Formal Language Theory

Gerhard Jäger

University of Tübingen

Workshop *Artificial Grammar Learning and Formal Language Theory*
Nijmegen, November 23, 2010

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Formal Language Theory

- **Formal Language**:
  - set of *strings* over a finite vocabulary
  - finite or infinite
- **Formal Language Theory**: collection of mathematical/algorithmic tools about
  - defining FL (with finite means)
  - processing FL (recognizing, parsing, translating)
- FLT is **not** about
  - semantics of FLs
  - statistical properties of FLs
- initiated by Chomsky in the 1950s to motivate generative grammar
- important role in formal linguistics and theoretical computer science
- recent new domain of application in bio-informatics

The Chomsky Hierarchy

- **Formal Grammar:** finite specification of a formal language
- Chomsky defined general format for FGs: *string rewriting systems*
- A **String Rewriting System** essentially consists of
  - a set of *rewrite rules*

    $$\alpha \rightarrow \beta$$

    (α and β are strings of symbols)
  - a designated *start symbol* S
- A **derivation** starts with S and applies rewrite rules to sub-strings until no further rules can be applied
- **language** defined by a grammar: set of strings that can be derived this way[1]

[1] I am skipping over the (at this point) inessential distinction between non-terminal and terminal symbols.

The Chomsky Hierarchy

- format of *String Rewriting Systems* is very general
- every (formal) language that can be defined algorithmically can be defined by a FG in this sense
- Chomsky Hierarchy:
  - hierarchy of ever more restricted versions of FGs
  - defines a hierarchy of formal languages
    1. Type 0: recursively enumerable
    2. Type 1: context-sensitive
    3. Type 2: context-free (phrase structure)
    4. Type 3: regular (finite state)

The Chomsky Hierarchy

Type-0 grammars and recursively enumerable languages

- no restrictions on general format of rewrite rules
- equivalent to Turing Machine
- describes all languages that can be defined algorithmically

Examples

- Peano arithmetics
- set of all numbers that are the sum of two primes
- set of first order theorems
- set of equivalent pairs of regular expressions with exponentiation (decidable but not context-sensitive)

The Chomsky Hierarchy

Context-sensitive grammars and languages

- restriction of format of rewrite rules:

  Rules are non-shrinking.
- $\alpha \to \beta$: *length*$(\alpha) \leq$*length*(β)
- ensures decidability
- membership problem in worst case is **PSPACE hard**

Examples

- set of all primes
- set of all square numbers
- copy language
- $a^n b^m c^n d^m$
- triple-copy language ($\{w^3 | w \in \Sigma^*\}$)
- $a^n b^n c^n$
- $a^n b^n c^n d^n e^n$

The Chomsky Hierarchy

Context-free grammars and languages

- further restriction of rule format:

  Left hand side contains exactly one symbol.

- $A \rightarrow \alpha$
- membership problem decidable in **cubic time**.

Examples

- mirror language
- $a^n b^n$
- $a^n b^m c^m d^n$
- well-formed parentheses
- algebraic expression

The Chomsky Hierarchy

Regular grammars and languages

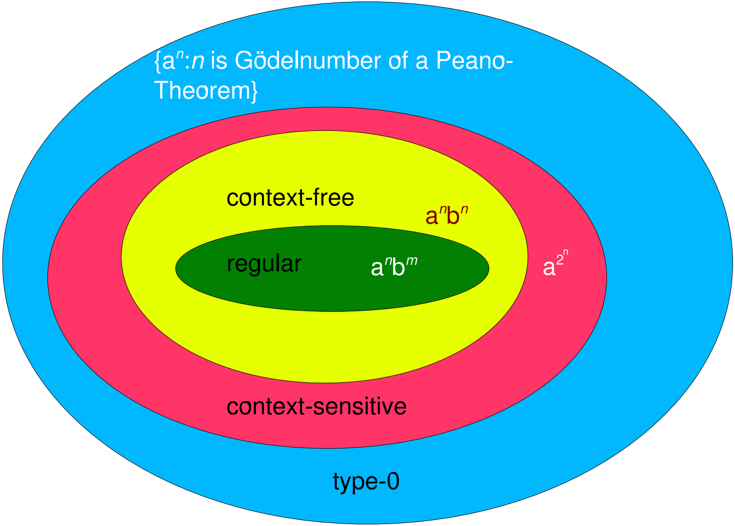
- further restriction of rule format:

