Grammar in Performance and Acquisition: acquisition

E Stabler, UCLA

ENS Paris • 2008 • day 4

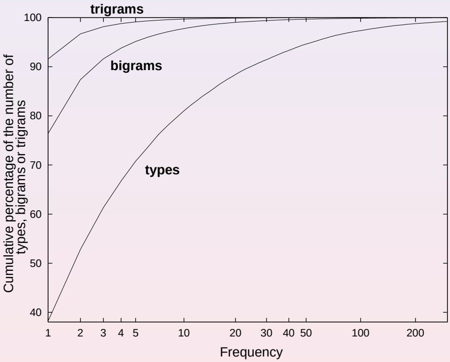
Q1 How are utterances interpreted 'incrementally'?

Q2 How is that ability acquired, from available evidence?

Q3 Why are some constituent orders unattested across languages?

Q4 What kind of grammar makes copying a natural option?

- we don't need to start from zero (start from grammar)
- frame explanations supported by convergent evidence



tb2: ≈40% words unique, 75% bigrams, 90% trigrams, 99.7% sentences

⇒ most sentences heard only once

Parameter setting: methodology

- How are fundamental properties of language learned? Important to distinguish 2 ideas:
  - Uncontroversially, we usually aim to understand how the basic parameters of language variation are set, abstracting away from other properties.
  - A controversial suggestion is that there may be a principled distinction between "core" parameters and "peripheral" parameters of variation, such that universal grammar "will make available only a finite class of possible core grammars, in principle," (Chomsky'81)

  The first idea is assumed here and in virtually all work on learning, in all domains; the second conjecture might or might not be true, and nothing mentioned here will depend on it.

Parameter setting: methodology

- How are fundamental properties of language learned?
- Gibson&Wexler'94: set n binary parameters on basis of input constituent orders
  $\langle vs, vos, vo_1o_2s, \ldots \rangle \mapsto$(spec-final, comp-final, not V2)
- ...in the case of Universal Grammar...we want the primitives to be concepts that can plausibly be assumed to provide a preliminary, prelinguistic analysis of a reasonable selection of presented data. it would be unreasonable to incorporate such notions as **subject of a sentence** or other grammatical notions, since it is unreasonable to suppose that these notions can be directly applied to linguistically unanalyzed data. (Chomsky, 1981)
- Suppose parameters are associated with (functional) heads, in the lexicon. (Presumably tightly constrained – more on this later) The learner needs to identify them...

Parameter setting: methodology

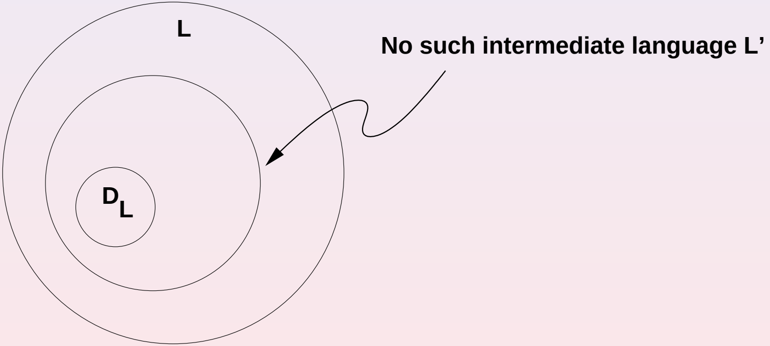
- How are fundamental properties of language learned?
- Gibson&Wexler'94: set n binary parameters on basis of input constituent orders
  $\langle vs, vos, vo_1o_2s, \ldots\rangle \mapsto$(spec-final, comp-final, not V2)

- ...in the case of Universal Grammar...we want the primitives to be concepts that can plausibly be assumed to provide a preliminary, prelinguistic analysis of a reasonable selection of presented data. it would be unreasonable to incorporate such notions as **subject of a sentence** or other grammatical notions, since it is unreasonable to suppose that these notions can be directly applied to linguistically unanalyzed data. (Chomsky, 1981)
- Suppose parameters are associated with (functional) heads, in the lexicon. (Presumably tightly constrained – more on this later) The learner needs to identify them...

