Language is flat!

Linearity and Hierarchy from Discourse to Phoneme

Dafydd Gibbon

Bielefeld University
Bielefeld, Germany

2017-11-15

Tongji University, Shanghai, China

Language is flat!

Linearity and Hierarchy from Discou

Well, fairly flat!

Dafydd Gibbon

Bielefeld University
Bielefeld, Germany

2017-11-15

Tongji University, Shanghai, China

Overview (1)

- Design Features of Language and Speech
- Multilinear Grammar / Ranks and Interpretations
  - On the Architecture of Language, Speech and Gesture
  - A ‘Layer Cake’ Model
- Occam’s Razor (and flatness as the null hypothesis)
  - A null hypothesis for modality interpretation
  - A null hypothesis for the semantic-pragmatic interpretion
- Conflict with the ‘recursion alone’ design feature
  - The Recursion Debate
  - A family of recursions; examples of recursive compositionality
  - A well-defined scale of syntagmatic structures by complexity
  - What is recursion?
  - Recursive explanations
  - There are many kinds of recursion outside human language

Overview (2)

- Multilinear Grammar: Summary and Challenges
- Uniqueness
  - Uniqueness: Semantic and Prosodic Perspectives
  - Finiteness, linearity:
    - Dialogue (discourse grammar)
    - Sentences (‘syntax’)
      - Serial patterns – transformational power?
      - People hate recursion!
    - Words (‘morphology’)
      - Mismatch: morphological seriality vs. semantic hierarchy
      - Some examples of recursive compositionality
    - Morpheme structure (‘(morpho)phonology’)
      - Which is more complicated, Mandarin or English syllable structure?
    - “The human brain is a finite machine, albeit a complex one.”
- Comments, Conclusions

Design Features of Language and Speech

Starting point

- What are the ‘design features’ of language, in particular speech, spoken language?
  - That is:

    What distinguishes speech from other human abilities?

    What distinguishes speech from other forms of human communication?

    What distinguishes speech from the different kinds of animal communication?

- Choice of design features has many consequences:
  - for the complexity of speech

    cf. the ‘recursion discussion’ of the past decade and a half)
  - for the understanding of the evolution of speech

Design Features / Key Features (1958: 574)

- Hockett's (original) Design Features:
  - duality: meaningful units (morphemes) vs. coding units (phonemes)
  - productivity: the ability to produce novel utterances
  - arbitrariness: no relation between the forms and meanings of words
  - interchangeability: exchange of dialogue roles
  - specialisation: no cause-effect relation between meaning and form
  - displacement: referents outside the communication situation
  - cultural transmission: strong influence of culture rather than genetics (unlike animals)
- These design features are essentially structural:
  - they ignore language and speech processes:
    - time production and perception
    - memory space requirements

Design Features / Key Features

- Chomsky's single design feature:

RECURSION

- In Hockett's terminology:
  - recursion would be one form of productivity
  - vs. repetition, iteration

Design Features / Key Features

- But what about the architecture of language and speech as a whole?

- One popular conception of the architecture of language is the Module+Interface metaphor:
  - a collection of modules connected by 'interfaces'

- Problems:
  - Modules:
    - How many modules are there? Sometimes seems arbitrary
    - Which constraints apply to modules and interfaces?
    - Do all modules connect to all other modules?
    - Are modules intrinsically different or of the same basic type?
  - Interfaces:
    - What exactly is does the interface metaphor mean?

Multilinear Grammar / Ranks and Interpretations

On the Architecture of Language, Speech and Gesture

Multilinear Grammar: Ranks and Interpretations

Gibbon, Dafydd and Sascha Griffiths. 2017. Multilinear Grammar: Ranks and Interpretations. *Open Linguistics* 2017 (3): 265–307

De Gruyter Open, Open Linguistics

Multilinear Grammar: Ranks and Interpretations

- Null hypothesis (or: Occam's Razor):
  - all modules of language have the same kind of structure
  - all interfaces are the same kind of function
- Constraints:
  - all modules of language are arranged as a finite set of ranks

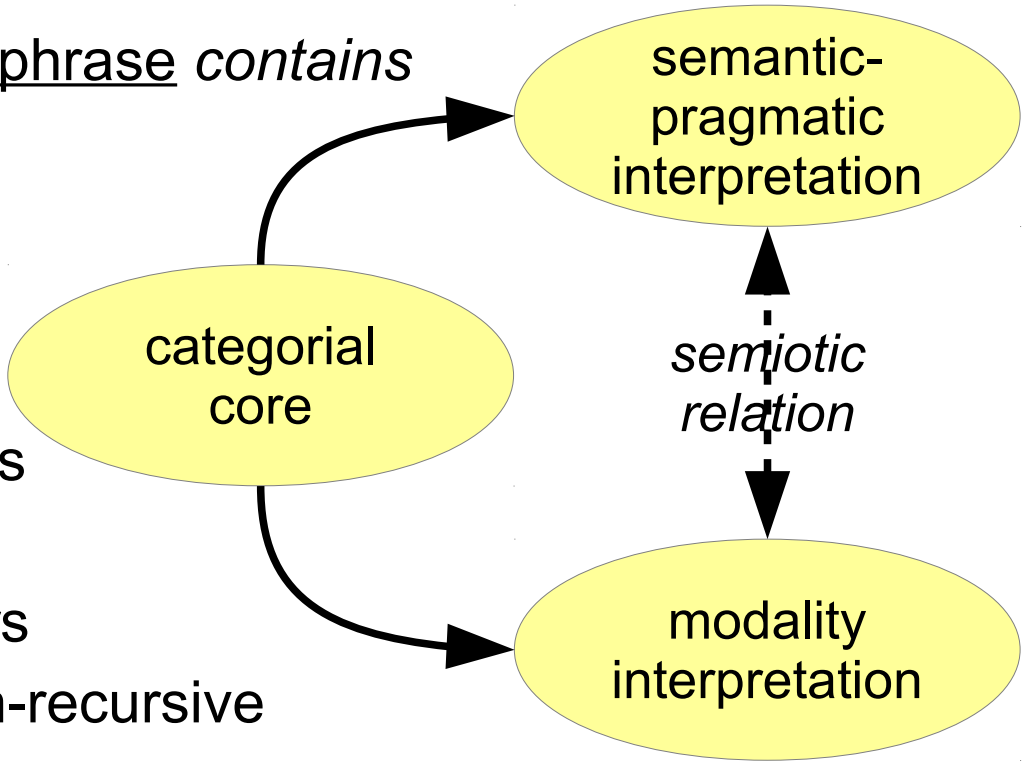
    discourse *contains*

    utterance / text *contains*

    sentence / clause / phrase *contains*

    word *contains*

    morpheme
  - each rank has a semiotic structure:
    - each semiotic component is interfaced only with its upper and lower neighbours
    - the architecture is thus non-recursive



Multilinear Grammar: Ranks and Interpretations

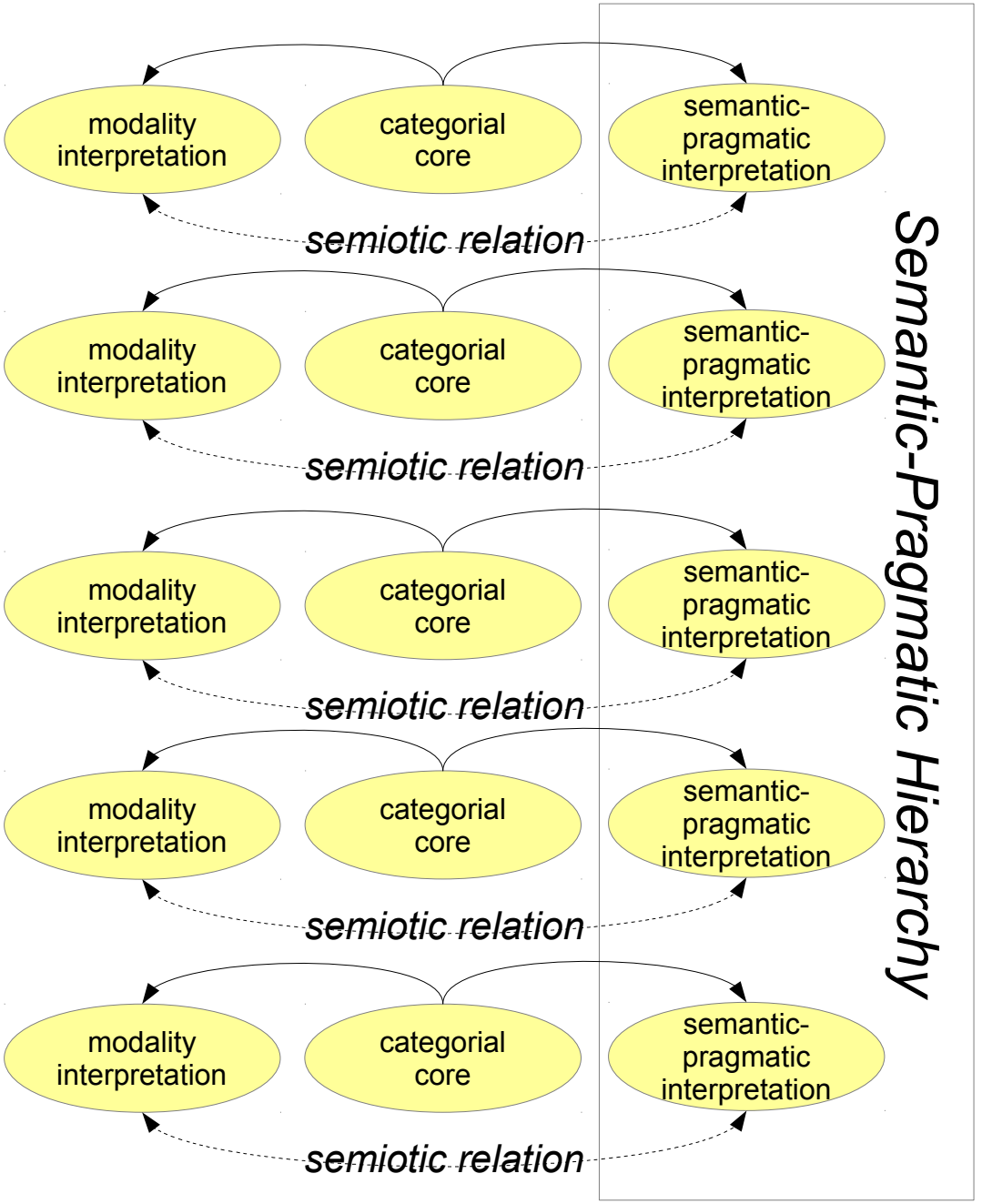
Discourse:

Utterance / text:

Sentence / clause / phrase:

Word:

Morpheme:



Multilinear Grammar: Ranks and Interpretations

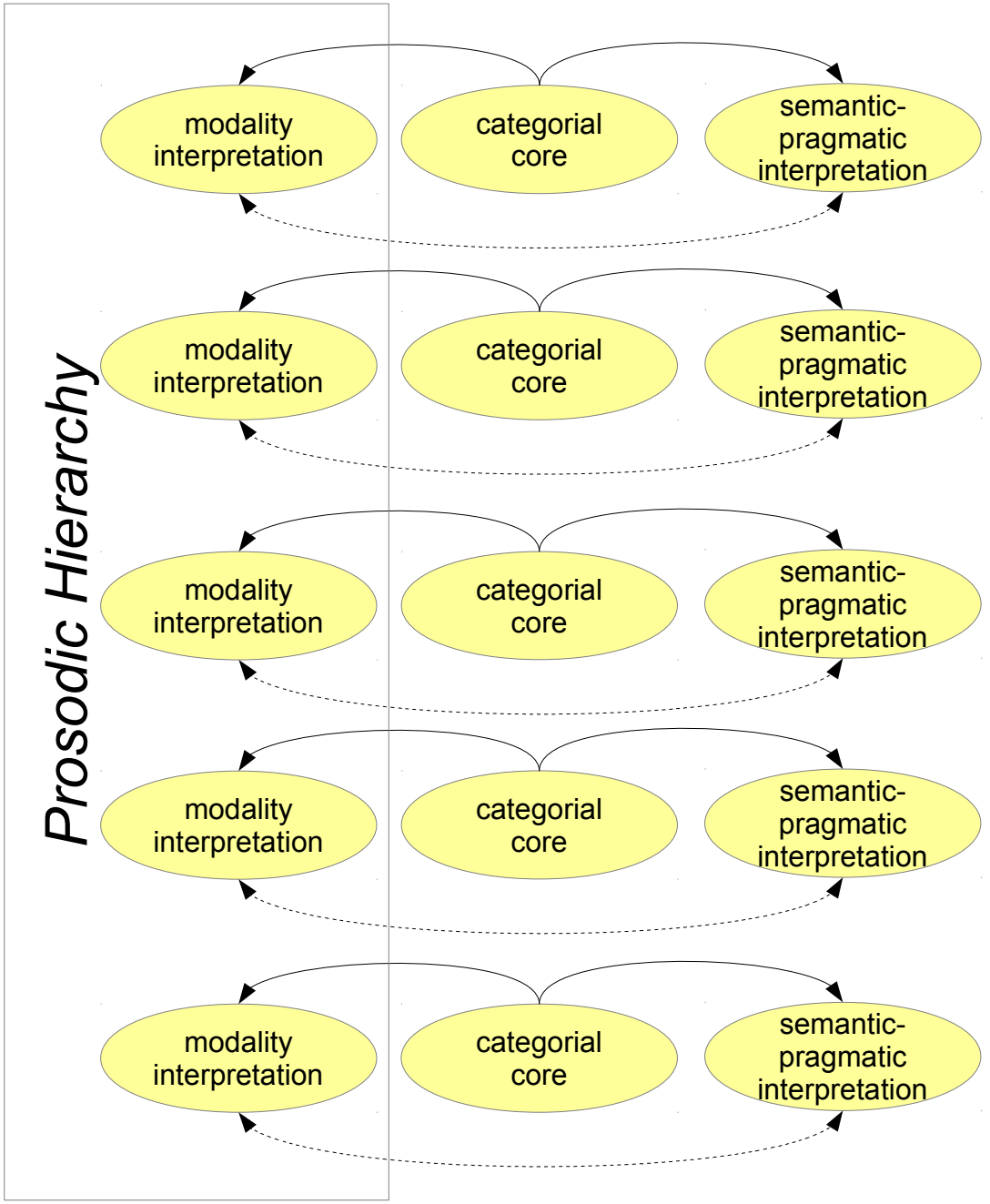
Discourse:

Utterance / text:

Sentence / clause / phrase:

Word:

Morpheme:



Multilinear Grammar: Ranks and Interpretations

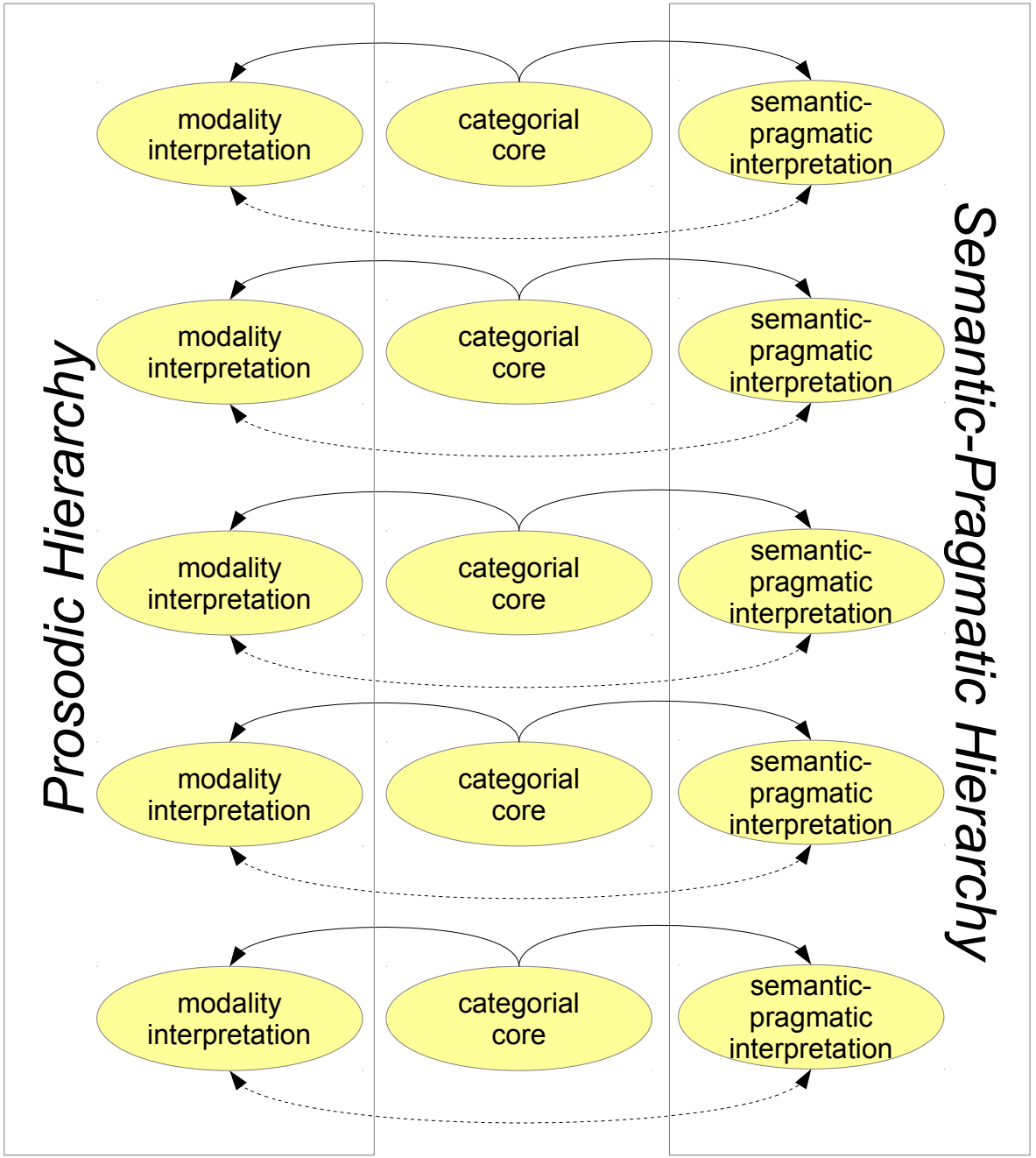
Discourse:

Utterance / text:

Sentence / clause / phrase:

Word:

Morpheme:



A ‘Layer Cake’ Model

Multilinear Grammar: Ranks and Interpretations

- Each rank has a ‘flat’ grammar for the categorial core

  and may be modelled by

  - right-branching grammars
  - iterative grammars
  - finite state automata / machines

Multilinear Grammar: Ranks and Interpretations

- Each rank has a ‘flat’ grammar for the categorial core

  and may be modelled by
  - right-branching grammars
  - iterative grammars
  - finite state automata / machines
- The ‘flat grammar’ hypothesis represents
  - the simplest kind of grammar in the *Chomsky-Schützenberger Hierarchy of Formal Grammars*
  - the most efficient kind of grammar to process
    - with finite space (memory) requirements
    - and linear time (production, perception) requirements

Multilinear Grammar: Ranks and Interpretations

- Each rank has a ‘flat’ grammar for the categorial core

  and may be modelled by
  - right-branching grammars
  - iterative grammars
  - finite state automata / machines
- The ‘flat grammar’ hypothesis represents
  - the simplest kind of grammar in the *Chomsky-Schützenberger Hierarchy of Formal Grammars*
  - the most efficient kind of grammar to process
    - with finite space (memory) requirements
    - and linear time requirements
  - in contrast to a general phrase structure grammar, which has
    - non-finite memory requirements (for arbitrary centre-embedding)
    - non-linear (polynomial or even exponential) time requirements
    - and is therefore an implausible model of speech

Multilinear Grammar: Ranks and Interpretations

- Each rank has a 'flat' grammar for the categorial core

  and may be modelled by
  - right-branching grammars
  - iterative grammars
  - finite state automata / machines

The No-Centre-Embedding Constraint

- The 'flat grammar' hypothesis represents
  - the simplest kind of grammar in the *Chomsky-Schützenberger Hierarchy of Formal Grammars*
  - the most efficient kind of grammar to process
    - with finite space (memory) requirements
    - and linear time requirements
  - in contrast to a general phrase structure grammar, which has
    - non-finite memory requirements (for arbitrary centre-embedding)
    - non-linear (polynomial or even exponential) time requirements
    - and is therefore an implausible model of speech

Multilinear Grammar: Ranks and Interpretations

- Each rank has a 'flat' grammar for the categorial core

  and may be modelled by
  - right-branching grammars
  - iterative grammars
  - finite state automata / machines

The No-Centre-Embedding Constraint

- The 'flat grammar' hypothesis represents
  - the simplest kind of grammar in th *ger Hierarchy of Formal Grammars*
  - the most efficient kind of grammar t
    - with finite space (memory) requirem
    - and linear time requirements
  - in contrast to a general phrase structure grammar, which has
    - non-finite memory requirements (for arbitrary centre-embedding)
    - non-linear (polynomial or even exponential) time requirements
    - and is therefore an implausible model of speech

Limited exception

If there is extra time and memory, as in writing and rehearsed speech

Null hypothesis for modality interpretations

- Ranks have ‘flat’ grammars for modality interpretations

  and may also be modelled by
  - right-branching grammars
  - iterative grammars
  - finite state automata / machines

- This null hypothesis for modality interpretation is compatible with the models of the prosodic hierarchy developed by Selkirk, Hayes, Nespor & Vogel, Féry and many others:
  - finite set of ranks (levels, layers)
  - non-recursive relation between ranks
  - each rank has flat structure (in the sense defined above)

A null hypothesis for the semantic-pragmatic interpretion

- Contrast to categorial core and modality interpretation:

  'Anything goes!'

- Constraints on semantic-pragmatic structures and processes
  - are shared by all human cognitive structures and processes
  - are not specifically determinants of language
  - can be represented by
    - tree-like hierarchies
    - heterarchies
    - arbitrary graph structures

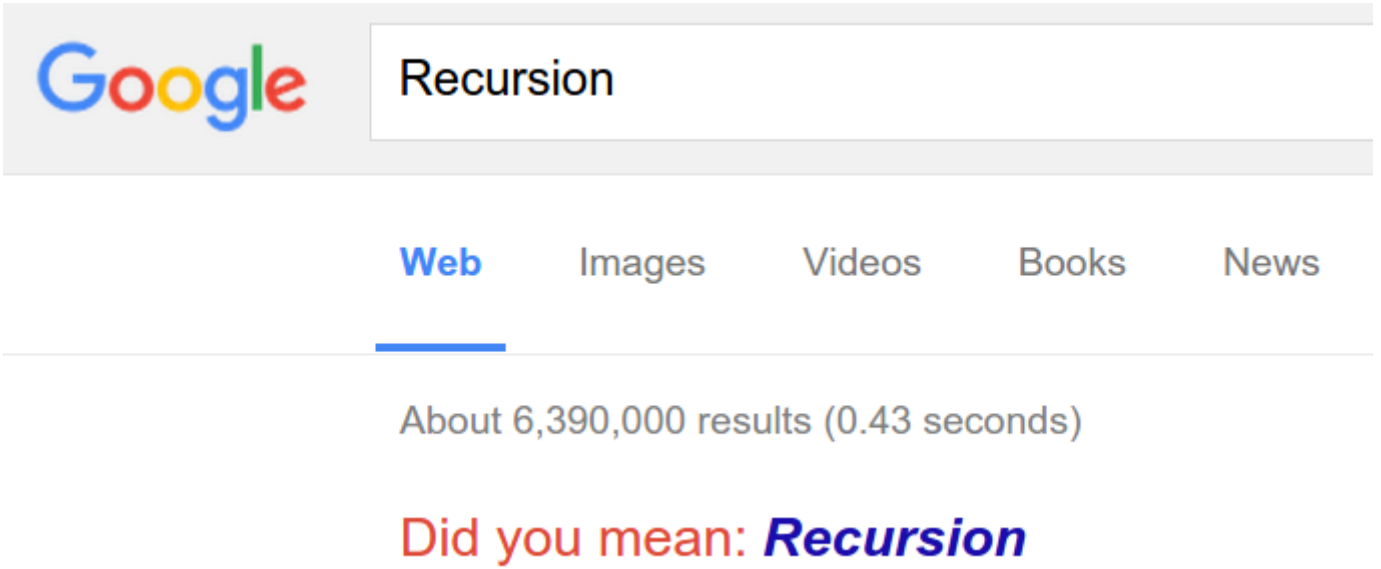
Contrast with the 'recursion alone' design feature

- The Rank Interpretation Architecture:
  - a null hypothesis
  - a complex design feature with several formal and empirically verifiable properties
    - the sentence and its properties are not the only rank
    - the set of ranks is finite:
      - discourse, utterance/text, sentence, word, morpheme
    - the core and modality at each rank are linearly structured
    - each rank has clear time and space processing requirements
    - there are clear constraints on possible 'interfaces'

- The 'architectural approach' contrasts with
  - Hockett's list of independent features
  - Chomsky's focus on sentences and the property of recursion

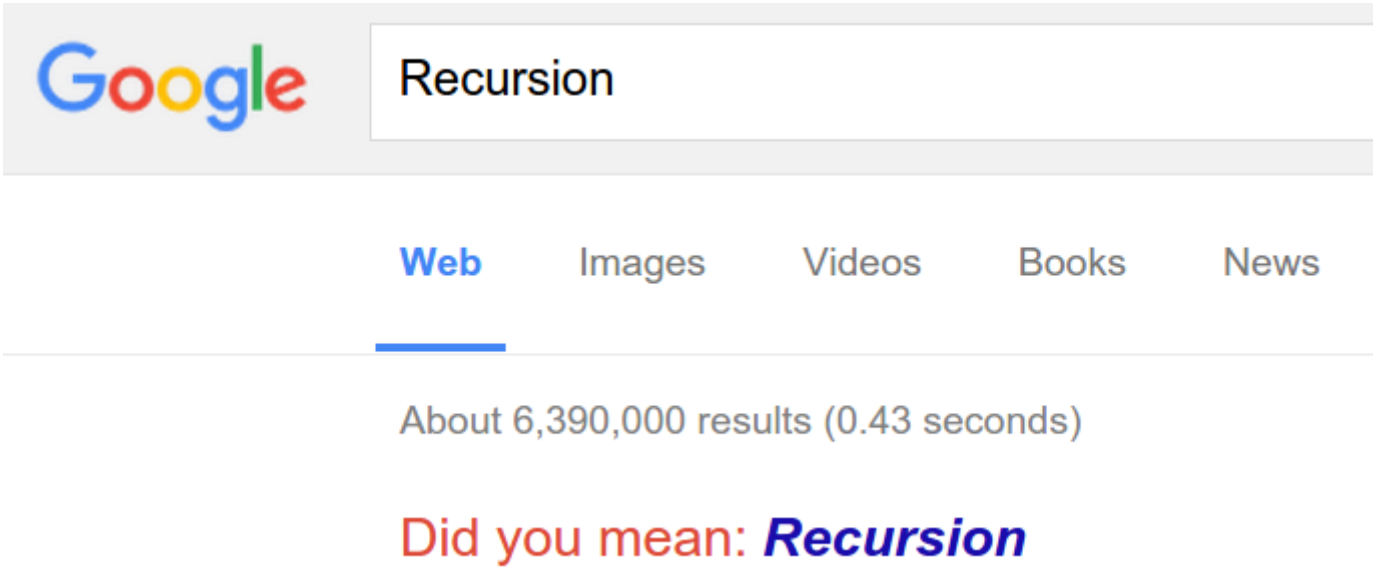
A closer look at recursion

What is recursion?