  Right-hand side contains at most one non-terminal symbol, preceding all terminal symbols.
- Terminal symbols: symbols that never occur at the RHS of a rule.
- $A \to (B)\alpha$, α a string of terminal symbols
- membership problem decidable in **linear time**.

Examples

- $a^n b^m$
- set of multiples of 4
- set of natural numbers that leave a remainder of 3 when divided by 4

The Chomsky Hierarchy



NL and the Chomsky Hierarchy

Where are natural languages located?

- hotly contested issue over several decades
- typical argument:
  - find a recursive construction C in a natural language L
  - argue that the **competence** of speakers admits unlimited recursion (while the performance certainly poses an upper limit)
  - reduce C to a formal language L' of known complexity via **homomorphisms**
  - make a case that L must be at least as complex as L'
  - extrapolate to all human languages: if there is one language which is at least as complex as ..., then the human language faculty must allow it in general

NL and the Chomsky Hierarchy

- Chomsky 1957: English is not regular.
- The following constructions can be arbitrarily embedded into each other:
  - If S_1, then S_2.
  - Either S_3 or S_4.
  - The man that said that S_5 is arriving today.
- Therefore—Chomsky says—English cannot be regular.

"It is clear, then that in English we can find a sequence a + S1 + b, where there is a dependency between a and b, and we can select as S1 another sequence c + S2 + d, where there is a dependency between c and d ... etc. A set of sentences that is constructed in this way...will have all of the mirror image properties of [the mirror language] which exclude [the mirror language] from the set of finite state languages."
(Chomsky 1957)

NL and the Chomsky Hierarchy

- Chomsky 1957: English is not regular.
- The following constructions can be arbitrarily embedded into each other:
  - If S_1, then S_2.
  - Either S_3 or S_4.
  - The man that said that S_5 is arriving today.
- Therefore—Chomsky says—English cannot be regular.

"It is clear, then that in English we can find a sequence a + S1 + b, where there is a dependency between a and b, and we can select as S1 another sequence c + S2 + d, where there is a dependency between c and d ... etc. A set of sentences that is constructed in this way...will have all of the mirror image properties of [the mirror language] which exclude [the mirror language] from the set of finite state languages."
(Chomsky 1957) Skip technical stuff

Closure properties of regular languages

Theorem 1: If L_1 and L_2 are regular languages, then $L_1 \cap L_2$ is also a regular language.

Theorem 2: The class of regular languages is closed under homomorphism.

Theorem 3: The class of regular languages is closed under inversion.

NL and the Chomsky Hierarchy

- argument is formally questionable because $either$ may occur without or, or without $either$, if without $then$ and $then$ without if
- logic of the argument is correct though; can be made formally water-tight with e.g. *neither-nor* constructions

Neither did John claim that he neither smokes while . . . nor snores, nor did anybody believe it.

- English has (in principle) unlimited number of nested dependencies of unbounded length

NL and the Chomsky Hierarchy

- argument is formally questionable because $either$ may occur without or, or without $either$, if without $then$ and $then$ without if
- logic of the argument is correct though; can be made formally water-tight with e.g. *neither-nor* constructions

Neither did John claim that he neither smokes while . . . nor snores, nor did anybody believe it.

- English has (in principle) unlimited number of nested dependencies of unbounded length

NL and the Chomsky Hierarchy

- argument is formally questionable because $either$ may occur without or, or without $either$, if without $then$ and $then$ without if
- logic of the argument is correct though; can be made formally water-tight with e.g. *neither-nor* constructions

Neither did John claim that he neither smokes while . . . nor snores, nor did anybody believe it.

