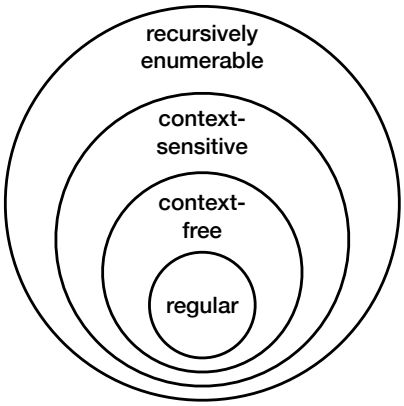
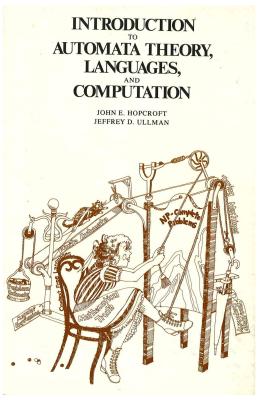
What Was Wrong with the Chomsky Hierarchy?

Makoto Kanazawa
Hosei University

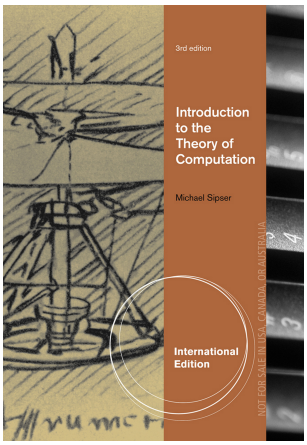
Chomsky Hierarchy of Formal Languages





Hopcroft and Ullman 1979

Enduring impact of Chomsky's work on theoretical computer science.
15 pages devoted to the Chomsky hierarchy.



Sipser 2013

No mention of the Chomsky hierarchy. The same with the latest edition of Hopcroft, Motwani, and Ullman (2006).

Semi-Thue System/String Rewriting System

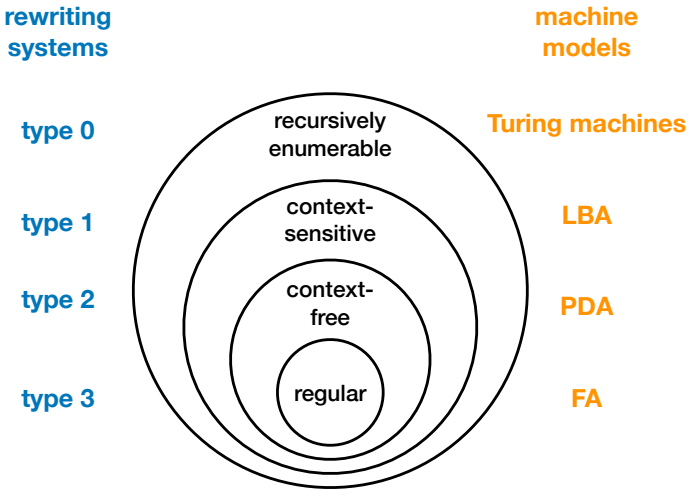
Production $\alpha \to \beta$ $\alpha, \beta \in (N \cup \Sigma)^*$

One-step rewriting $\gamma\alpha\delta \Rightarrow \gamma\beta\delta$

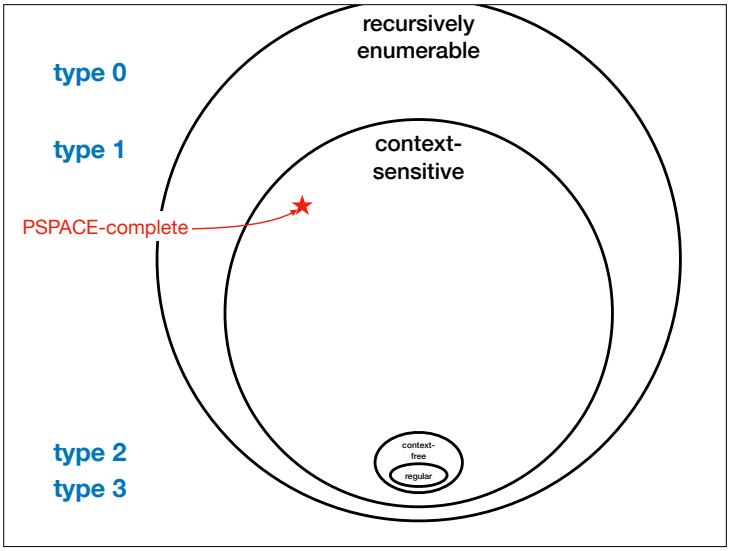
Language $L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* w \}$

Type 0	unrestricted
Type 1	$\lvert\alpha\rvert \leq \lvert\beta\rvert$
Type 2	$\alpha \in N$
Type 3	$\alpha \in N, \beta \in \Sigma N$

N: finite alphabet of nonterminal symbols
Σ: finite alphabet of terminal symbols



LBA: linear-bounded automata = NSPACE(n) Turing machines
PDA: pushdown automata
FA: finite automata



Context-sensitive languages are computationally very complex.
Some context-sensitive languages are PSPACE-complete.