Parameter setting: methodology

- How are fundamental properties of language learned?
- Gibson&Wexler'94: set n binary parameters on basis of input constituent orders
  $\langle vs, vos, vo_1 o_2 s, \ldots \rangle \mapsto$(spec-final, comp-final, not V2)
- ...in the case of Universal Grammar...we want the primitives to be concepts that can plausibly be assumed to provide a preliminary, prelinguistic analysis of a reasonable selection of presented data. it would be unreasonable to incorporate such notions as **subject of a sentence** or other grammatical notions, since it is unreasonable to suppose that these notions can be directly applied to linguistically unanalyzed data. (Chomsky, 1981)
- Suppose parameters are associated with (functional) heads, in the lexicon. (Presumably tightly constrained – more on this later) The learner needs to identify them...

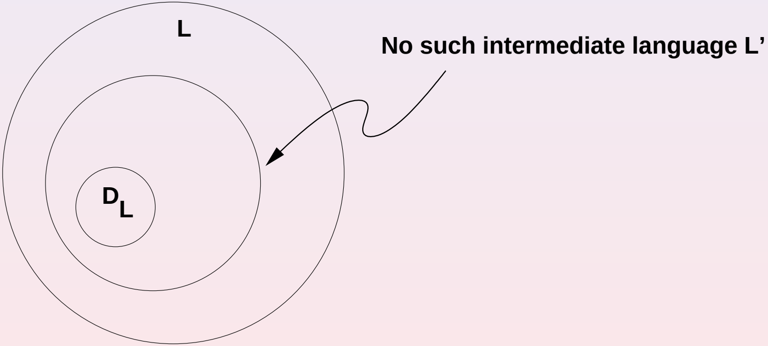
- (Gold, 1967; Angluin, 1980) A collection of languages is *perfectly identifiable from positive text* iff every **L** has finite subset $\mathbf{D_L}$



$\Rightarrow$ no superset of the class of finite languages is learnable in this sense

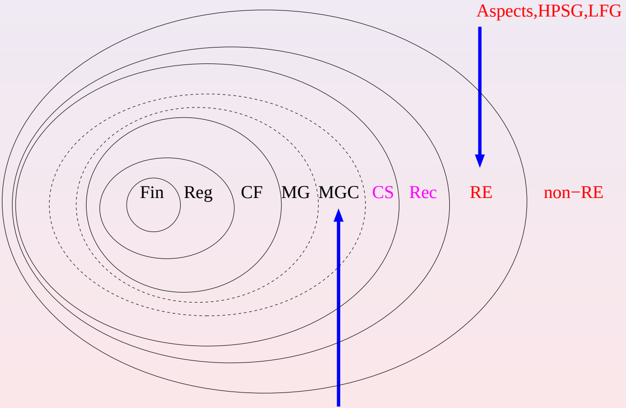
- (Pitt, 1989) If collection identifiable with $p > \frac{1}{2}$, then learnable in Gold's sense

- (Gold, 1967; Angluin, 1980) A collection of languages is *perfectly identifiable from positive text* iff every **L** has finite subset $\mathbf{D_L}$



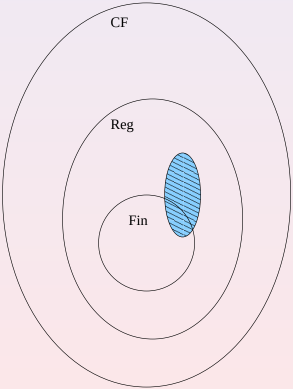
$\Rightarrow$ no superset of the class of finite languages is learnable in this sense

- (Pitt, 1989) If collection identifiable with $p > \frac{1}{2}$, then learnable in Gold's sense



$$\mathrm{CF} \subset \boxed{\mathrm{TAG} \equiv \mathrm{CCG}} \subset \boxed{\mathrm{MCFG} \equiv \mathrm{MG}} \subset \boxed{\mathrm{MGC} \subseteq \mathrm{PMCFG}} \subset \mathrm{CS}$$

A regular language is **0-reversible** iff $xz, yz \in L$ implies $\forall w,\ xw \in L$ iff $yw \in L$

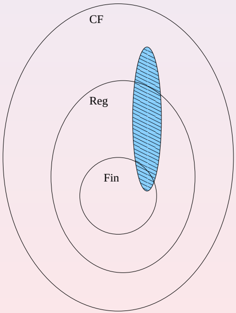


(Angluin'82): 0-reversible languages are learnable from positive text

A CFG is **very simple** iff every rule has form $A \longrightarrow a\alpha$ for pronounced (terminal) symbol a and sequence of categories α, where no two rules have the same pronounced element a.