- English has (in principle) unlimited number of nested dependencies of unbounded length

NL and the Chomsky Hierarchy

- argument is formally questionable because $either$ may occur without or, or without $either$, if without $then$ and $then$ without if
- logic of the argument is correct though; can be made formally water-tight with e.g. *neither-nor* constructions

Neither did John claim that he neither smokes while . . . nor snores, nor did anybody believe it.

- English has (in principle) unlimited number of nested dependencies of unbounded length

NL and the Chomsky Hierarchy

- argument is formally questionable because $either$ may occur without or, or without $either$, if without $then$ and $then$ without if
- logic of the argument is correct though; can be made formally water-tight with e.g. *neither-nor* constructions

Neither did John claim that he neither smokes while . . . nor snores, nor did anybody believe it.

- English has (in principle) **unlimited** number of nested dependencies of unbounded length

NL and the Chomsky Hierarchy

- argument is formally questionable because *either* may occur without *or*, *or* without *either*, *if* without *then* and *then* without *if*
- logic of the argument is correct though; can be made formally water-tight with e.g. *neither-nor* constructions

Neither did John claim that he neither smokes while . . . nor snores, nor did anybody believe it.

- English has (in principle) unlimited number of **nested dependencies** of unbounded length

NL and the Chomsky Hierarchy

- argument is formally questionable because $either$ may occur without or, or without $either$, if without $then$ and $then$ without if
- logic of the argument is correct though; can be made formally water-tight with e.g. *neither-nor* constructions

Neither did John claim that he neither smokes while ... nor snores, nor did anybody believe it.

- English has (in principle) unlimited number of nested dependencies of **unbounded length**

NL and the Chomsky Hierarchy

- homomorphism:

$$\text{neither} \mapsto a$$

$$\text{nor} \mapsto b$$

$$\textit{everything else} \mapsto \varepsilon$$

If it neither rains nor snows, then if it rains then it snows.
$\mapsto ab$

NL and the Chomsky Hierarchy

- maps English not to the mirror language, but to the language L_1:

$$\begin{array}{lcl} S & \rightarrow & aST \\ T & \rightarrow & bST \\ T & \rightarrow & bS \\ S & \rightarrow & \varepsilon \end{array}$$

- This is the language over $\{a, b\}$ where each a is followed by a number of bs.

NL and the Chomsky Hierarchy

- maps English not to the mirror language, but to the language L_1:

$$\begin{array}{rcl} S & \rightarrow & aST \\ T & \rightarrow & bST \\ T & \rightarrow & bS \\ S & \rightarrow & \varepsilon \end{array}$$

- This is the language over $\{a, b\}$ where each a is followed by a number of bs.

▸ Skip technical stuff

The pumping lemma for regular languages

Let L be a regular language. Then there is a constant n such that if z is any string in L, and *length*$(z) \geq n$, we may write $z = uvw$ in such a way that *length*$(uv) \leq n$, $v \neq \varepsilon$, and for all $i \geq 0, uv^iw \in L$.

NL and the Chomsky Hierarchy

- Suppose English is regular.
- Due to closure under homomorphism, L_1 is regular.
- a^*b^* is a regular language.
- Thus $a^*b^* \cap L_1$ is a regular language

  $$L_2 = L_1 \cap a^*b^* = \{a^n b^m | n \leq m\}$$

  due to Theorem 1
- Due to closure under inversion and homomorphism,

  $$L_3 = \{a^n b^m | n \geq m\}$$

  is also regular.
- Hence L_4 is regular:

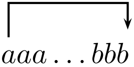
  $$L_4 = L_2 \cap L_3 = a^n b^n$$

NL and the Chomsky Hierarchy

- If English is regular, L_1 is regular.
- If L_1 is regular, $a^n b^n$ is regular. (This is the technical stuff.)
- $a^n b^n$ is not regular