- Context-sensitive languages are computationally **very** complex.
- There is a **huge** gap between context-free and context-sensitive.
- Many intermediate classes of languages have been considered in formal language theory.
- In particular, **"mildly context-sensitive"** grammars have been investigated by computational linguists.

Indexed Grammars

Hopcroft and Ullman 1979

14.3 INDEXED LANGUAGES

Of the many generalizations of context-free grammars that have been proposed, a class called "indexed" appears the most natural, in that it arises in a wide variety of contexts. We give a grammar definition here. Other definitions of the indexed languages are cited in the bibliographic notes.

An indexed grammar is a 5-tuple (V, T, I, P, S), where V is the set of variables, T the set of terminals, I the set of *indices*, S in V is the start symbol, and P is a finite set of productions of the forms

1) $A \to \alpha$, 2) $A \to Bf$, or 3) $Af \to \alpha$,

where A and B are in V, f is in I, and α is in $(V \cup T)^*$.

Example 14.2 Let $G = (\{S, T, A, B, C\}, \{a, b, c\}, \{f, g\}, P, S)$, where P consists of

$$S \to Tg, \qquad Af \to aA, \qquad Ag \to a,$$
$$T \to Tf, \qquad Bf \to bB, \qquad Bg \to b,$$
$$T \to ABC, \qquad Cf \to cC, \qquad Cg \to c.$$

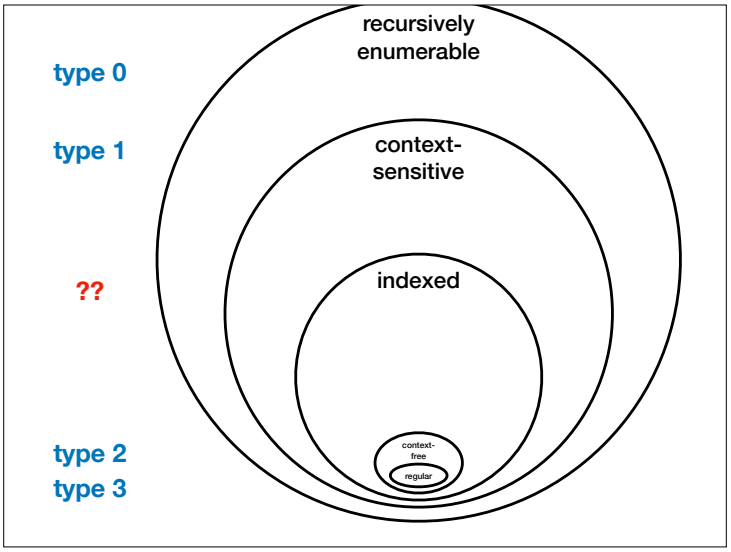
An example derivation in this indexed grammar is

$$\begin{aligned} S &\Rightarrow Tg \Rightarrow Tfg \Rightarrow AfgBfgCfg \\ &\Rightarrow aAgBfgCfg \Rightarrow aaBfgCfg \Rightarrow aabBgCfg \\ &\Rightarrow aabbCfg \Rightarrow aabbcCg \Rightarrow aabbcc. \end{aligned}$$

In general,

$$S \overset{*}{\Rightarrow} Tf^i g \Rightarrow Af^i gBf^i gCf^i g \overset{*}{\Rightarrow} a^{i+1}b^{i+1}c^{i+1}.$$

Theorem 14.7 (a) If L is accepted by a one-way nondeterministic stack automaton, then L is an indexed language. (b) If L is an indexed language, then L is a context-sensitive language.



What type of rewriting system is an indexed grammar??