Example:

S $\longrightarrow$ & S S
S $\longrightarrow$ ¬ S
S $\longrightarrow$ p
S $\longrightarrow$ q

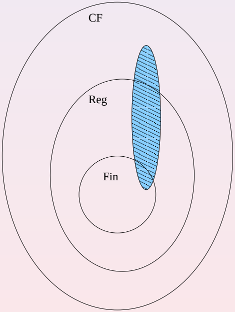


(Yokomori'03): VSLs are learnable from positive text

A CG is *k*-**valued** if no pronounced (terminal) symbol has more than *k* categories.

Example:
&::(S\S)/S
¬::S/S
p::S
q::S

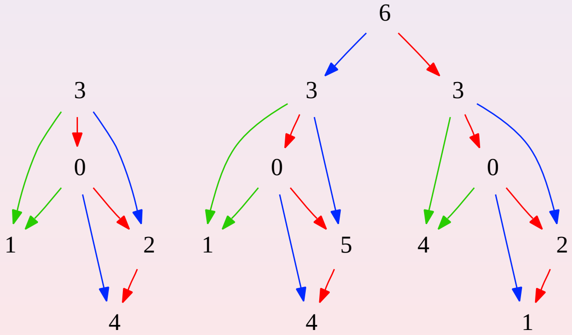
Example:
and::(S\S)/S
saw::(D\S)/D
saw::N
student::N
vegetarian::N
some::D/N
every::D/N



(Kanazawa'94): *k*-valued categorial languages are learnable from function-argument trees (and learnable in principle from strings)

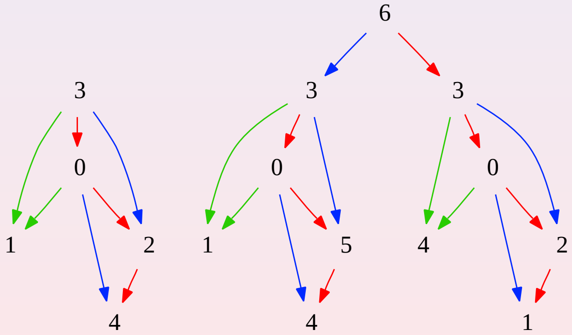
- **input:** 12340, 15340642310,. . .
- **Problem:** What is the language? Does the language have structures you have not seen?

- **input:** 12340, 15340642310,. . .
  dependencies (r,b,g)



- **Problem:** What is the language? Does the language have structures you have not seen?

- **input:** 12340, 15340642310, . . .
  dependencies (r,b,g), MG, lex unambiguous

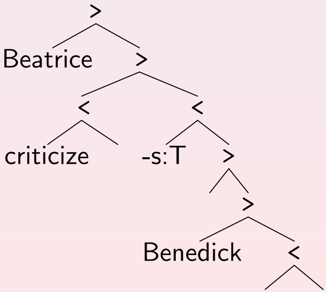


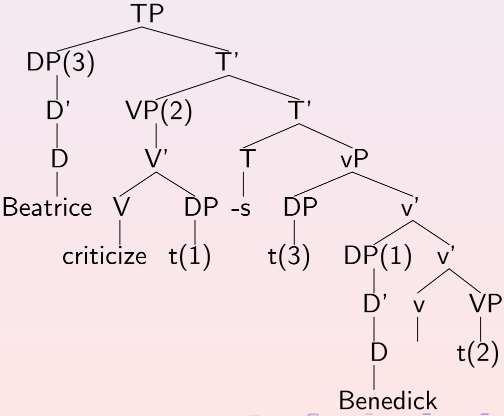
- **Problem:** What is the language? Does the language have structures you have not seen?