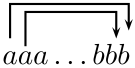
$$aaa \ldots bbb$$

- If English is regular, L_1 is regular.
- If L_1 is regular, $a^n b^n$ is regular. (This is the technical stuff.)
- $a^n b^n$ is not regular



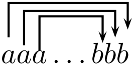
NL and the Chomsky Hierarchy

- If English is regular, L_1 is regular.
- If L_1 is regular, $a^n b^n$ is regular. (This is the technical stuff.)
- $a^n b^n$ is not regular



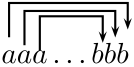
NL and the Chomsky Hierarchy

- If English is regular, L_1 is regular.
- If L_1 is regular, $a^n b^n$ is regular. (This is the technical stuff.)
- $a^n b^n$ is not regular



NL and the Chomsky Hierarchy

- If English is regular, L_1 is regular.
- If L_1 is regular, $a^n b^n$ is regular. (This is the technical stuff.)
- $a^n b^n$ is not regular



- Therefore English cannot be a regular language.

NL and the Chomsky Hierarchy

Dissenting view:

- *all arguments to this effect use center-embedding*
- *humans are extremely bad at processing center-embedding*
- *notion of competence that ignores this is dubious*
- *natural languages are regular after all*

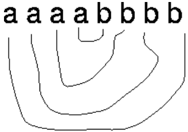
Are natural languages context-free?

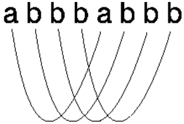
- history of the problem:
  - Chomsky 1957: conjecture that natural languages are not cf
  - sixties, seventies: many attempts to prove this conjecture
  - Pullum and Gazdar 1982:
    - all these attempts have failed
    - for all we know, natural languages (conceived as string sets) might be context-free
  - Huybregts 1984, Shieber 1985: proof that Swiss German is not context-free
  - Culy 1985: proof that Bambara is not context-free

NL and the Chomsky Hierarchy

Nested and crossing dependencies

- CFLs—unlike regular languages—can have unbounded dependencies
- however, these dependencies can only be **nested**, not **crossing**
- example:
  - $a^n b^n$ has unlimited nested dependencies → context-free
  - the copy language has unlimited crossing dependencies → not context-free





NL and the Chomsky Hierarchy

The respectively argument

- Bar-Hillel and Shamir (1960):
  - English contains copy-language
  - cannot be context-free
- Consider the sentence

John, Mary, David, ... are a widower, a widow, a widower, ..., respectively.

- Claim: the sentence is only grammatical under the condition that if the nth name is male (female) then the nth phrase after the copula is *a widower (a widow)*

NL and the Chomsky Hierarchy

The respectively argument

- Bar-Hillel and Shamir (1960):
    - English contains copy-language
    - cannot be context-free
- Consider the sentence

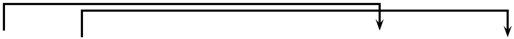
John, Mary, David, ... are a widower, a widow, a widower, ..., respectively.

- Claim: the sentence is only grammatical under the condition that if the nth name is male (female) then the nth phrase after the copula is *a widower (a widow)*

NL and the Chomsky Hierarchy

The respectively argument

- Bar-Hillel and Shamir (1960):
  - English contains copy-language
  - cannot be context-free
- Consider the sentence



John, Mary, David, ... are a widower, a widow, a widower, ..., respectively.

- Claim: the sentence is only grammatical under the condition that if the nth name is male (female) then the nth phrase after the copula is *a widower (a widow)*

The respectively argument

- Bar-Hillel and Shamir (1960):
  - English contains copy-language
  - cannot be context-free
- Consider the sentence

John, Mary, David, ... are a widower, a widow, a widower, ..., respectively.

- Claim: the sentence is only grammatical under the condition that if the nth name is male (female) then the nth phrase after the copula is *a widower (a widow)*

NL and the Chomsky Hierarchy

The respectively argument

- dependency structure of the copy language
- formal argument:
    - If English is cf, then the copy language is cf.
    - Copy language is not cf.
    - Hence English is not cf.

