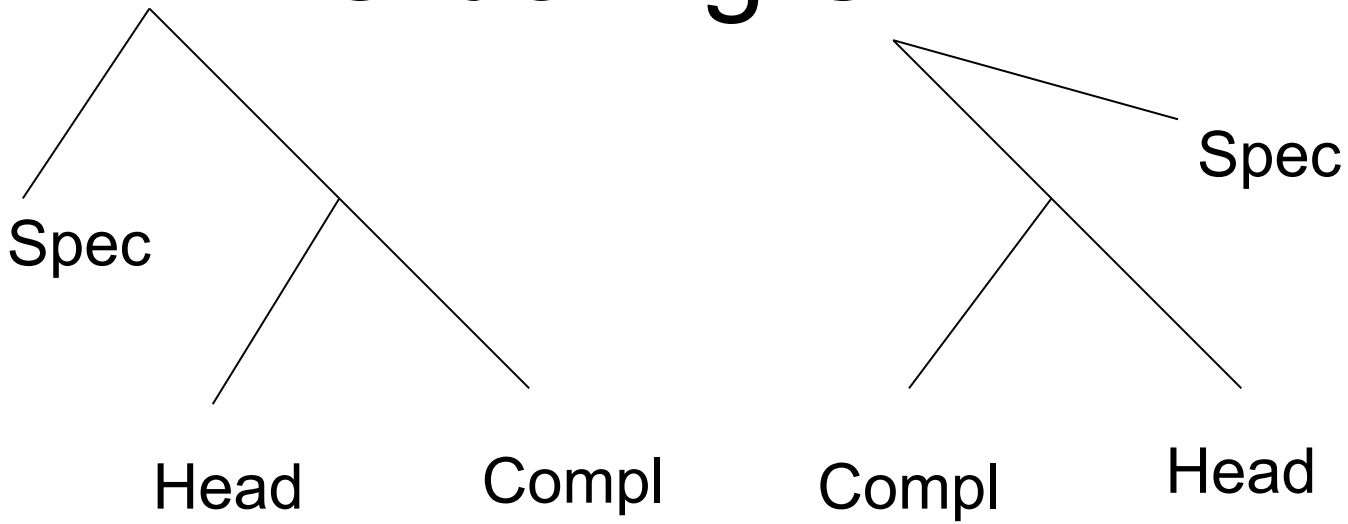
What does syntax do?

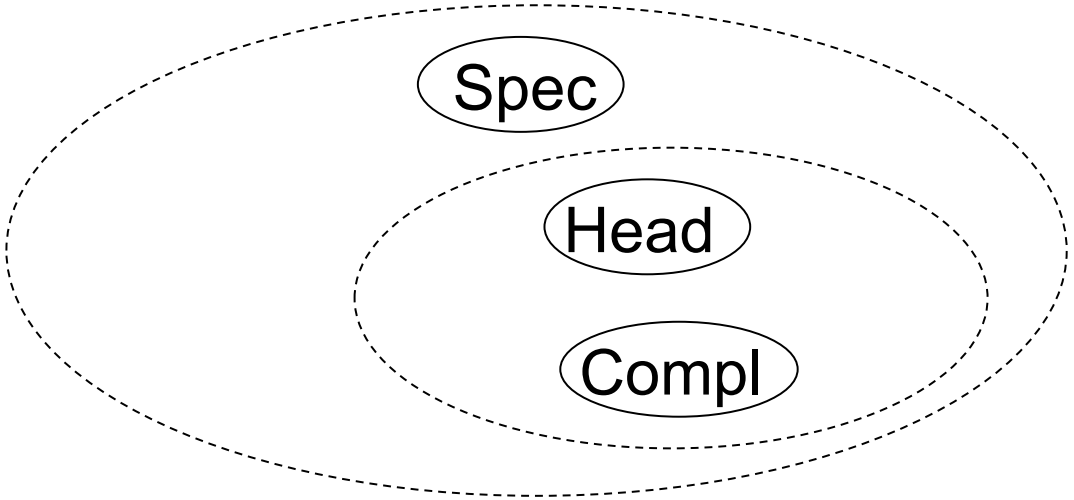
- Create sets
  - Hierarchical
  - Recursive
- Question 1: are these sets t-sets, or merological ones
- Question 2: are these sets ordered or unordered