criticize::=D V -v　　praise::=D V -v

-s::=v +v +case T　　ϵ::=V +case =D v

Beatrice::D -case　　Benedick::D -case　　and:=T =T T





The same derivation in *tuple form*, fully explicit:

Beatrice criticize -s Benedick:T

criticize -s Benedick:+case T,Beatrice:-case

-s Benedick:+i +case T,criticize:-i,Beatrice:-case

-s::=v +i +case T   Benedick:v,criticize:-i,Beatrice:-case

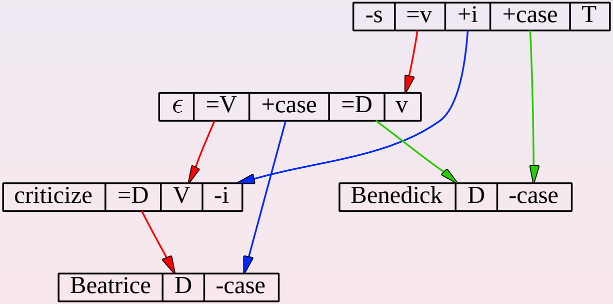
Benedick:=D v,criticize:-i   Beatrice::D -case

ϵ:+case =D v,criticize:-i,Benedick:-case

ϵ::=V +case =D v   criticize:V -i,Benedick:-case

criticize::=D V -i   Benedick::D -case

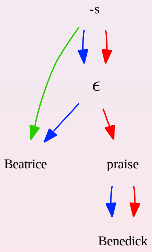
The same derivation as a *matching graph*:



(This graph completely determines the derivation)

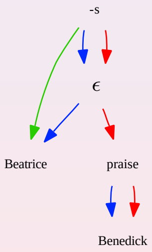
- Suppose the learner can identify these dependencies using semantic reasoning, but not the syntactic features. . . what do we have when features are removed?

Let's call these MG *dependency structures*:



- From these, the learner can identify the language.

Let's call these MG *dependency structures*:

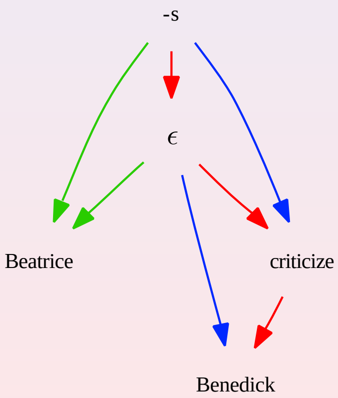


- From these, the learner can identify the language.

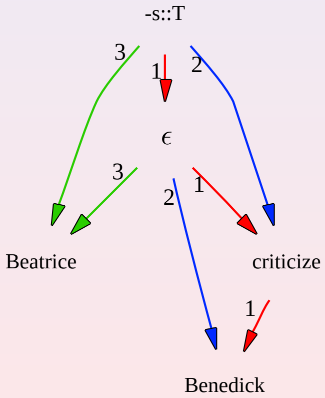
the learner: given a sequence of dependency structures. . .

1. label the root category
2. identify first arcs of non-root nodes, add new category labels
3. add new licensee features for each other incoming arc
4. add pre-category feature to match each outgoing arc
5. collect the lexicon
6. assuming no lexical ambiguity, unify features