NL and the Chomsky Hierarchy

Counterargument

- crossing dependencies triggered by *respectively* are semantic rather than syntactic
- compare above example to

  (Here are John, Mary and David.) They are a widower, a widow and a widower, respectively.

Cross-serial dependencies in Dutch

- Huybregt (1976):
  - Dutch has copy-language like structures
  - thus Dutch is not context-free

(1) dat Jan Marie Pieter Arabisch laat zien schrijven
THAT JAN MARIE PIETER ARABIC LET SEE WRITE
'that Jan let Marie see Pieter write Arabic'

NL and the Chomsky Hierarchy

Cross-serial dependencies in Dutch

- Huybregt (1976):
  - Dutch has copy-language like structures
  - thus Dutch is not context-free

(1) dat Jan Marie Pieter Arabisch laat zien schrijven
THAT JAN MARIE PIETER ARABIC LET SEE WRITE
'that Jan let Marie see Pieter write Arabic'

NL and the Chomsky Hierarchy

Cross-serial dependencies in Dutch

- Huybregt (1976):
  - Dutch has copy-language like structures
  - thus Dutch is not context-free

(1) dat Jan Marie Pieter Arabisch laat zien schrijven
THAT JAN MARIE PIETER ARABIC LET SEE WRITE
'that Jan let Marie see Pieter write Arabic'

Cross-serial dependencies in Dutch

- Huybregt (1976):
  - Dutch has copy-language like structures
  - thus Dutch is not context-free

(1) dat Jan Marie Pieter Arabisch laat zien schrijven
that Jan Marie Pieter Arabic let see write
'that Jan let Marie see Pieter write Arabic'

Proof of non-context freeness

German

dass der Karl die Maria
dem Peter den Hans schwimmen lehren helfen lässt
'that Karl lets Maria help Peter to teach Hans how to swim'

Proof of non-context freeness

German

dass der Karl die Maria
dem Peter den Hans schwimmen lehren helfen lässt
'that Karl lets Maria help Peter to teach Hans how to swim'

Dutch

dat Karel Marie Piet Jan laat helpen leren zwemmen

Proof of non-context freeness

German

dass der Karl die Maria
dem Peter den Hans schwimmen lehren helfen lässt
'that Karl lets Maria help Peter to teach Hans how to swim'

Dutch

dat Karel Marie Piet Jan laat helpen leren zwemmen

Proof of non-context freeness

German

dass der Karl die Maria
dem Peter den Hans schwimmen lehren helfen lässt
'that Karl lets Maria help Peter to teach Hans how to swim'

Dutch

dat Karel Marie Piet Jan laat helpen leren zwemmen

Proof of non-context freeness

German

dass der Karl die Maria
dem Peter den Hans^m schwimmen lehren^m helfen lässt
'that Karl lets Maria help Peter to teach Hans how to swim'

Dutch

dat Karel Marie Piet Jan^m laat helpen leren^m zwemmen

Proof of non-context freeness

German

dass der Karl die Maria
dem Peter^n den Hans^m schwimmen lehren^m helfen^n lässt
'that Karl lets Maria help Peter to teach Hans how to swim'

Dutch

dat Karel Marie Piet^n Jan^m laat helpen^n leren^m zwemmen

Proof of non-context freeness

German

dass der Karl die Maria
dem Peter^n den Hans^m schwimmen lehren^m helfen^n lässt
'that Karl lets Maria help Peter to teach Hans how to swim'

- German structure corresponds to formal language $a^m b^n d^n c^m$ — context-free

Dutch

dat Karel Marie Piet^n Jan^m laat helpen^n leren^m zwemmen

- Dutch structure corresponds to formal language $a^m b^n c^m d^n$ — not context-free