Type 0

Production $\alpha \to \beta$

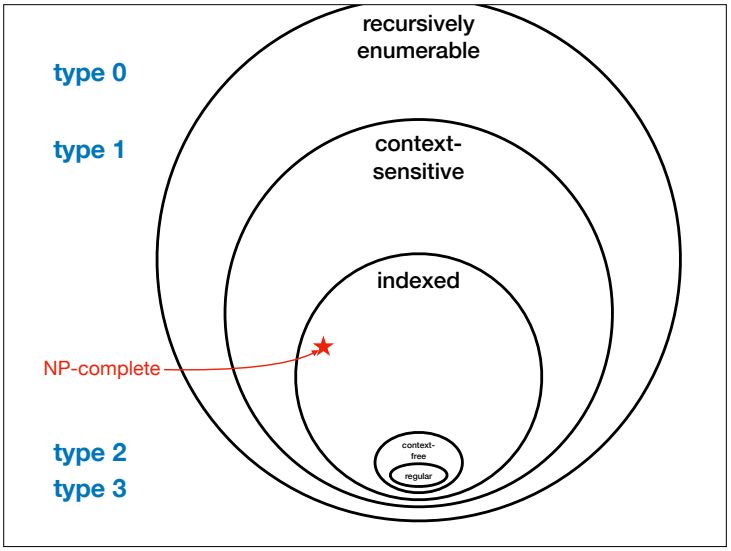
One-step rewriting $\gamma\alpha\delta \Rightarrow \gamma\beta\delta$

Indexed

Production $T \to ABC$

One-step rewriting $Tfg \Rightarrow AfgBfgCfg$

An indexed grammar is **not** an instance of a type 0 grammar.



Indexed languages are also sort of complex.

GERALD GAZDAR
UNIVERSITY OF SUSSEX
SCHOOL OF SOCIAL SCIENCES
BRIGHTON

APPLICABILITY
OF INDEXED GRAMMARS
TO NATURAL LANGUAGES

In the standard formulations, an indexed grammar can contain rules of three different sorts:

(1) i. $A[..] \rightarrow W[..]$
ii. $A[..] \rightarrow B[i,..]$
iii. $A[i,..] \rightarrow W[..]$

Suppose we allow the rule types shown in (2) in addition to those listed in (1):

(2) i. $A[..] \rightarrow W_1[]\ B[..]\ W_2[]$
ii. $A[..] \rightarrow W_1[]\ B[i,..]\ W_2[]$
iii. $A[i,..] \rightarrow W_1[]\ B[..]\ W_2[]$

Gazdar 1988

Vijayashanker 1987

The productions in an Indexed Grammar can be written as (refer [Gazdar 85a], the original definition of Indexed Grammars appears in [Aho 68])

$X[\cdot\cdot]\rightarrow X_1[i,\cdot\cdot]\ldots X_n[i,\cdot\cdot]$ (push) or

$X[i,\cdot\cdot]\rightarrow X_1[\cdot\cdot]\ldots X_n[\cdot\cdot]$ (pop) or

$X[\cdot\cdot]\rightarrow a$

In an LIG the productions have the form

$X[\cdot\cdot]\rightarrow X_1[]\ldots X_j[i,\cdot\cdot]\ldots X_n[]$ (push) or

$X[i,\cdot\cdot]\rightarrow X_1[]\ldots X_j[\cdot\cdot]\ldots X_n[]$ (pop) or

$X[\cdot\cdot]\rightarrow a$

LIG differs from IG in that the stack is passed on to only one of the children.

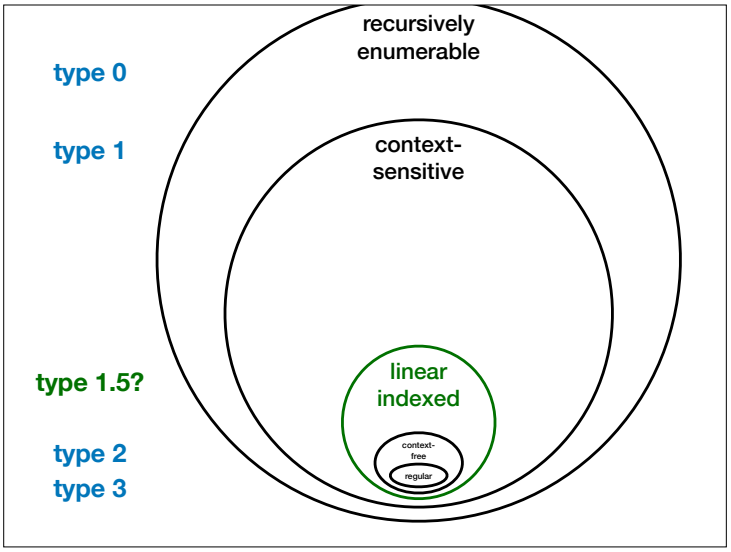
“Convergence” of Mildly Context-Sensitive Grammar Formalisms

TAG ≡ LIG ≡ HG

Vijay-Shanker and Weir 1994

TAG: tree-adjoining grammar; LIG: linear indexed grammar; HG: head grammar

---



There is a book where the author calls the linear indexed grammars “type 1.5”.