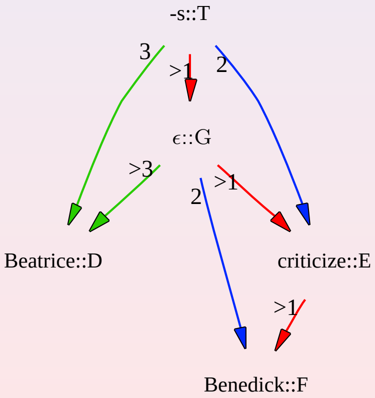
Input: $\langle d_1 \rangle$



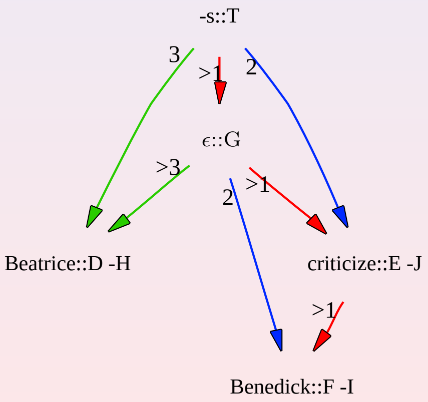
Step 1: Label root category



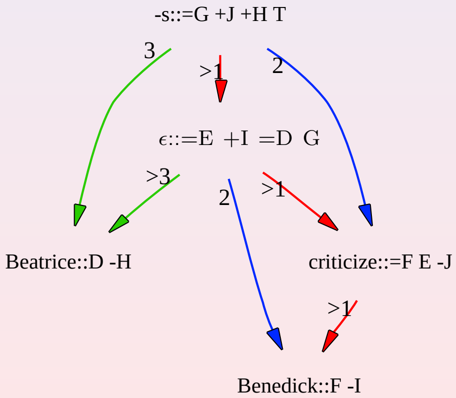
Step 2: Identify least incoming arcs of non-root nodes; add new category labels:



Step 3: Add new licensee features for each later incoming arc:



Step 4: Add precategory features to match other end of each outgoing arc, in order (r,b,g):



Step 5. collect lexicon: $GF(\langle d_1 \rangle)$ is then:

criticize::=F E -J	
-s::=G +J +H T	ϵ::=E +I =D G
Beatrice::D -H	Benedick::F -I

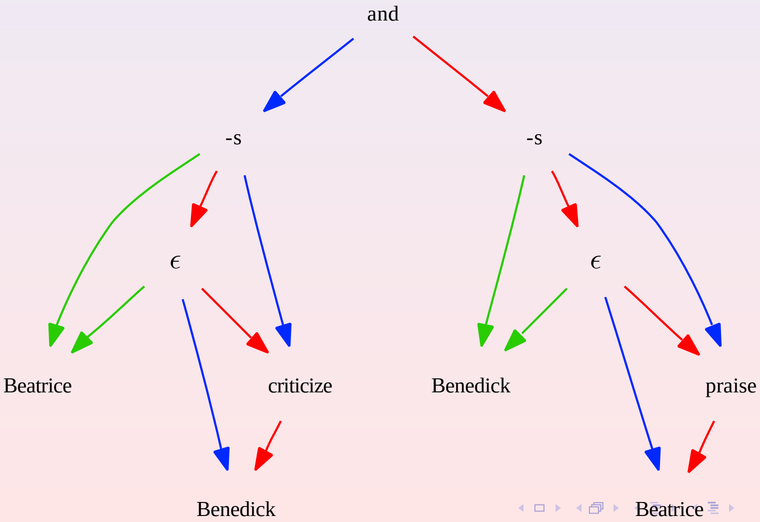
The result of this step is always a grammar that defines exactly the dependency trees given in the input; nothing more. The grammar generates exactly the input string(s).

Step 6. unify to make rigid: $GF(\langle d_1 \rangle)$ already rigid, so $GF(\langle d_1 \rangle) = RG(\langle d_1 \rangle)$

criticize::=F E -J	
-s::=G +J +H T	ϵ::=E +I =D G
Beatrice::D -H	Benedick::F -I

criticize::=D V -v	praise::=D V -v	
-s::=v +v +case T	ϵ::=V +case =D v	
Beatrice::D -case	Benedick::D -case	and::=T =T T

Input: $\langle d_1, d_2 \rangle$



Step 5: $GF(\langle d_1, d_2 \rangle)$ is then:

Beatrice::P -U	Benedick::O -V	and::=L =K T
Beatrice::S -Y	Benedick::C -X	
-s::=M +W +U K	ϵ::=N +V =P M	
criticize::=O N -W	praise::=S R -Z	
-s::=Q +Z +X L	ϵ::=R +Y =C Q	

criticize::=F E -J	
-s::=G +J +H T	ϵ::=E +I =D G
Beatrice::D -H	Benedick::F -I

NB: Again, $GF(\langle d_1, d_2 \rangle)$ does not generalize at all.