Types of sets - ordering

- There is no a priori reason to assume that the representation in the syntax consists of ordered sets.
  - Syntactic relationships are not sensitive to order:
    - Spec-Head
    - Head-complement

Ordering CD





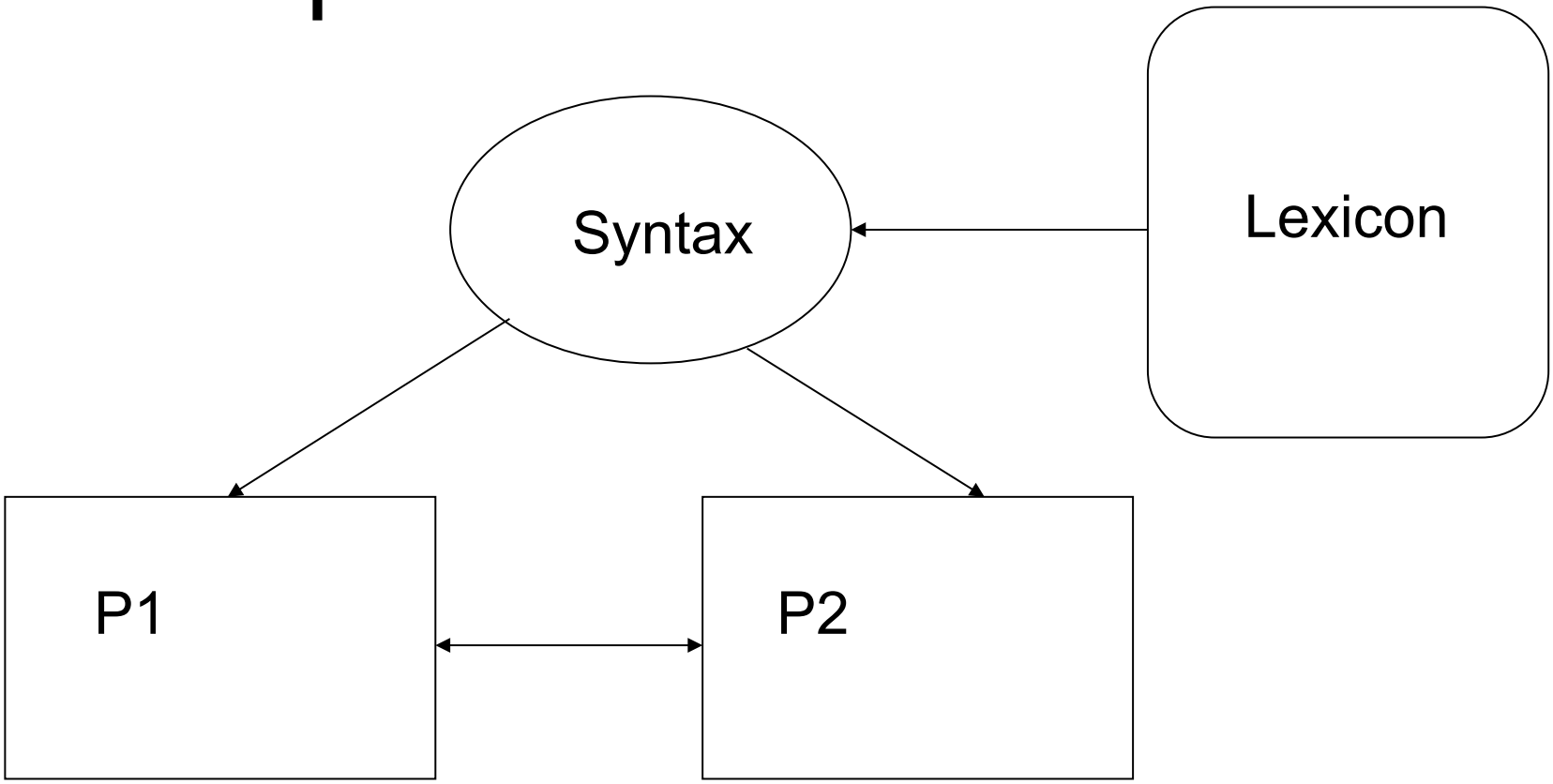
Sets are unordered

- Let me assume that sets generated by the syntax are unordered
  - Simpler assumption, no extra machinery needed, just a set building algorithm like Merge
  - Supported by evidence that hierarchy plays a role in syntax not ordering

Why merological

- Merological sets
  - Can overlap
  - No impenetrability
- Standard t-sets are actually a special case of merological sets.
- To assume that syntactic representations involve t-sets is an additional assumption.

Proposed model



Lexical Items

- Lexical items are bundles of features
  - Phonological
  - Semantic
  - Morphological
  - Syntactic
- Syntax reads syntactic features:
  - C-selection
  - Long distance dependency

Nature of syntactic features

- Syntactic features are related to other features:
  - C-selection:
    - Semantic features
  - Long distance dependency:
    - Wh features
    - Topic
    - Case
    - Scope
- Syntax cannot read those dependencies

Syntax only reads syntactic features

- It is a stipulation that syntax reads features that are not part of its computation
  - To build recursive representations you do not need wh-features, case, etc…
- Let us assume that syntax only operates on features that are necessary to build recursive sets:
  - C-selection
  - Long Distance dependencies

Is syntax static or derivational?