Type 0

Production $\alpha \rightarrow \beta$

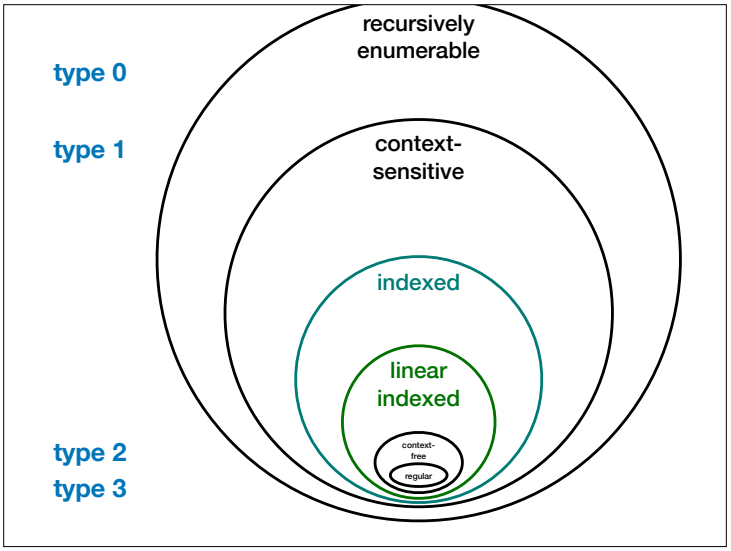
One-step rewriting $\gamma\alpha\delta \Rightarrow \gamma\beta\delta$

Linear indexed

Production $T[] \rightarrow AB[]C$

One-step rewriting $T[fg] \Rightarrow AB[fg]C$

A **linear** indexed grammar is **not** an instance of a type 0 grammar.



There are actually almost no other grammar formalisms that are instances of Chomsky's type 0 grammars.

Thue→Post→Chomsky

THE JOURNAL OF SYMBOLIC LOGIC
Volume 12, Number 1, March 1947

RECURSIVE UNSOLVABILITY OF A PROBLEM OF THUE

EMIL L. POST

Chomsky 1965

Confusion over this matter has been sufficiently persistent to suggest that a terminological change might be in order. Nevertheless, I think that the term "generative grammar" is completely appropriate, and have therefore continued to use it. The term "generate" is familiar in the sense intended here in logic, particularly in Post's theory of combinatorial systems. Furthermore, "generate" seems to be the most appropriate translation for Humboldt's term *erzeugen*, which he frequently uses, it seems, in essentially the sense here intended. Since this use of the term "generate" is well established both in logic and in the tradition of linguistic theory, I can see no reason for a revision of terminology.

How did Chomsky come to use semi-Thue systems as the most general type of grammars?
I believe Post (1947) coined the term "semi-Thue system".

Post Canonical Systems

FORMAL REDUCTIONS OF THE GENERAL COMBINATORIAL DECISION PROBLEM.*

By EMIL L. POST.

The operations of the system are a specified finite set of *productions*, each of the following form:

$$g_{11}P_{i'_1}\,g_{12}P_{i'_2}\cdots g_{1m_1}P_{i'_{m_1}}\,g_{1(m_1+1)}$$
$$g_{21}P_{i''_1}\,g_{22}P_{i''_2}\cdots g_{2m_2}P_{i''_{m_2}}\,g_{2(m_2+1)}$$
$$\cdots\cdots\cdots\cdots$$
$$g_{k_1}P_{i_1}{}^{(k)}\,g_{k_2}P_{i_2}{}^{(k)}\cdots g_{km_k}P_{i_{m_k}}{}^{(k)}\,g_{k(m_k+1)}$$

produce

$$g_1P_{i_1}\,g_2P_{i_2}\cdots g_mP_{i_m}\,g_{m+1}.$$

An indexed grammar is an instance of a Post canonical system.

Why Didn't Chomsky Use Post Canonical Systems?