Step 6. unify to make rigid: $RG(\langle d_1, d_2 \rangle) =$

criticize::=D E -J   praise::=D E -J
-s::=G +J +H T ϵ::=E +H =D G
Beatrice::D -H   Benedick::D -H   and::=T =T T

criticize::=D V -v   praise::=D V -v
-s::=v +v +case T ϵ::=V +case =D v
Beatrice::D -case   Benedick::D -case   and::=T =T T

This strategy always works

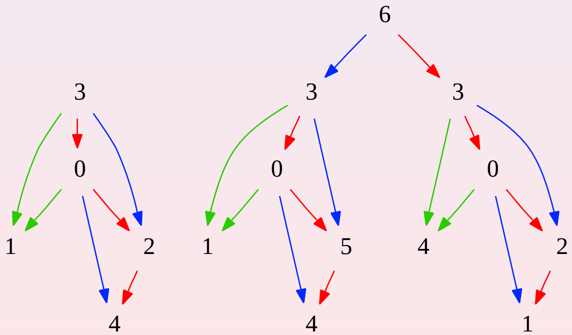
Step 6. unify to make rigid: $RG(\langle d_1, d_2 \rangle) =$

criticize::=D E -J	praise::=D E -J	
-s::=G +J +H T	ϵ::=E +H =D G	
Beatrice::D -H	Benedick::D -H	and::=T =T T

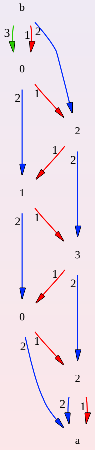
criticize::=D V -v	praise::=D V -v	
-s::=v +v +case T	ϵ::=V +case =D v	
Beatrice::D -case	Benedick::D -case	and::=T =T T

This strategy <u>always</u> works

- **input:** 12340, 15340642310
  Beatrice praise -s Benedick ϵ,
  Beatrice criticize -s Benedick ϵ and Benedick praise -s Beatrice ϵ.
  dependencies (r,b,g), MG, lex unambiguous



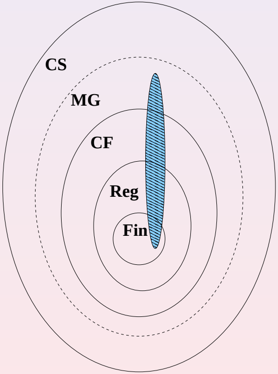
- **Problem:** What is the language?



a::C -r -l	b::=C +r +l T	ϵ::T
2::=C +r A -r	3::=C +r B -r	
0::=A +l C -l	1::=B +l C -l	

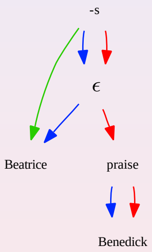
cross-serial dependencies by 'rolling-up' (non-CF)

A MG is **rigid** if each pronounced (terminal) symbol has at most 1 set of syntactic features.



Thm Given any rigid MG G, and any text of dependency structures t defined by G, this learning method will exactly identify the language after finitely many examples

structures from strings



- **Selection:** inferred from cognitively salient events
  - conditioned variation $\longrightarrow$ lexical categories
  - tight constraints on functional categories $\star$
- **Movement:** non-adjacency with related elements

ambiguity

- -s::=v +v +case T -s:=N Num (and more)
  ϵ::=V +case =D v ϵ::=T C (and more)
- read::=D V read::V read::N reed::N (and more)
  bill::=D V bill::V bill::N (and more)
- much ambiguity is systematic
- semantic features reduce syntactic ambiguity
- topic, semantic features from distributions?

ambiguity

- -s::=v +v +case T   -s:=N Num   (and more)
  ϵ::=V +case =D v ϵ::=T C   (and more)
- read::=D V   read::V   read::N   reed::N   (and more)
  bill::=D V   bill::V   bill::N   (and more)
- much ambiguity is systematic
- semantic features reduce syntactic ambiguity
- topic, semantic features from distributions?

ambiguity

- -s::=v +v +case T   -s:=N Num   (and more)
  ϵ::=V +case =D v ϵ::=T C   (and more)