- A set can have history, just like any equation

$$E = m \times c^2$$

- There is no derivational history there, yet we know the subparts of E

An equation has many solutions

- Take a simple equation

  $X^2 = 3+(X-1)$

  $X=2$

  There is no other solution

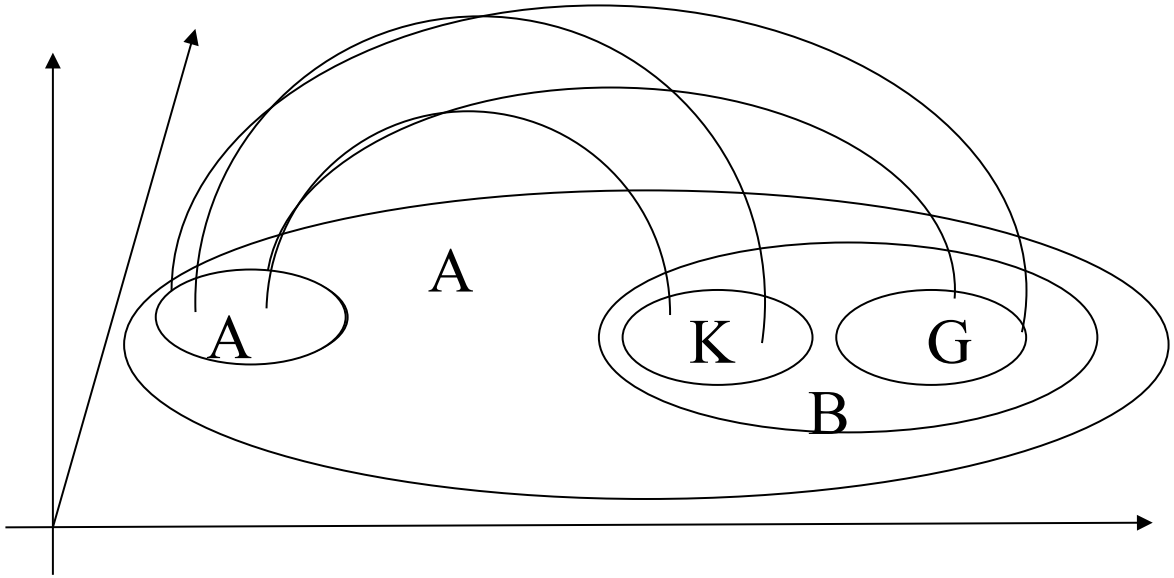
- Take a more complex equation

  $X^2=X+Y$

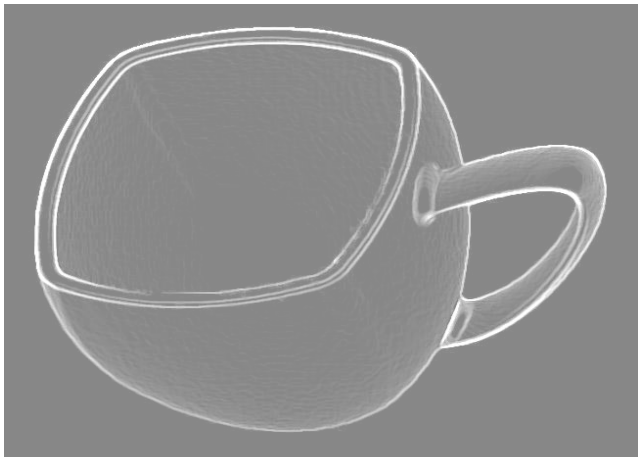
  X=2, Y=2; X=4. Y+12…

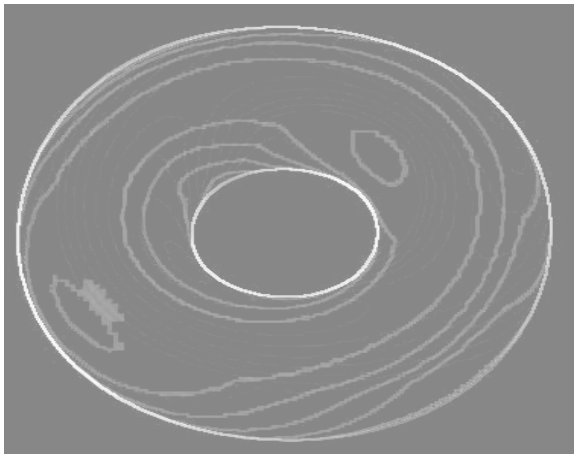
Syntax generates an equation that has multiple solutions

- Syntax builds sets
- Sets are unordered, and have topological properties
  - Homeomorphic symmetry
  - Long Distance Dependancy feature interpret as a hole (literally in 3D space)


A
A
K
G
B

Topological transformations where holes count





No movement

- In this system there no movement in the syntax
  - There are just different solutions to the same equation, one of them is that A given X can be in two places at the same time.
  - In a sense this is saying what looks like a X and its copy is the same element in two positions (Remerger, Frampton 2006)