What is recursion?

(Google likes 'nerdy' jokes)



Recursive explanations

A scientist was explaining in a public talk how the earth did not need a support – it just depended on velocity and gravity.

Recursive explanations

A scientist was explaining in a public talk how the earth did not need a support – it just depended on velocity and gravity.

After the talk an old lady came up to him and said: “Very clever! But I have a better explanation. The world rests on the back of a turtle.”

Recursive explanations

A scientist was explaining in a public talk how the earth did not need a support – it just depended on velocity and gravity.

After the talk an old lady came up to him and said: “Very clever! But I have a better explanation. The world rests on the back of a turtle.”

“Oh really,” replied the scientist politely. And what is the turtle resting on?”

Recursive explanations

A scientist was explaining in a public talk how the earth did not need a support – it just depended on velocity and gravity.

After the talk an old lady came up to him and said: “Very clever! But I have a better explanation. The world rests on the back of a turtle.”

“Oh really,” replied the scientist politely. And what is the turtle resting on?”

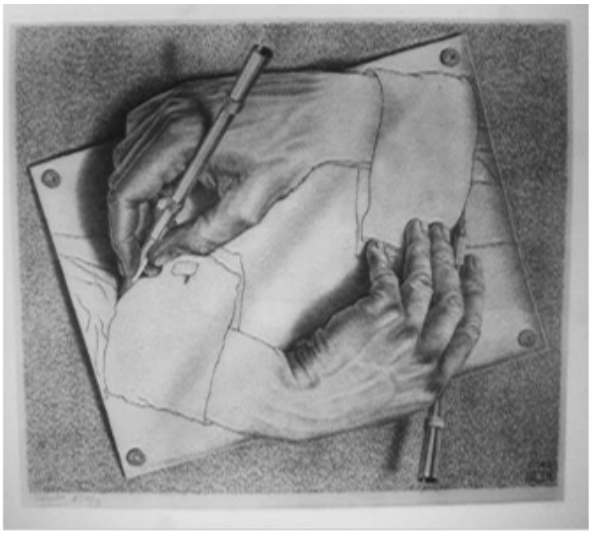
“Young man, isn’t it obvious? It’s turtles all the way down!”

?

There are many kinds of recursion outside human language

Recursive broccoli

Recursive art



Recursion revisited: the 'recursion debate'

- The term 'recursion' is a formal term, from logic, arithmetic, algebra:
  - a *recursive function* maps
    - a (usually finite) set of objects
    - to a non-finite set of objects

      for example from the set of digits to the set of integers
  - a typical recursive function in linguistics or logic is implemented as a grammar in which a symbol which occurs on the left-hand side of a rule occurs
    - on the right-hand side of the same rule (direct recursion):

      S → *if* S *then* S
    - or on the right-hand side of a rule which applies later in the part of the derivation which starts from this rule (indirect recursion):
      - S → NP VP
      - NP → ... N RelCl
      - RelCl → whX S

A family of recursions

holistic (e.g. lexical items, atoms)

flat string (e.g. single phrases, clauses)

Chomsky-Schützenberger Hierarchy of Formal Languages

3 iterative string – also flat, despite apparent branching:

head recursion, left-branching:

((((John's) father's) wife's) ring)

tail recursion, right-branching:

(this is the dog (that chased the cat (that killed the rat ...)))

2 nested strings:

(the car (whose owner (who you met) was drunk) crashed)

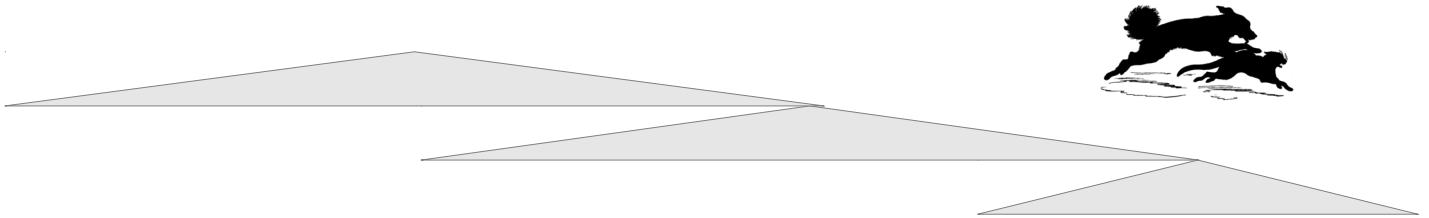
1 cross-linear nesting:

Denis, Bill and Bert married Sue, Molly and Charlene respectively.

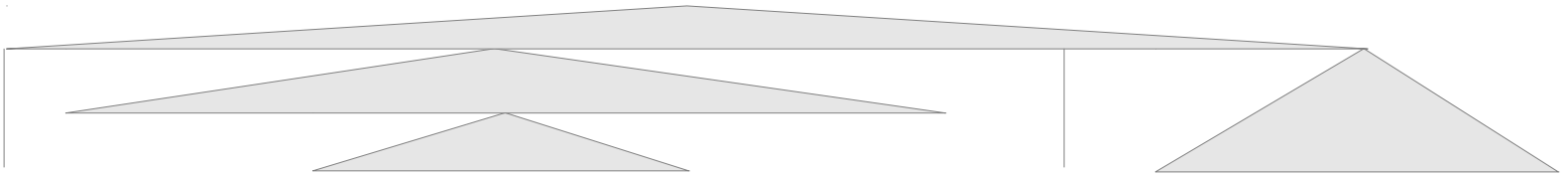
0 arbitrary nesting:

Bill came and – ouch! - I forgot – damn his cat – where was I, how come I stubbed my toe? – and anyway he did come.

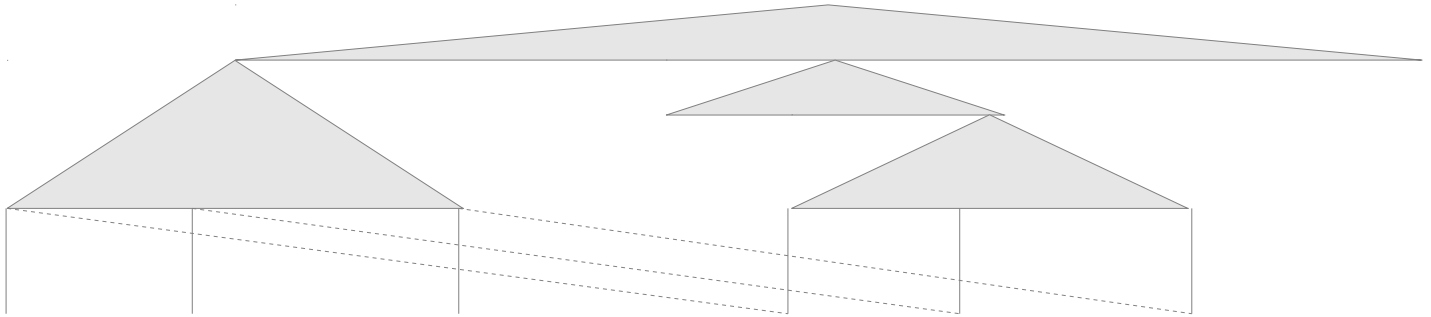
Some examples of recursive compositionality



This is the dog that chased the cat that ate the mouse ...
Right-branching (tail) recursion (≈ iteration)

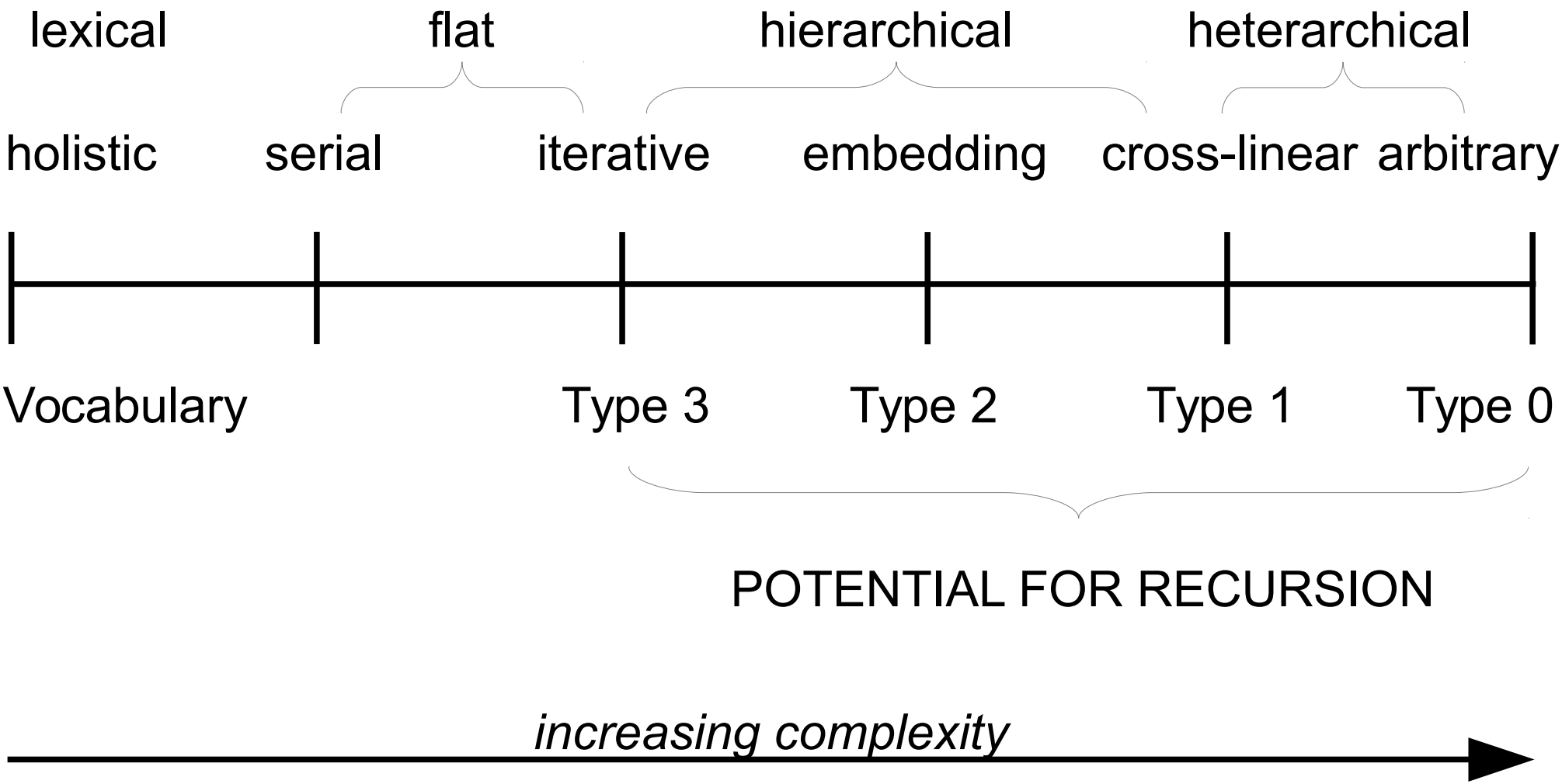


If the man who John met goes home then Jane will smile
Centre-embedding recursion.



June, Jane and Jean love Mick, Dick and Nick, respectively
Recursive cross-serial dependency.

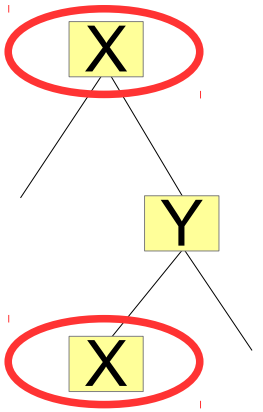
A well-defined scale of syntagmatic structures by complexity



The ‘recursion debate’

- As already noted, the term ‘recursion’ is a formal term,
  - from logic, arithmetic, algebra:
- Yet some linguists feel free to interpret it intuitively,

  for example, as a tree in which a symbol dominates an instance of the same symbol somewhere else in the tree
  but does not define a non-finite set of structures



- But this is not an adequate definition:
  - recursion must be defined in the grammar
  - not in the grammar output alone (the tree)
- If only trees are discussed, without reference to the grammar and its properties
  - there is a temptation use to the same symbol
  - for what should be treated as two different categories from an empirical point of view
  - and thus create ‘fake recursion’

The ‘recursion debate’

- Ambiguity of ‘recursion’:

  R1. Recursion in general definitions of infinite sets of structures which can be represented by rooted tree graphs
  - metatheoretical, applies to ‘life, the universe and everything’

  R2. Apparent recursion in strictly layered and other finite depth tree hierarchies:
  - as in the Prosodic Hierarchy of Selkirk and others

  R3. Iterative recursion in purely head-recursive (left-branching) or purely tail-recursive (right-branching) grammars:
  - as in very many linguistic descriptions

  R4. General context-free recursion over tree hierarchies, as permitted by general context-free grammars:
  - as postulated in generative linguistics and the recursion debate.