Proof of non-context freeness

Swiss German

dass de Karl d'Maria em Peter^n de Hans^m laat hälfe^n lärne^m schwüme

Proof of non-context freeness

Swiss German

dass de Karl d'Maria em Peter^n de Hans^m laat hälfe^n lärne^m schwüme

Proof of non-context freeness

Swiss German

dass de Karl d'Maria em Peter^n de Hans^m laat hälfe^n lärne^m schwüme

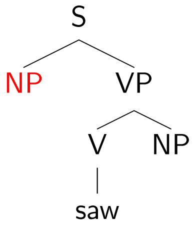
- Swiss German structure corresponds to formal language $a^m b^n c^m d^n$ — not context-free

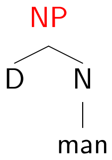
Tree Adjoining Grammars

- since around 1980 several attempts to move slightly beyond context-free power
- perhaps most influential: Aravind Joshi's **Tree Adjoining Grammars** (TAG)

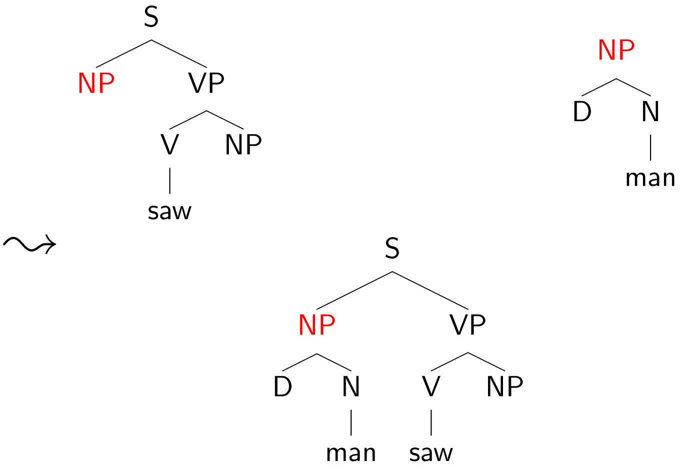
Tree Adjoining Grammars

Context-free derivations as tree growth



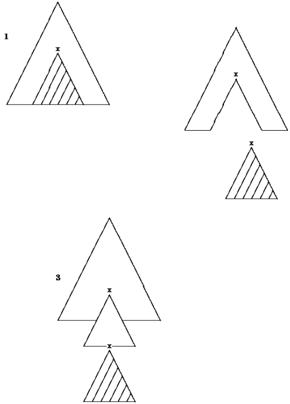


Context-free derivations as tree growth



Tree Adjoining Grammars

- TAG generelizes this to insertion of trees in the middle of other trees



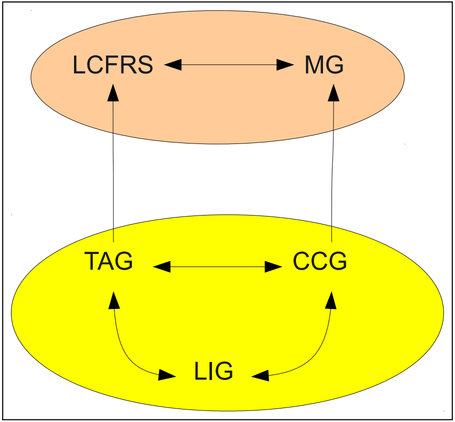
- creates **vertical** nested dependencies
- may cash out as crossing dependencies in the string

Related formalisms

- **Linear Indexed Grammars**: pushdown stack as part of context-free rules
- **Combinatory Categorial Grammar, Linear context-free rewriting systems, Head grammars**:
  - partial reshuffling of constituents during (essentially context-free) derivation
- **Minimalist grammars**
  - version of Chomsky's latest paradigm; formalized by Ed Stabler
  - lexically controlled movement of constituents during derivation possible

Mildly context-sensitive grammar formalisms

- two closely related families of mutually equivalent formalisms



Mildly context-sensitive grammar formalisms

TAG and relatives

- parsing problem $\mathcal{O}(n^6)$
- examples
  - $a^n b^m c^n d^m$
  - copy language
  - $a^n b^n c^n (d^n)$