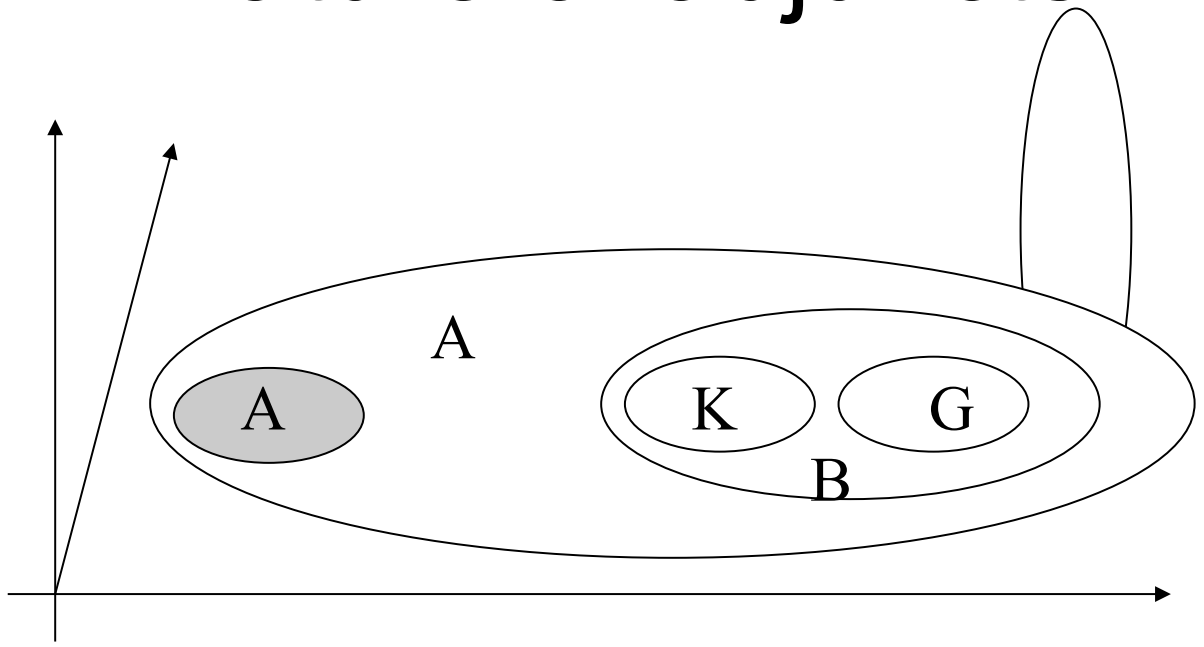
No movement - no island violations

- The only thing syntax encodes is the existence of a long distance dependency.
- The only restriction that there is that the long distance dependency feature can only be interpreted as one element being in two positions at once (no multiple movement to the same position)