  R5. Cross-serial recursion in tree hierarchies with connections across the branches:
  - as in: *Peter and Paul married Jean and Joan, respectively*.

The ‘recursion debate’

- Strictly linear cases:

  R2. Apparent recursion in strictly layered and other finite depth tree hierarchies:
  - as in the Prosodic Hierarchy of Selkirk and others

  R3. Iterative recursion in purely head-recursive (left-branching) or purely tail-recursive (right-branching) grammars:
  - as in very many linguistic descriptions

- Non-linear cases:
  - R4. General context-free recursion over tree hierarchies, as permitted by general context-free grammars:
    - as postulated in generative linguistics and the recursion debate.

  R5. Cross-serial recursion in tree hierarchies with connections across the branches:
  - as in: *Peter and Paul married Jean and Joan, respectively*.

Multilinear Grammar: Summary of Theses

Multilinear Grammar: Summary of Theses

- Semantic and pragmatic patterns
  - are arbitrarily complex (hierarchical, heterarchical, arbitrary)
  - because this is how our cognition structures the world
  - no problem with any kind of recursion

Multilinear Grammar: Summary of Theses

- Prosodic at interpretation at each rank
  - Discourse: sequential or overlapping adjacency pairs/triples
  - Utterance: intonation, rhythm
  - Phrase: intonation, accent patterns
  - Word: foot, syllable, phoneme
  - Morpheme: stress, accent, tone

  is flat:

  - restricted to finite or linear patterns
  - clear example of finite structures: syllables (any language)
  - clear example of linear structures:
    - sequences of stressed syllables (English)

Multilinear Grammar: Summary of Theses

- Categorial cores at each rank are highly restricted to
  - a finite set of ranks
    - Discourse: dialogue interaction
    - Utterance: component of discourse; text
    - Sentence: clause, phrase
    - Word: (inflected), compound, (derived), simple
    - Morpheme
  - the No-Centre-Embedding Constraint:
    - no general centre-embedding recursion
    - only linear patterns:
      - finite sequences
      - left branching or right branching (not mixed left and right)
      - linear recursion, iteration

Multilinear Grammar: Summary of Theses

- There are two apparent kinds of exception to the No-Centre-Embedding Constraint:

1) if finite memory is supported by
   - rehearsal (learning by rote)
   - external media, in writing

   then non-embedded right-branching structures
   - such as relative clauses on final, rhematic nouns

   may be generalised with moderate success to embedded structures
   - to relative clauses on non-final, thematic nouns

2) arbitrary semantic structures, which are encoded linearly by
   - coordination, asyndeton, serial verb constructions
   - partly constrained, partly unconstrained anaphora

Multilinear Grammar / Ranks and Interpretations:

Four Challenges

Challenges for the Multilinear Grammar framework

1) Uniqueness:

   Show that the ranks in the architecture are *sui generis*, i.e. have their own distinctive properties.

2) Finiteness:

   Show that each rank has a fundamentally linear structure:

   - in the categorial core
   - in the modality interpretation (e.g. prosody, pitch patterns)

3) Linearity:

   1) Show that the linear structure may be represented
      - as paths through a finite state automaton
   - as right-branching graphs, derived from a right-linear grammar

4) Chunking:

   Show that hierarchical representations serve only

   - a parsing function, i.e. to represent groups in the linear sequence
   - and do not represent represent recursion in the general sense

Uniqueness

Uniqueness: the Ranks in Semantic Perspective

σ_{disc}:

- discourse framing, adjacency pairs, dialogue acts (speech acts in context), turn-taking, genres such as debate and conversation, participant role;

σ_{utt}:

- speech acts; information structure; argumentation, narration, poetry and other text types;

σ_{phrase}:

- propositional meaning, time, aspect, modality;

σ_{word}:

- names, predicates and compositional operators;

σ_{morph}:

- simple names and predicates;

Uniqueness: the Ranks in Categorial Perspective

τ_{disc}:

- adjacency pairs; turn alternation; discourse framing (introduction – body – termination); finite state models

τ_{utt}:

- sequences of sentences, phrases, clauses (connected semantically by anaphora and grounded in background context)

τ_{phrase}:

- phrases with finite length in the simple case; right-branching in the complex case; fragile centre-embedding by generalisation of right branching

τ_{word}:

- inflection (finite length); compound (V*); derivation (finite length); linear patterns

τ_{morph}:

- finite set of syllables, each with finite maximum length

Uniqueness: the Ranks in Prosodic Perspective

π_{disc}:

- greeting intonations (e.g. ‘call contours’)

π_{utt}:

- continuity and finality indicated by rising, falling, suspended intonations

π_{word}:

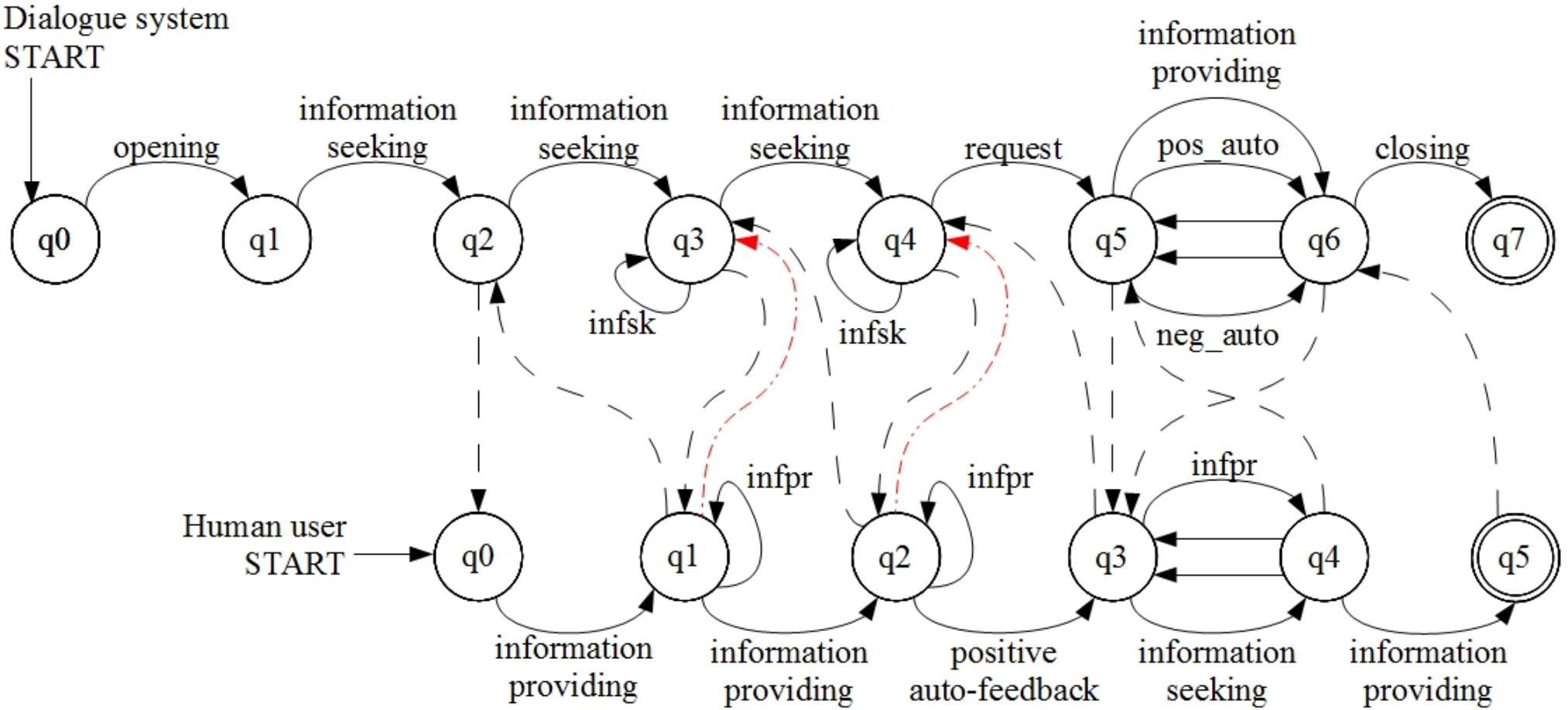
- morphological stress, pitch accent, tone

π_{morph}:

- phonological stress, pitch accent, tone

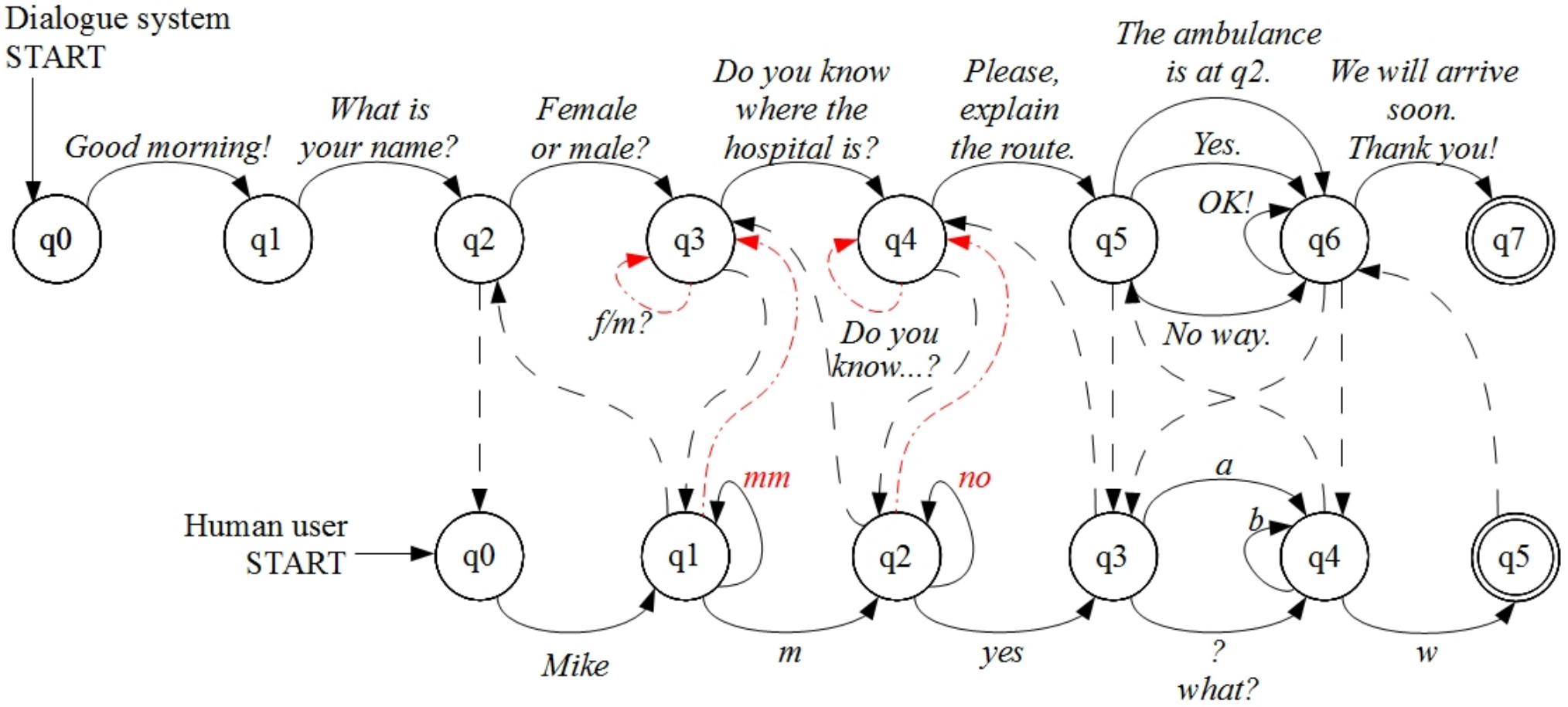
Finiteness, linearity: Dialogue (discourse grammar)

Dialogue: finite state transition diagramme



Dialogue grammar describing discourse between a caller and emergency services (Polish, Bachan 2013)

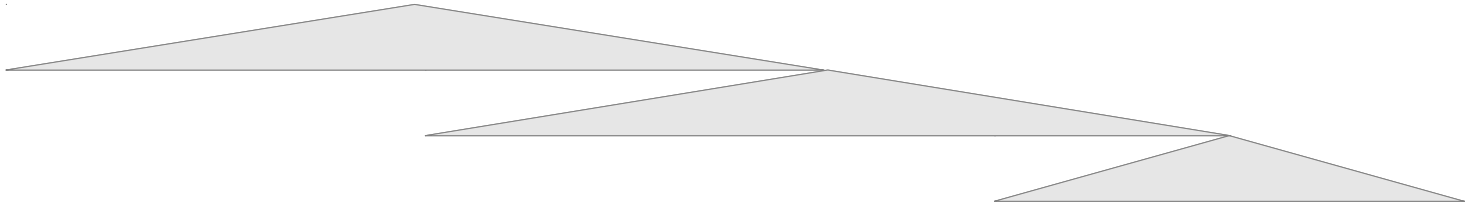
Dialogue: finite state transition diagramme



Dialogue example describing discourse between a caller and emergency services (Polish, Bachan 2013)

Finiteness, linearity: Sentences (‘syntax’)

What about serial patterns with recursion – i.e. iteration?