LCFRS and relatives

- parsing problem in $PTIME$
- examples
  - $a^n b^n c^n d^n e^n$
  - triple-copy language (actually any k-copy language for fixed k)
  - $a_1^n \cdots a_k^n$ for fixed k

General properties of MCS grammar formalisms

- Joshi 1985: introduced the notion
- semi-formal characterization: a class of languages is mildly context-sensitive if
  - it contains all context-free languages
  - it can describe a limited number of types of cross-serial dependencies
  - its parsing problem is in PTIME
  - all languages in it have *constant growth property*
- last property excludes set of primes, set of square numbers etc.

Are all natural languages MCS?

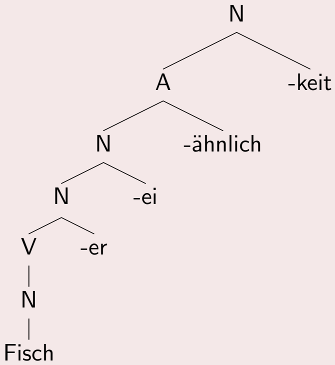
- Michaelis and Kracht 1997:
  Old Georgian is not semilinear.
- All MCS formalisms mentioned above describe semilinear languages, thus not all NL describable by TAGs or LCFRSs.

(2) govel-i igi sisxl-i saxl-isa-j m-is Saul-is-isa-j
ALL-NOM ART-NOM BLOOD-NOM HOUSE-GEN-NOM
ART-GEN SAUL-GEN-GEN-NOM
'all the blood of the house of Saul'

- subordinate nouns carry case marking for all superordinate nouns ("case stacking")
- if productive, this makes Old Georgian a non-LCFRS language
- open issue whether the pattern was productive or whether it exists productively in living languages

Recursion

German Morphology I: derivation

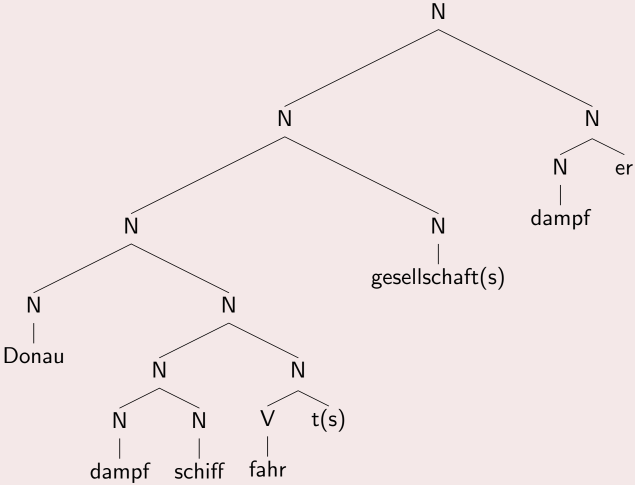


- recursive pattern
- regular

All infinite regular languages are recursive!

Phrase structure

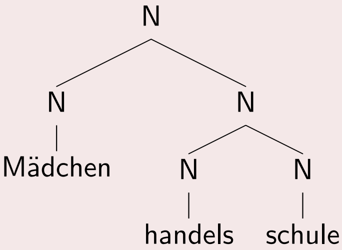
German Morphology II: compounding



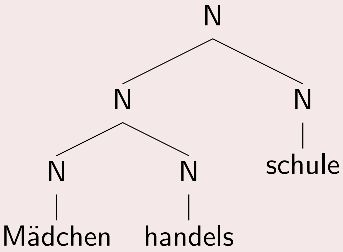
Phrase structure

German Morphology II: compounding

- On the level of strings, all of German morphology is regular!
- regular grammar for words does not capture semantic distinctions though:



'trade school for girls'



'school of girls trade'