  - This is a result of the topological nature of the set.

Island violations exist

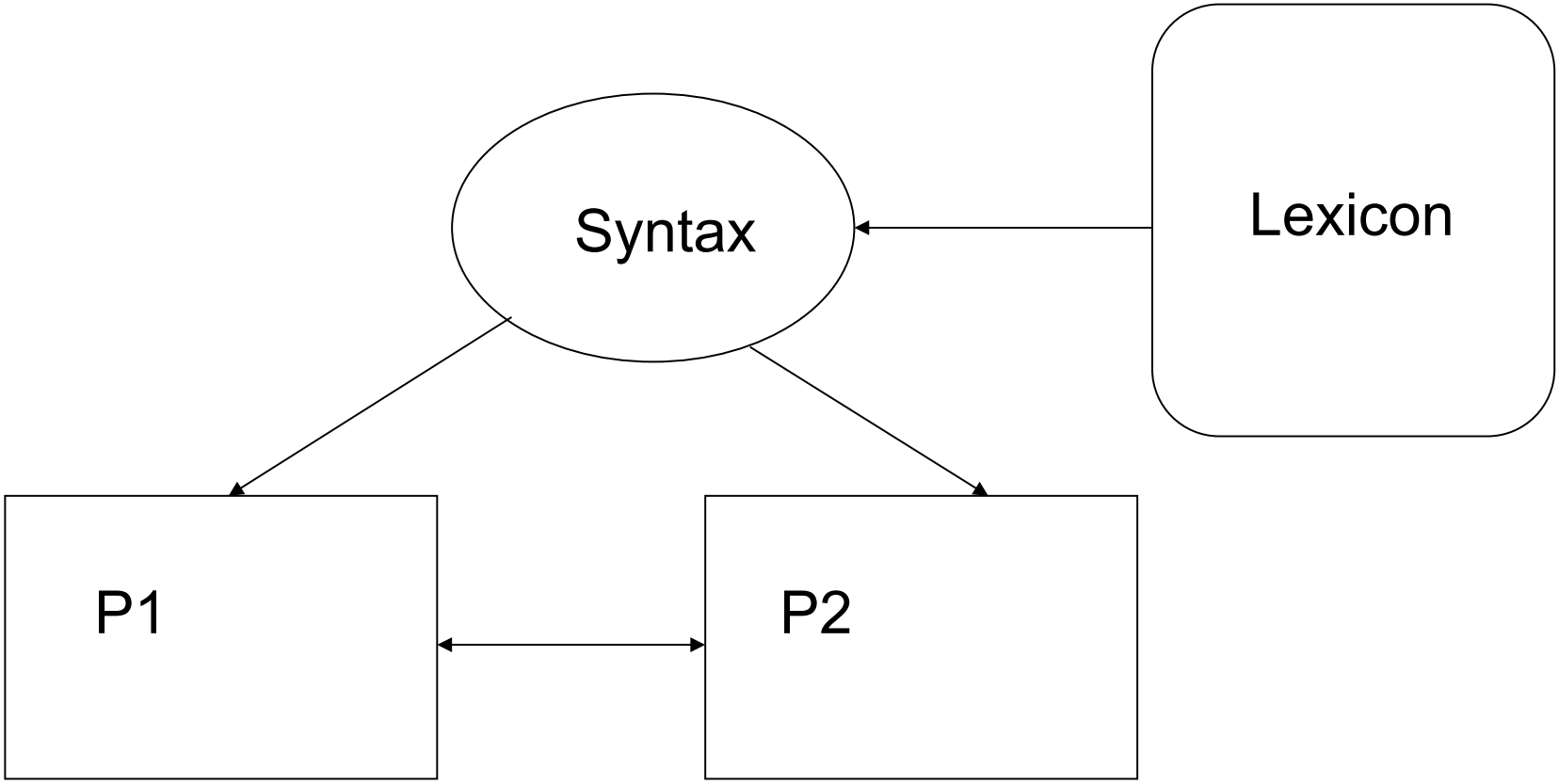
- If there is no movement in syntax, what is the nature of restrictions on displacement
  - Inability to read syntactic structure by P1 and/or P2
    - Adjunct Islands
  - Miscommunication between P1 and P2
    - Subjacency Islands