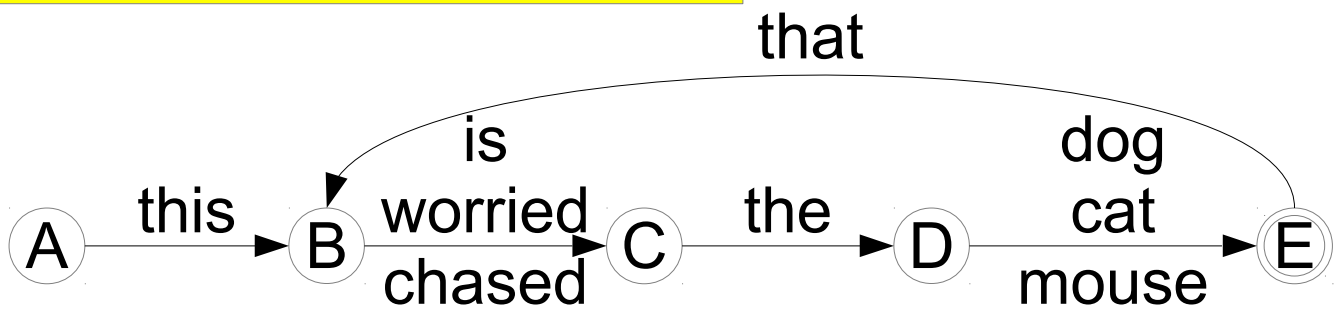
This is the dog that worried the cat that chased the mouse ...

Is this a right-branching tree generated by a right-branching grammar?

A → *this* B
B → {*is*, *worried*, *chased*, ...} C
C → *the* D

D → {*dog*, *cat*, *mouse*, ...} E
D → {*dog*, *cat*, *mouse*, ...}
E → *that* B

Or is it just a linear sequence?

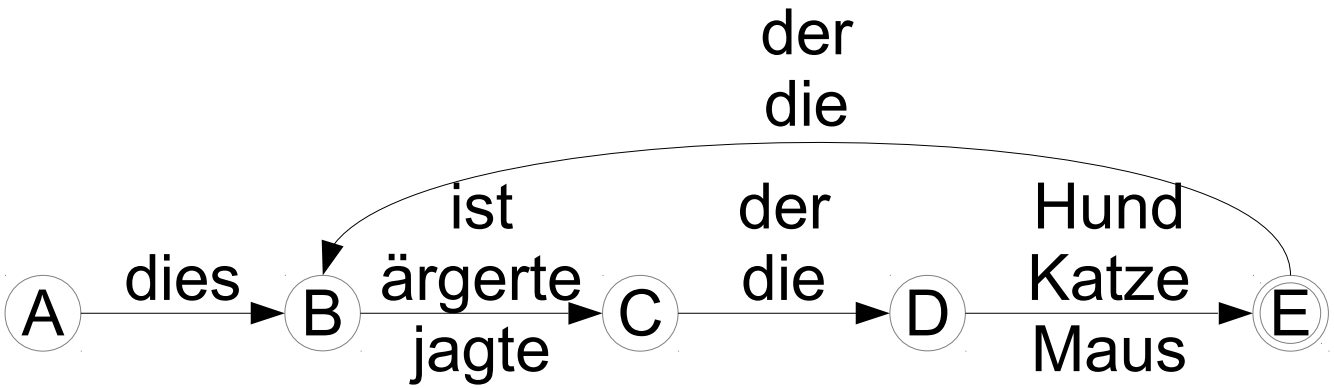


But what about seriality in languages like German?

In formal German grammar, centre-embedding contrasts with seriality in informal spoken German:

Dies ist der Hund, der die Katze, die die Maus ... jagte, ärgerte.

Dies ist der Hund, der die Katze ärgerte, die die Maus jagte, ...

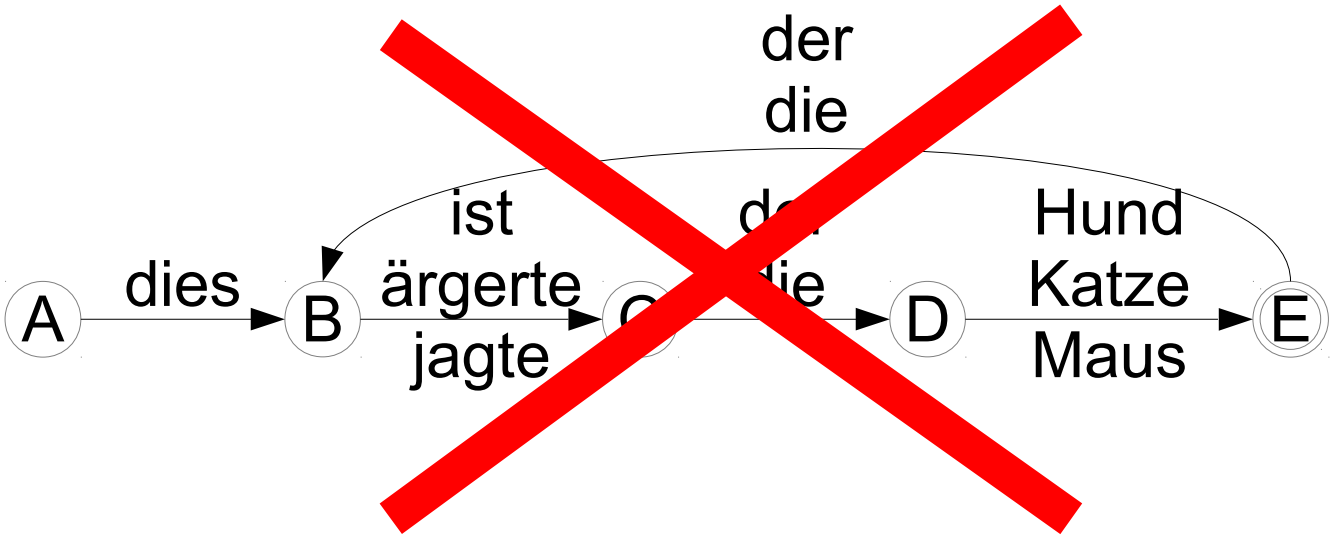


But what about seriality in languages like German?

In formal German grammar, centre-embedding contrasts with seriality in informal spoken German:

Dies ist der Hund, der die Katze, die die Maus ... jagte, ärgerte.

Dies ist der Hund, der die Katze ärgerte, die die Maus jagte, ...

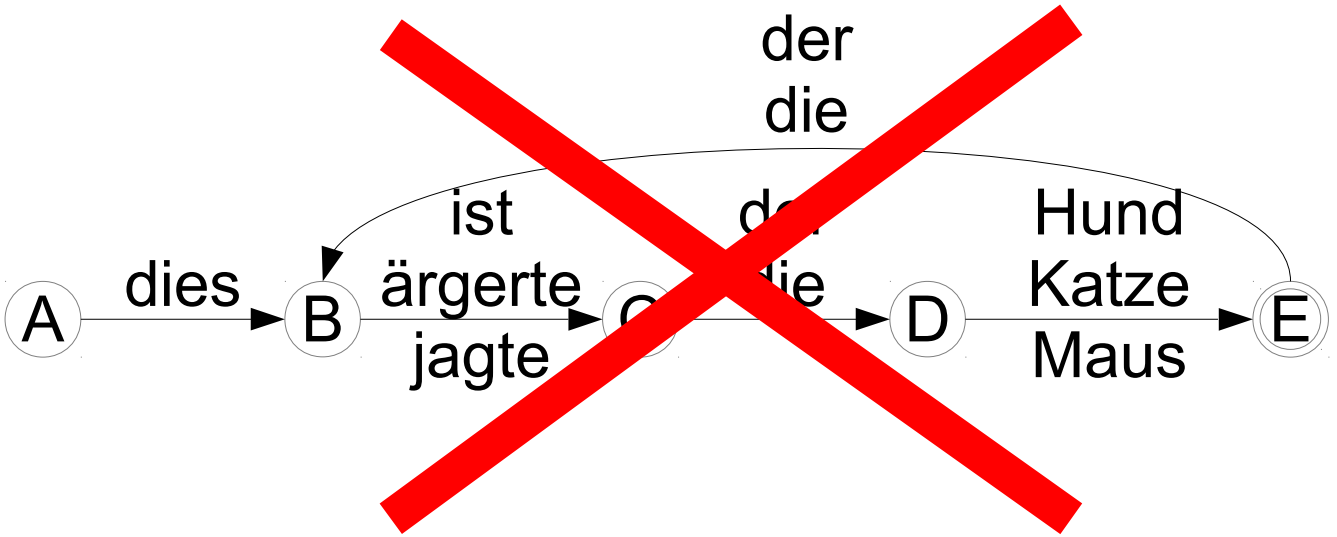


But what about seriality in languages like German?

In formal German grammar, centre-embedding contrasts with seriality in informal spoken German:

Dies ist der Hund, der die Katze, die die Maus ... jagte, ärgerte.

Dies ist der Hund, der die Katze ärgerte, die die Maus jagte, ...



So do we need context-sensitive rules for German?

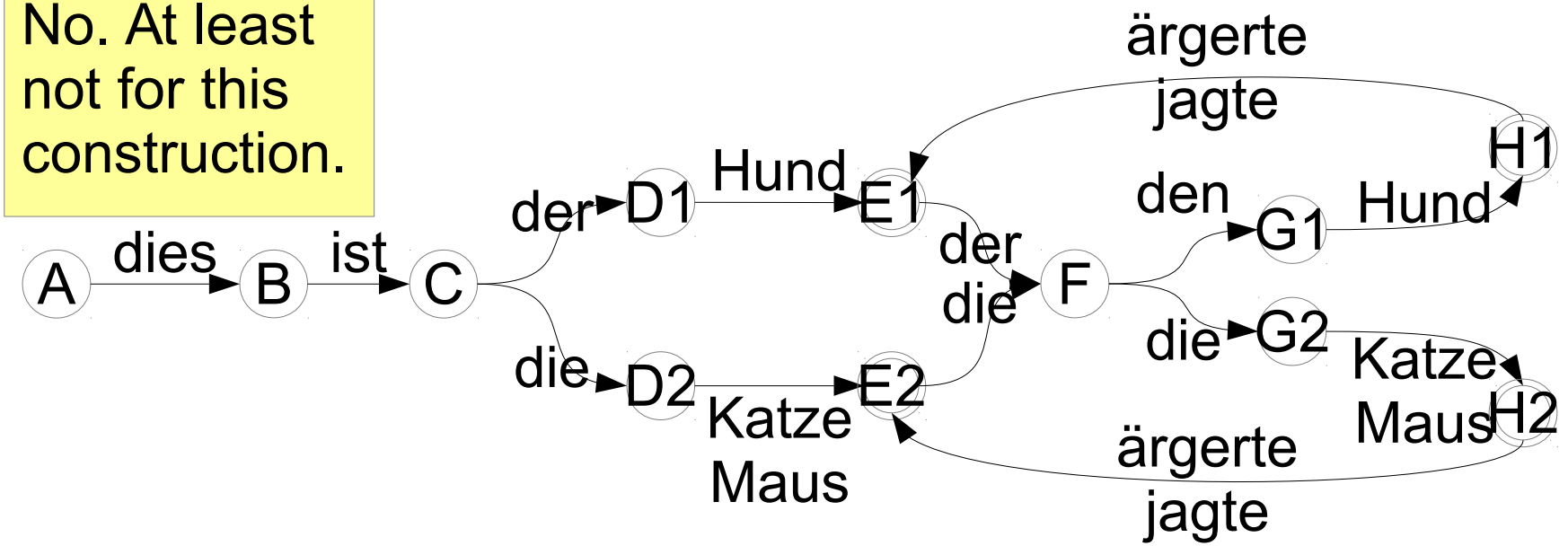
But what about seriality in languages like German?

In formal German grammar, centre-embedding contrasts with seriality in informal spoken German:

Dies ist der Hund, der die Katze, die die Maus ... jagte, ärgerte.

Dies ist der Hund, der die Katze ärgerte, die die Maus jagte, ...