- read::=D V   read::V   read::N   reed::N   (and more)
  bill::=D V   bill::V   bill::N   (and more)

- much ambiguity is systematic
- semantic features reduce syntactic ambiguity
- topic, semantic features from distributions?

ambiguity

- -s::=v +v +case T -s:=N Num (and more)
  ϵ::=V +case =D v ϵ::=T C (and more)
- read::=D V read::V read::N reed::N (and more)
  bill::=D V bill::V bill::N (and more)
- much ambiguity is systematic
- semantic features reduce syntactic ambiguity
- topic, semantic features from distributions?

Summary

- simple formalisms can model many linguistic proposals
  - **Q3** Why are some constituent orders unattested? (perhaps DTC?)
  - **Q4** What grammars make copying a natural option? (MGC?)
    - many open questions

Q1 What performance models allow incremental interpretation (and remnant movement, doubling constructions?)

- a straightforward semantics can value every MGC constituent
- CKY, Earley efficiently parses every MGC
- fit the performance data with a parser that works!

Q2 How is this ability acquired, from available evidence?

- rigid MGs can be learned from structures
- restricted possible structures aids: strings→structures
- many open questions!

Recap: the learner given a sequence of dependency structures. . .

1. label the root category
2. identify first arcs of non-root nodes, add new category labels
3. add new licensee features for each other incoming arc
4. add pre-category feature to match each outgoing arc
5. collect the lexicon
6. assuming no lexical ambiguity, unify features

Thm Given any rigid MG G, and any text of dependency structures t defined by G, this learning method will exactly identify the language after finitely many examples

Angluin, Dana. 1980. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135.

Angluin, Dana. 1982. Inference of reversible languages. *Journal of the Association for Computing Machinery*, 29:741–765.

Buszkowski, Wojciech and Gerald Penn. 1990. Categorial grammars determined from linguistic data by unification. *Studia Logica*, 49:431–454.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Foris, Dordrecht.

Gold, E. Mark. 1967. Language identification in the limit. *Information and Control*, 10:447–474.

Jain, Sanjay, Daniel Osherson, James S. Royer, and Arun Sharma. 1999. *Systems that Learn: An Introduction to Learning Theory (second edition)*. MIT Press, Cambridge, Massachusetts.

Kanazawa, Makoto. 1998. *Learnable Classes of Categorial Grammmars*. CSLI Publications, Stanford, California.

Kearns, Michael J. and Umesh V. Vazirani. 1994. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, Massachusetts.

Niyogi, Partha. 2006. *The Computational Nature of Language Learning and Evolution*. MIT Press, Cambridge, Massachusetts.

Pitt, Leonard. 1989. *Probabilistic inductive inference*. Ph.D. thesis, University of Illinois.

Retoré, Christian and Roberto Bonato. 2001. Learning rigid Lambek grammars and minimalist grammars from structured sentences. In L. Popelínský and M. Nepil, editors, *Proceedings of the Third Learning Language in Logic Workshop, LLL3*, pages 23–34, Brno, Czech Republic. Faculty of Informatics, Masaryk University. Technical report FIMU-RS-2001-08.

Rizzi, Luigi. 1994. Early null subjects and root null subjects. In Teun Hoekstra and Bonnie D. Schwartz, editors, *Language Acquisition Studies in Generative Grammar*. John Benjamins, Amsterdam, pages 151–176.

Shawe-Taylor, John and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press, NY.

Stabler, Edward P. 2002. Structures for learning. In *CoLogNet Lecture, ESSLLI'02*, Trento.

Stabler, Edward P., Travis C. Collier, Gregory M. Kobele, Yoosook Lee, Ying Lin, Jason Riggle, Yuan Yao, and Charles E. Taylor. 2003. The learning and emergence of mildly context sensitive languages. In W. Banzhaf, T. Christaller, P. Dittrich, J.T. Kim, and J. Ziegler, editors, *Advances in Artificial Life*. Springer, NY.

Yokomori, Takashi. 2003. Polynomial-time identification of very simple grammars from positive data. *Theoretical Computer Science*, 298:179–206.