Nature of adjuncts



Syntax operates only on C-selection and LDD features, adjuncts are added in a different dimension and to be read have to be linearized (Chomsky 2002)

Linearization



- P1 linearizes hierarchical structure
  - It reads the representation in chunks due to memory constraints
  - That is what phases are

Nature of P1

- Linearizes (maybe a la LCA)
- Is a phonological loop (Baddelley 1986)
- Reads phonological and morphological features
- Cannot read semantic features
  - But it can communicate with P2 saying it has an unreadable feature.

Nature of P2

- Computes thematic relations
- Computes variable operator relations
- Reads
  - Theta roles
  - Wh features
  - Case
  - Topic/Focus

Typical A’ movement

- In the syntax there is a LDD feature on C linked to +wh
- P1sees that there is an LDD feature but cannot read +wh (a semantic feature)
- It sends info to P2
- P2 matches it with a +wh word
  - Signals it to P1 which decides where to pronounce it.

Covert vs. overt wh movement

- Covert wh-movement wait for pronunciation until you linearize everything
- Overt do it ASAP
- Pestsky's single Spell out model

Why covert movement sensitive to ECP

- Huang 1982 pointed that covert movement not sensitive to subjacency but sensitive to ECP
- Adjuncts have to be linearized by P1
  - When P1 queries P2 for a wh candidate there is none since no linearization P2 cannot read adjuncts

Covert movement not sensitive to Subjacency

- P1 queries P2 and a candidate is available
- But P1 does not have to hold in memory the PF properties just the fact that when it hits the right moment to pronounce it
- P2 has to hold in memory the variable operator construction

Overt movement sensitive to subjacency

- *‘Which linguist admirers of t were ignored’
- P1 has to hold in memory the XP in order not to linearize it twice
- Memory restriction, old bounding nodes:
  - CP/TP
  - DP

Why not other phrases?

Good question

Sensitivity to N features?

Cyclic movement

- There are no phases in syntax
- P1 and P2 read syntax in chunks
- P1 outside in (top down in tree terms)
- P2 inside out (bottom up in tree terms)
- Chunk for P1: CP (or LP in Rizzi's terms)
- Chunk for P2: vP
- Both P1 and P2 have to keep in memory the LDD - cyclic movement to the edge of every phase.