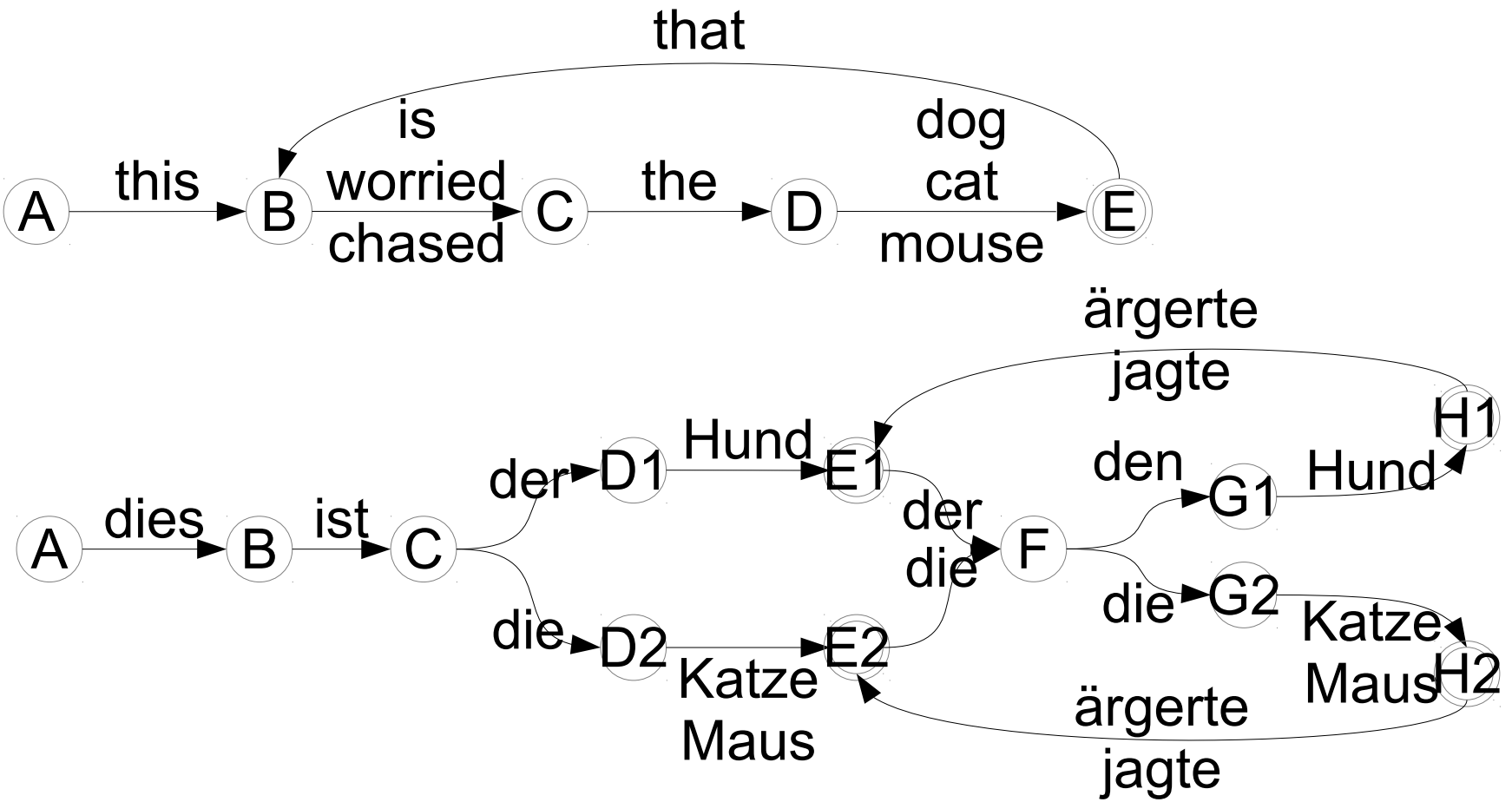
No. At least not for this construction.



Seriality is enough.

This aspect of English is simpler than German:

- just more of the same
- with the same serial rule types



Sometimes structure looks sort of non-serial

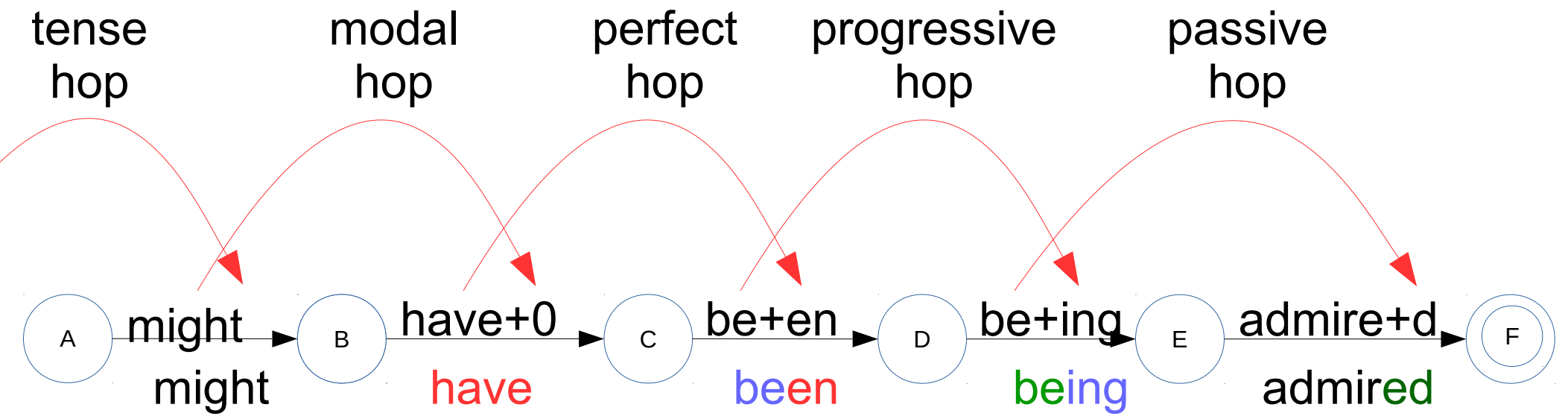
Chomsky's affix-hopping ('flipflop') transformation (1957:39):

Let *Af* stand for any of the affixes *past*, Ø, *en*, *ing*. Let *v* stand for any *M* or *V*, or *have* or *be* (i.e. for any non-affix in the phrase *Verb*). Then:

$Af + v \rightarrow v + Af\ \#$,

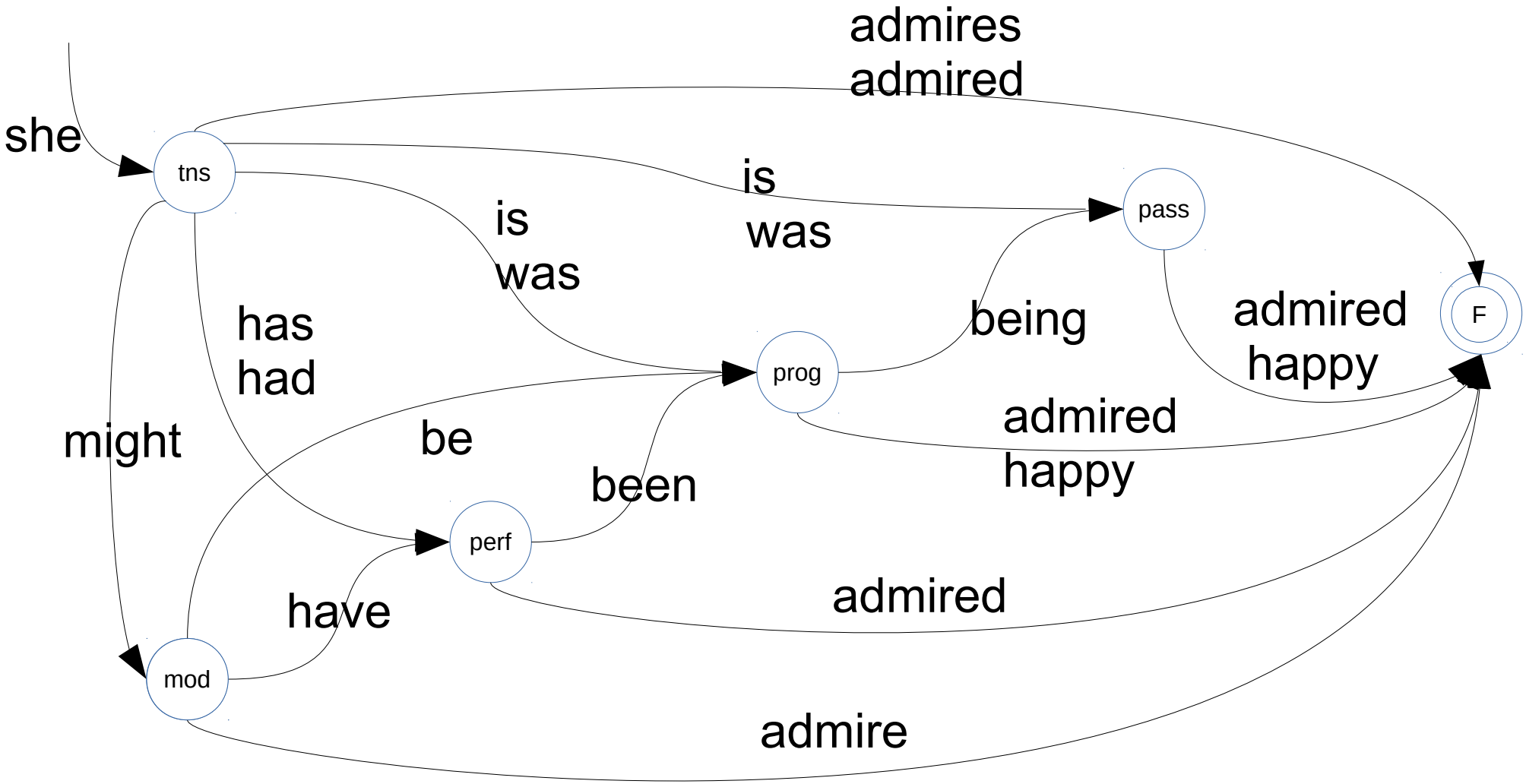
where # is interpreted as a word boundary.

modal or verb



So: do we need such transformations?

No transformation needed, seriality works ...



The ‘transformation’ turns out to be a notation for generalising linear structures, and is not strictly necessary for processing.

People *hate* recursion!

A free-text search for indices of nested recursion

marked by *wh*- items, in Sampson's CHRISTINE1 treebank of informal spoken English (abt. 14,000 words from the CHRISTINE database of 35,000 words)

revealed hardly any wh-recursions of any kind:

145 *who*/*whose* pronoun occurrences (*whom* did not occur)

- 129 sentence-initial interrogatives
- 16 relative *who*/*whose* clauses
  - 9 interrupted fragments (missing mandatory constituents
  - 7 were complete relative clauses, but none nested
- 1 (!) example of potential nesting

  which has an incomplete main clause and peters out incohesively

People *hate* recursion!

So what is going on with this potential nesting?

> *we found out that the neighbours on the left hand side who were in fact an elderly couple and his was erm and he had his own business working at home*

People *hate* recursion!

So what is going on with this potential nesting?

we found out that the neighbours on the left hand side who were in fact an elderly couple and his was erm and he had his own business working at home

main clause with object complement:

"we found out that ..."

subject of object complement:

"the neighbours on the left hand side ..."

rel. clause in subject:

"who were in fact an elderly couple"

People *hate* recursion!

The one example of attempted nesting is broken!

we found out

that the neighbours on the left hand side

who were in fact an elderly couple

and his was erm

and he had his own business working at home

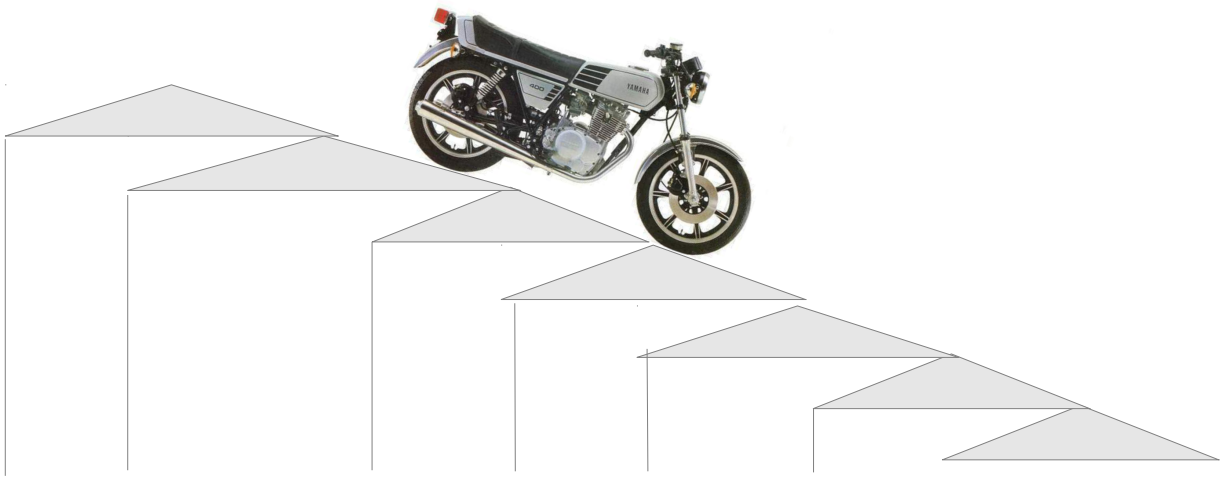
But… *where is the matching main verb?*

The speaker apparently regretted starting a nested relative clause, later ignoring the ‘who’ and reverting to coordination.