Phrase structure

- issue of phrase structure is largely orthogonal to string complexity (↗ Tecumseh's talk yesterday)
- context free rules look like phrase structure rules, but:
  - a given cf language can be generated by a multitude of cf grammars (Greibach Normal Form, Chomsky Normal Form)
  - identification of the "correct" grammar (which predicts the correct constituent structure, according to standard tests) is non-trivial
  - induction of phrase structure from plain strings is very hard, even if context freeness is known
  - virtually impossible without statistical information (Klein & Manning 2004)
- as example of German word structure illustrates, regular string languages may have non-trivial constituent structure

Unbounded dependencies

Negation in Moroccan Arabic

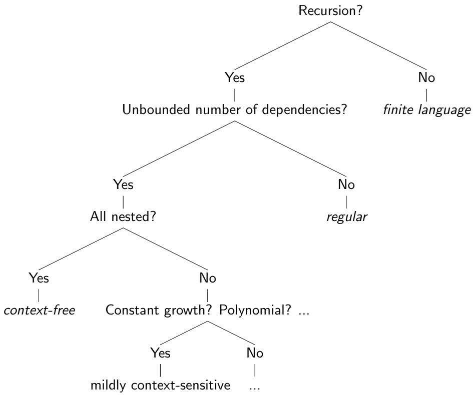
- Past: /kteb/ 'he wrote' /ma-kteb-ʃi/ 'he didn't write'
- Present: /ka-y-kteb/ 'he writes' /ma-ka-y-kteb-ʃi/ 'he doesn't write'
- /ma-kteb-hom-li-ʃi/ 'he didn't write them to me'
- /ma-ka-y-kteb-hom-li-ʃi/ 'he doesn't write them to me'
- /ma-ɣadi-y-kteb-hom-li-ʃi/ 'he won't write them to me'
- /waʃ ma-kteb-hom-li-ʃi/ 'didn't he write them to me?'
- / waʃ ma-ka-y-kteb-hom-li-ʃi/ 'doesn't he write them to me?'
- /waʃ ma-ɣadi-y-kteb-hom-li-ʃi/ 'won't he write them to me?'

All this is still regular!

Unbounded dependencies

- regular languages my display unbounded dependencies, i.e. dependencies of arbitrary length
- **However:** A regular language can only have a **bounded number** of unbounded dependencies
- it needs at least a context-free language to get an **unbounded number** of dependencies of **unbounded length**

To sum up...


Recursion?
Yes
No
Unbounded number of dependencies?
finite language
Yes
No
All nested?
regular
Yes
No
context-free
Constant growth? Polynomial? ...
Yes
No
mildly context-sensitive
...

Is the Chomsky Hierarchy a measure of cognitive complexity?

- short answer: NO
  - phone directory of New York City: finite, thus regular
  - set of first order theorems: recursive enumerable
  - still, a bright undergraduate performs fairly well on recognizing the latter after a few weeks of training, while only a few extraordinary individuals would be able to master the first
- long answer: yes, up to a point
  - you have to control the size of the grammar somehow (proposal: Kolmogorov complexity of shortest grammar generating a language)
  - given this, it is a good starting point

Relation of Chomsky Hierarchy to other complexity measures

Algorithmic complexity

- weak link: each class in the Chomsky hierarchy places an upper bound on the algorithmic complexity:
  - Type 0: recursively enumerable
  - context-sensitive: PSPACE
  - LCFRS: PTIME
  - TAG: $\mathcal{O}(n^6)$
  - context-free: $\mathcal{O}(n^3)$
  - regular: linear
- individual languages may have much lower complexity than to be expected from the smallest CH class they are contained in:
  - linear: copy language, k-copy language, $a_1^n \ldots a_k^n$
  - polynomial: set of square numbers

Relation of Chomsky Hierarchy to other complexity measures

Kolmogorov complexity

- intuitive idea: length of the shortest computer program that produces the object in question as its output
- measures complexity of strings (or objects representable as strings)
- not directly applicable to languages, i.e. sets of strings