QR

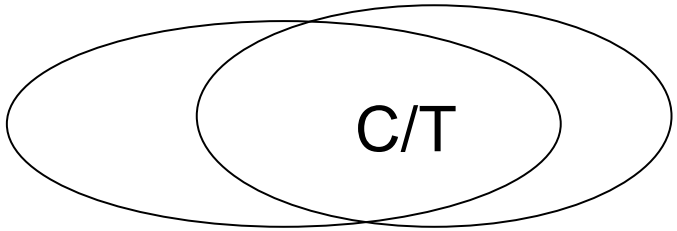
- Nothing prevents form the LDD feature to be on a lexical head
- Quantifiers, if they are selected to have wide scope will have LDD, but since CP will be already linearized by P1 movement will be covert
- Sensitivity to Adjunct Islands

Superieriority - the dual nature of case

- English has superiroty violations
- Russian doesn't (Rudin 1987, Fedorenko 2005, etc)
- In English Case is a semantic feature
- In Russian, Polish, etc it is both a morphological and semantic feature
- P1 and P2 can match on 1-1 basis multiple wh-phrases when both read case
- In English only P2 reads case - first wh to be linearized is moved
- Bulgarian: is a mystery to me

That trace effects

- What is the nature of the relationship between C and T?
- In the syntax same thing - sets are merological



That-t effects continued

- P1 reads C/T encounters a bundle of semantic features that are separate from morphological ones, the morphological set is interpreted as T, the semantic set is represented as C
- If the features are inseparable, we have what Chomsky would call C-T (see also Meyer 2006)

That t effects part 3

- C and T are separated for example when there is an overt complementizer in English
  - Subject raises to T and is frozen since there is no T in P2, interpreted as in C
- C and T are fused, English null complementizers, pro-drop languages (no EPP), agreeing Complementizers (Bavarian)
  - No that trace effects since no mismatch between P1 and P2 representations. Subject not frozen

That T effects and Bounding

- CP is a bounding node in Italian
- Italian has pro-drop
- Italian has no that-t effects
- Maybe a possible connection

There expletives

- How come the verbs agree with the NP that follows it
- There are many men in the room
- P1 linearizes
  - ‘There’ has no agr features
  - Verb has agr features
  - Following NP has agr features
  - No need for AGR phrases
  - Agreement ASAP does not mean the NP has to precede the V

Processor - Syntax

- P1 and P2 are the processor
- P1 and P2 are also what we would call syntax
- What I call syntax, is just a set representation

Processing

- Garden paths (Frazier 1977)
  - The horse that raced past the barn fell
- Memory constraints (Gibson 1998)
  - N CP N V (slower)
  - N CP V N

  P1 processes Relative clauses with no input from P2

Konieczny (2000)

- a. He has the book, that Lisa yesterday bought had, laid down.
  - Verb read faster
- b. He has the book laid down, that Lisa yesterday bought had.
  - Pronoun read slower
- P1 linearizes but has top down expectations from P2

Ellipsis

- Ellipsis makes use of semantic parallelism (Merchant 2001)
  - Comprehension of elided structures P2 maps to Syntax
  - Other Evidence - sluicing in Polish (Szczegielniak 2005)
  - Semantic but no phonological activation of elided structures (work in progress with with Fedorenko & Gibson)

How does P1 and P2 read syntax

- P1 outside in, in chunks the CP/TP complex
- P2 inside out, in chunks the vP complex
- What if there is more than one CP/TP and vP?
- Read in parallel (only production)
  - Anti locality…

Deficient phases

- Syntax has no phases
- vP is deficient in passives (Chomsky 2002)
- This is not a syntactic deficiency
- P2 can have truncated v
- This is a conceptual impoverishment
- That is why passive acquired late (Hirsh and Wexler 2005)

Conclusions

- Syntax is an equation
- Its read by P1 and P2in chunks/phases
- Can be misread
- P1 and P2 communicate but only as far as things they cannot read
- …