Finiteness, linearity: Words (‘morphology’)

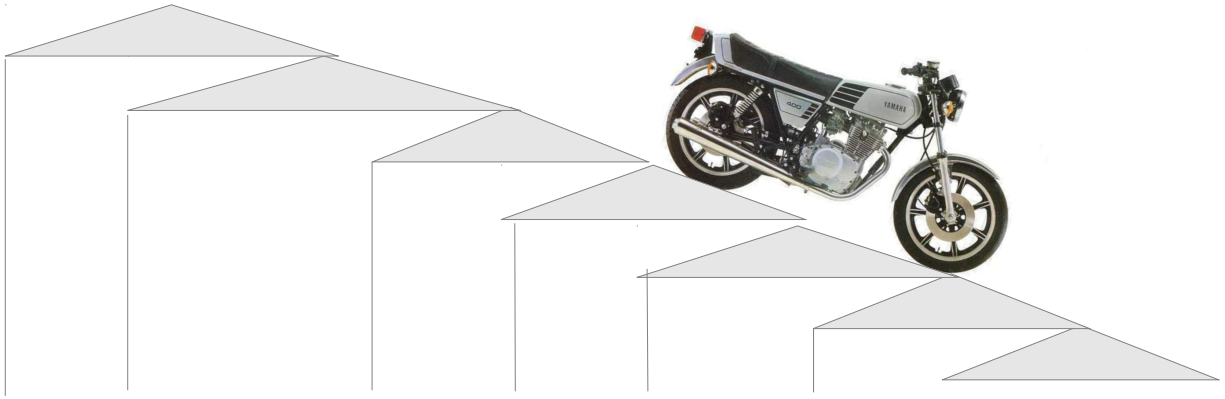
Mismatch: morphological seriality vs. semantic hierarchy

Morphological structure: serial (≈ right branching)



Mismatch: morphological seriality vs. semantic hierarchy

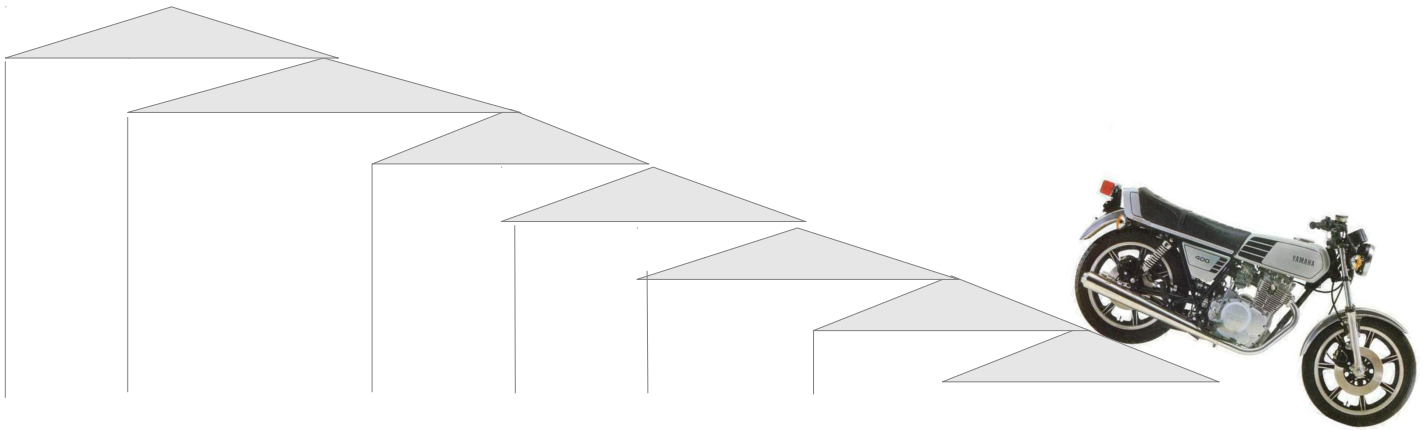
Morphological structure: serial (≈ right branching)



twin cylinder over head cam shaft motor bike

Mismatch: morphological seriality vs. semantic hierarchy

Morphological structure: serial (≈ right branching)

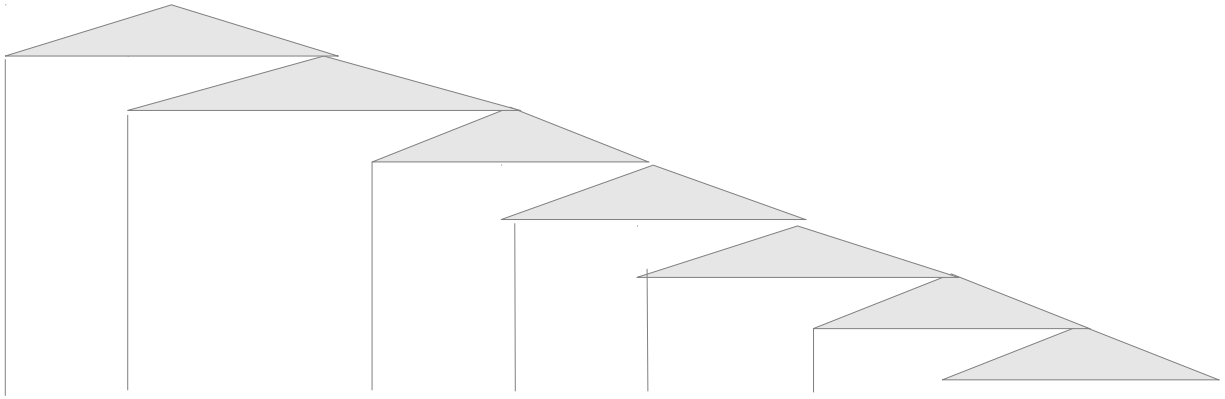


twin cylinder over head cam shaft motor bike

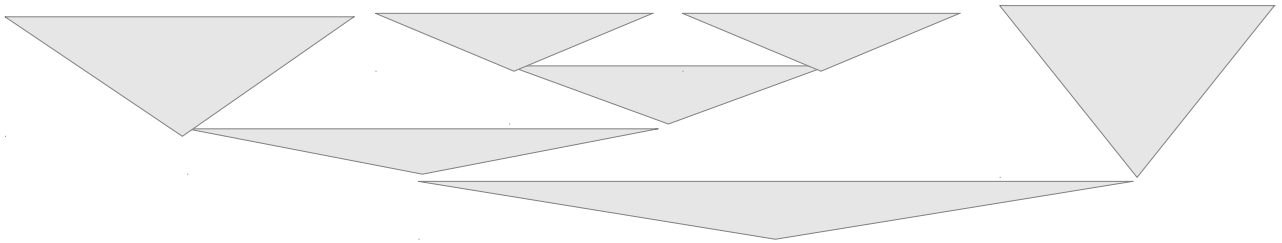
Semantic interpretation

Mismatch: morphological seriality vs. semantic hierarchy

Morphological structure: serial (≈ right branching)



twin cylinder over head cam shaft motor bike



SHOCK!!!

Semantic interpretation: *centre-embedding*

Finiteness, linearity: morphemes (‘phonology’)

Serial non-hierarchical structures in phonology

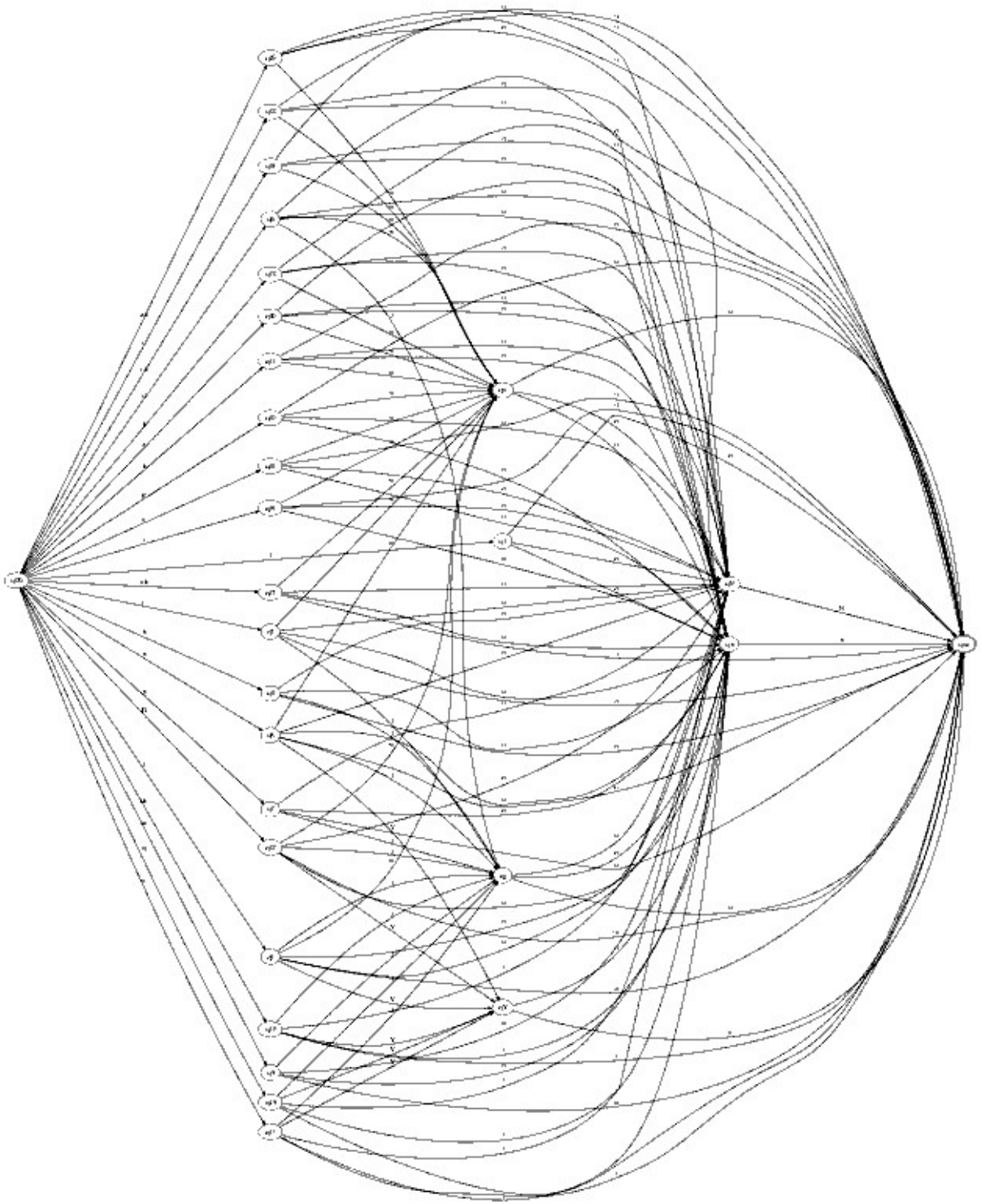
Mandarin syllable structure

Over 300 syllables

Traditional analyses only have two constituents
cf. the Pinyin table.

This analysis includes all optional positions and has four constituents:

Obstruent + Glide + Vowel/Diphthong + Nasal



Serial non-hierarchical structures in phonology

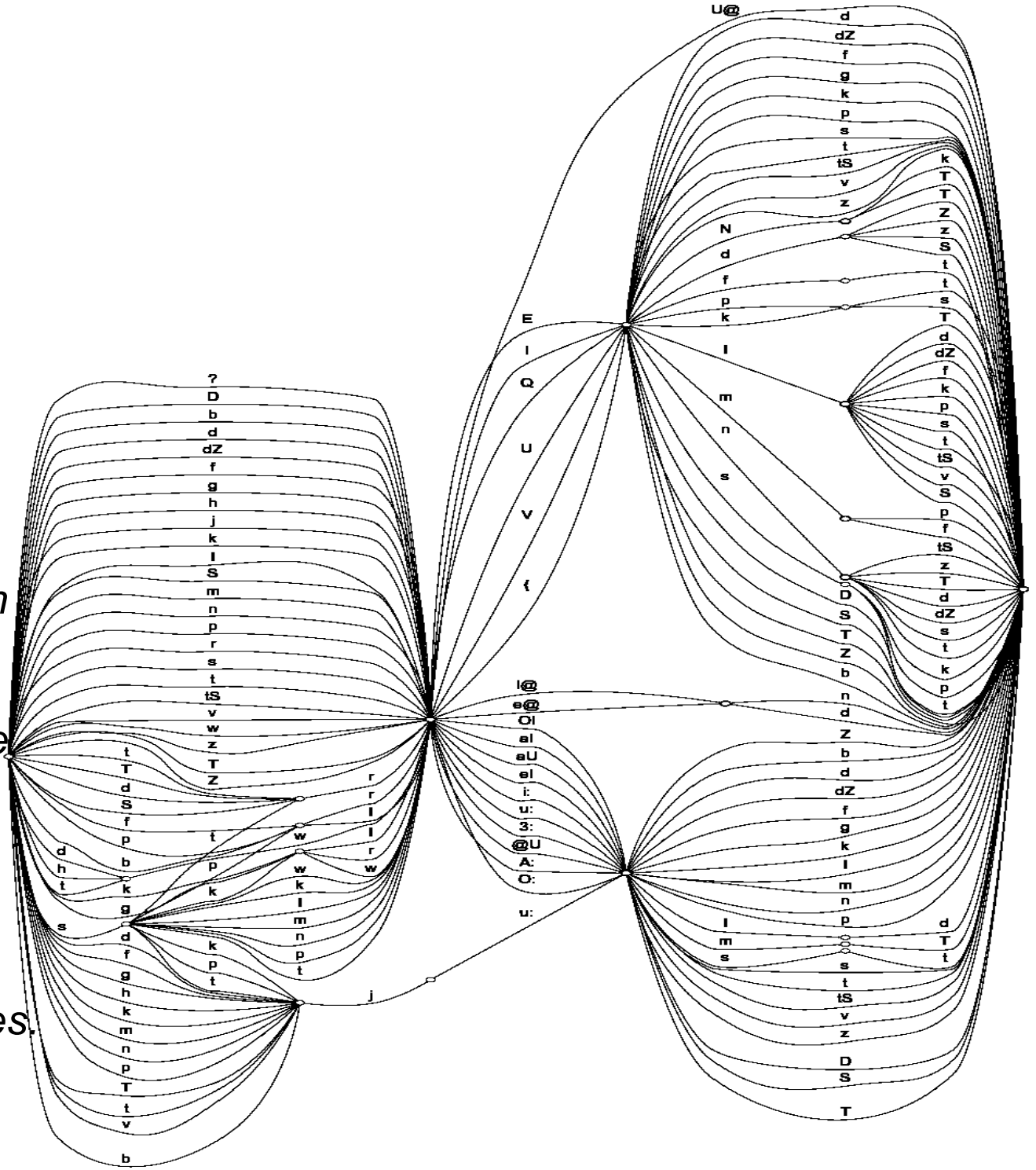
<u>English</u> syllable structure

Over 30000 potential syllables

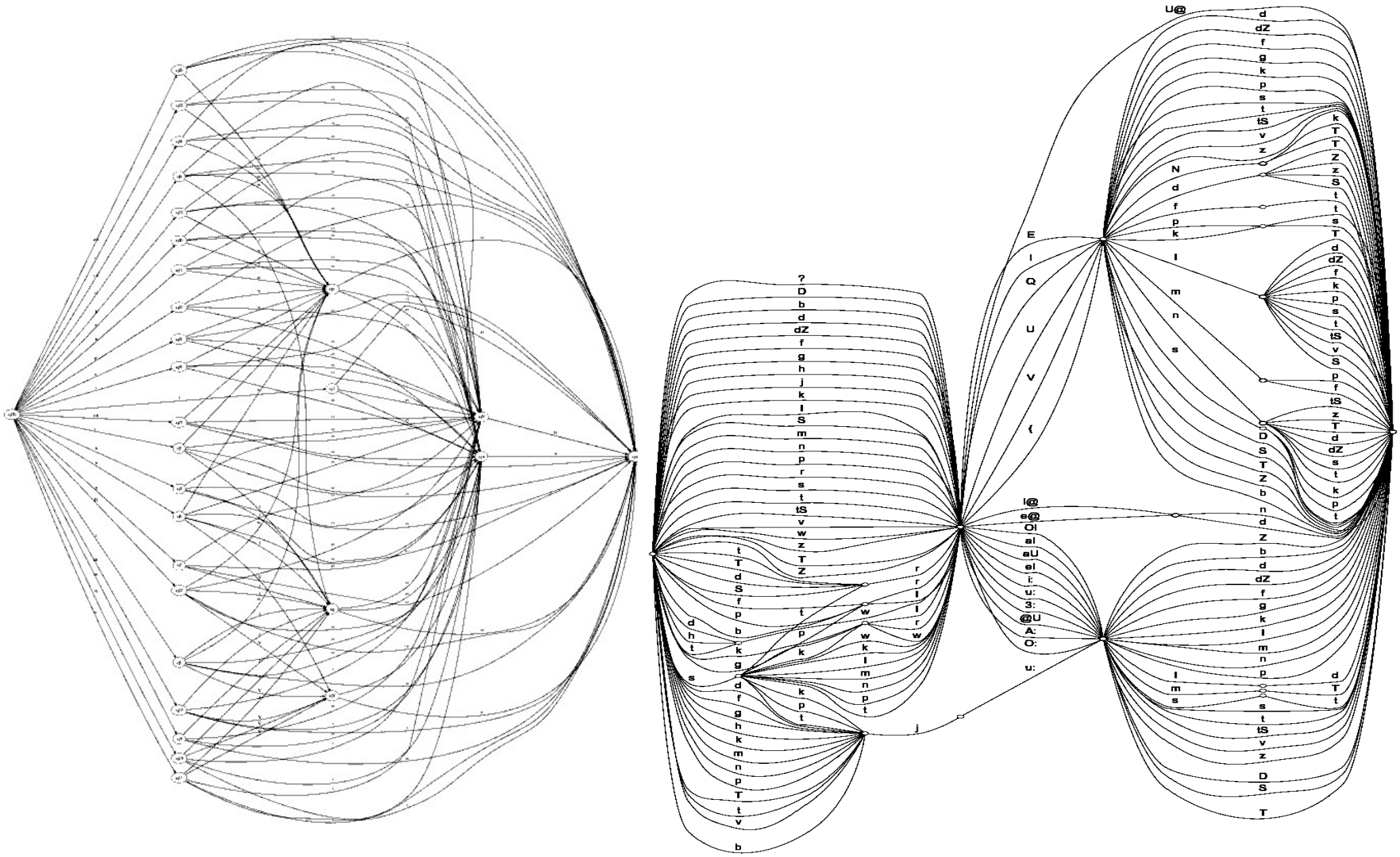
Traditional analyses (except Whorf) use very general categories (C, V etc.) often with trees.

These analyses fail to show the intricacy of structure of English syllables, however.

NB: This grammar shows potential syllables, not simply actual lexically attested syllables.

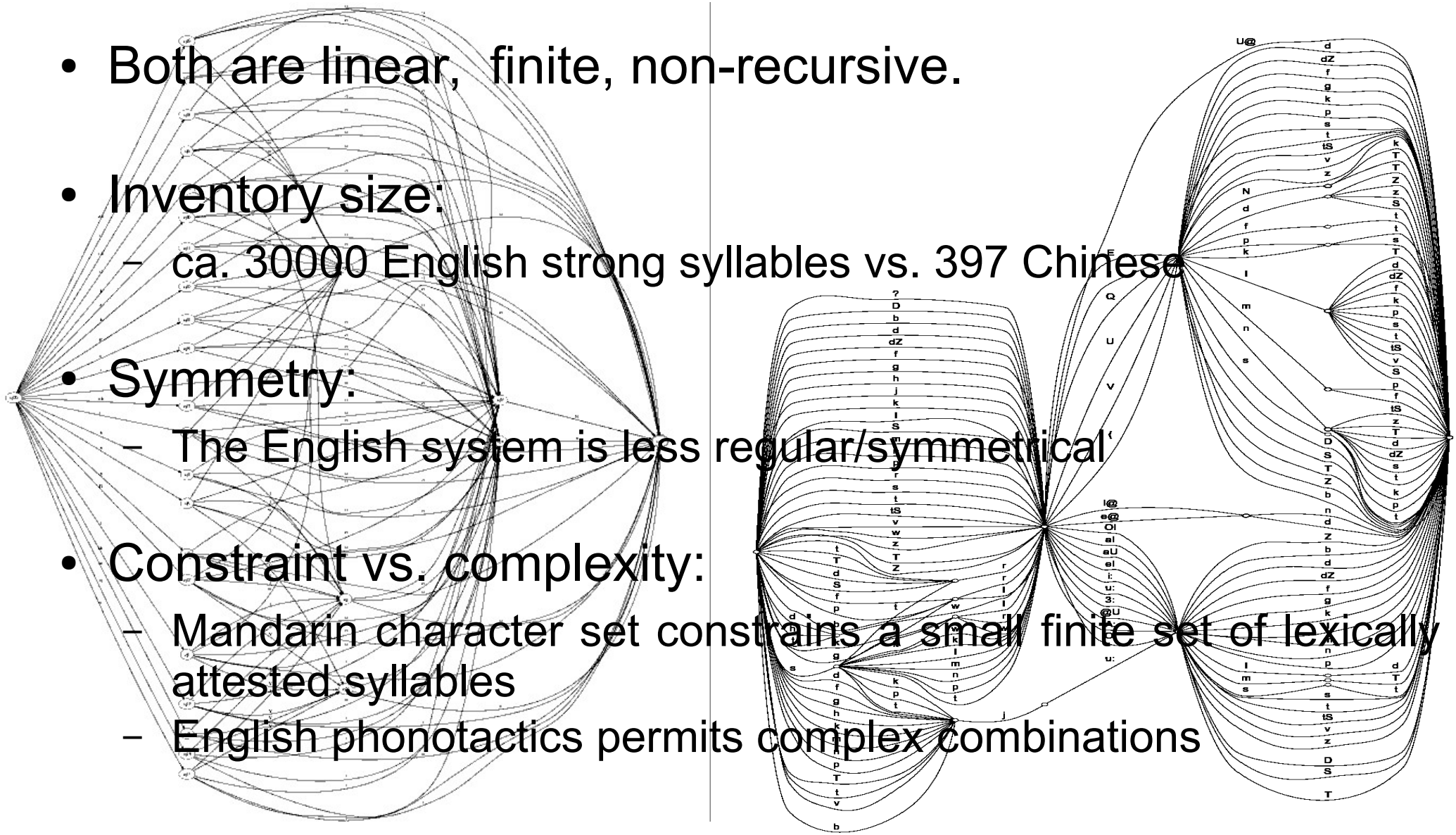


Which is more complicated, Mandarin or English syllable structure?



Which is more complicated, Mandarin or English syllable structure?

- Both are linear, finite, non-recursive.
- Inventory size:
  - ca. 30000 English strong syllables vs. 397 Chinese
- Symmetry:
  - The English system is less regular/symmetrical
- Constraint vs. complexity:
  - Mandarin character set constrains a small finite set of lexically attested syllables
  - English phonotactics permits complex combinations



“The human brain is a finite machine, albeit a complex one.”

Naively speaking:

The skull is a finite container which fully encloses the brain.

Finite machine determined by finite total memory:

1. processor registers
2. cache
3. RAM
4. swap
5. mass storage
6. archive

“The human brain is a finite machine, albeit a complex one.”

How could the brain be seen as non-finite, or at least as extendable?

Maybe

- by means of external storage media, e.g. writing
- by some so far unknown form of telepathic communication with other brains
- by reducing component size to and beyond atomic size
- any combination of these

“The human brain is a finite machine, albeit a complex one.”

Finite machine determined by finite total memory:

1. processor registers
2. cache
3. RAM
4. swap
5. mass storage
6. archive

Finite machine status - claimed by many, often in a conditional sentence, e.g. Gödel 1952: “if the human mind were equivalent to a finite machine then objective mathematics would not only be incompletable … “

But note - ‘the human mind’ not the human brain

How about the human mind?

No idea.

"Idealizing away death and other contingent resource constraints ..."

Copeland, B. Jack and Oron Shagrir. 2013. Turing vs. Gödel on Computability and the Mind. In: Copeland, B. Jack Posy, Carl J. Shagrir, Oron, eds. *Computability. Turing, Gödel, Church, and Beyond*. Boston MA: MIT Press.

“The human brain is a finite machine, albeit a complex one.”

So this applies to the whole of cognition, not just language or languages.

The human brain may – or may not – embody recursively defined grammars.

The human brain may – or may not – embody recursive algorithms for processing these grammars.

But

- memory size is finite – though huge – even if ‘it’s turtles all the way down’ to quantum size
- attention spans are finite

Comments and Conclusions

Comments and Conclusions on Recursion

Hauser & al. (2002) Fitch & al. (2005) claimed:

recursion is the only necessary property of natural languages.

This claim has been taken up by many linguists.

In fact the claim is not new.

It echoes Hockett's (1960) 'productivity', one his 15 'design features' of language: all of these are necessary, none is sufficent.

The claim is actually quite fuzzy, because

- only very informal characterizations of 'recursion' are given
- there are many recursion types, with processing differences
- other species besides humans use behavioural recursion

Comments and Conclusions on Recursion

One much discussed critique:

- no recursion has been found in languages like Pirahã (Everett 2005).

- But is this critique valid?
  - logic: not finding something doesn't mean it doesn't exist
    but if you look hard you may tentatively assume so (Karl Popper)
  - functionality: languages are aggregates of registers
    many restricted registers have no recursion (IKEA instructions, ...)
  - choice: people don't like recursion and use it in writing if at all
    so clearly if you don't write you may curse but you don't recurse :)
  - counterexample: claims that there is recursion in Pirahã
    - but the known examples show only 'flat recursion' i.e. iteration

Summary and General Conclusion

I have aimed

- to demythologize the issues concerned
- by
  - looking at different kinds of recursion in languages,
  - reducing apparent nesting in as many cases as possible to simple iteration,
  - noting that people hate recursion (except in writing)

    and are not very good at it!

A reminder:

One condition – recursion – not enough

Hockett had fifteen necessary conditions in the final version of his 'design features' framework, not just one!

Multilinear Grammar / Ranks and Interpretations:

The framework provides not only design features but a coherent architecture

Finally

Look at the facts...

→ *People **hate** recursion*
and avoid it except in writing!

→ *People **love** iteration*
and even repetition!

Processes – with finite memory and finite time – are as important as structures

- Apply Occam's Razor...
  - → *Go for the least complex solution!*

Finally

Look at the facts...

→ *People **hate** recursion*
and avoid it except in writing!

→ *People **love** iteration*
and even repetition!

Processes – with finite memory and finite time – are as important as structures

- Apply Occam's Razor...
  - → *Go for the least complex solution!*

And avoid turtles ...