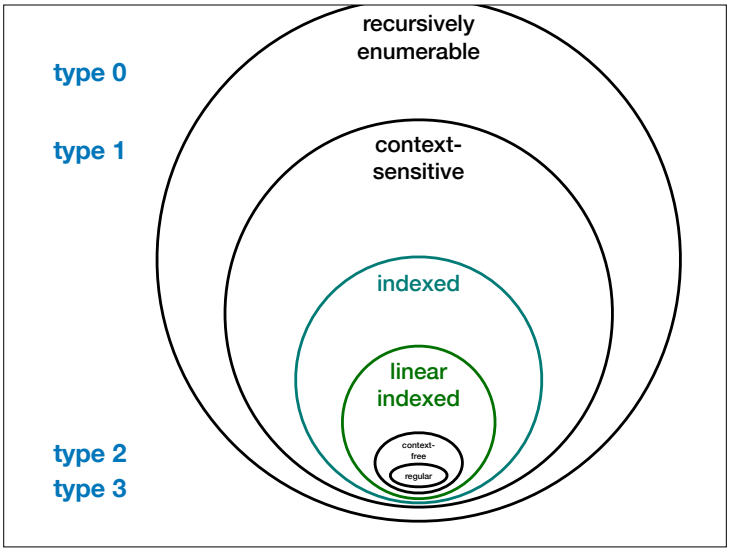
Post 1947

A very special case of the canonical form is what we term the normal form. A system in canonical form will be said to be in *normal form* if it has but one primitive assertion, and, each of its productions is in the form

$$g\,P$$
produces
$$P\,g'.$$

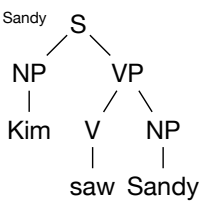
The main purpose of the present paper is to demonstrate that every system in canonical form can formally be reduced to a system in normal form. The two forms may therefore in fact be said to be *equipotent*.

Perhaps because of this striking result?



Should Context-Free Grammars Be Thought of as String Rewriting Systems?

S ⇒ NP VP ⇒ Kim VP ⇒ Kim V NP ⇒ Kim saw NP ⇒ Kim saw Sandy
S ⇒ NP VP ⇒ Kim VP ⇒ Kim V NP ⇒ Kim V Sandy ⇒ Kim saw Sandy
S ⇒ NP VP ⇒ NP V NP ⇒ Kim V NP ⇒ Kim saw NP ⇒ Kim saw Sandy
S ⇒ NP VP ⇒ NP V NP ⇒ Kim V NP ⇒ Kim V Sandy ⇒ Kim saw Sandy
S ⇒ NP VP ⇒ NP V NP ⇒ NP saw NP ⇒ Kim saw NP ⇒ Kim saw Sandy
S ⇒ NP VP ⇒ NP V NP ⇒ NP saw NP ⇒ NP saw Sandy ⇒ Kim saw Sandy
S ⇒ NP VP ⇒ NP V NP ⇒ NP V Sandy ⇒ Kim V Sandy ⇒ Kim saw Sandy
S ⇒ NP VP ⇒ NP V NP ⇒ NP V Sandy ⇒ NP saw Sandy ⇒ Kim saw Sandy



It is silly to define well-formed sentences in terms of derivations if what we really want is to assign tree structures to sentences.

Bottom-up Interpretation of Context-Free Rules

$$\frac{B_i \Rightarrow_G^* v_i \ (i = 1,\ldots,n) \qquad A \rightarrow w_0\, B_1\, w_1 \ldots B_n\, w_n \in P}{A \Rightarrow_G^* w_0\, v_1\, w_1 \ldots v_n\, w_n}$$

$$L(G) = \{\, w \in \Sigma^* \mid S \Rightarrow_G^* w \,\}$$

All we need is the subrelation of $\Rightarrow_G^*$ restricted to $N \times \Sigma^*$.
Just a general type of inductive definition.

Inductive Definition = CFG Interpreted Bottom-up

By ME_a is understood the set of *meaningful expressions* of type a; this notion has the following recursive definition:

(1) Every variable and constant of type a is in ME_a.
(2) If $\alpha \in ME_a$ and u is a variable of type b, then $\lambda u \alpha \in ME_{\langle b,a\rangle}$.
(3) If $\alpha \in ME_{\langle a,b\rangle}$ and $\beta \in ME_a$, then $\alpha(\beta) \in ME_b$.
(4) If $\alpha,\beta \in ME_a$, then $\alpha = \beta \in ME_t$.
(5) If $\phi,\psi \in ME_t$ and u is a variable, then $\neg\phi$, $[\phi \wedge \psi]$, $[\phi \vee \psi]$, $[\phi \rightarrow \psi]$, $[\phi \leftrightarrow \psi]$, $\forall u\phi$, $\wedge u\phi$, $\Box\phi$, $W\phi$, $H\phi \in ME_t$.
(6) If $\alpha \in ME_a$, then $\hat{}\alpha \in ME_{\langle s,a\rangle}$.

CFGs as Logic Programs on Strings

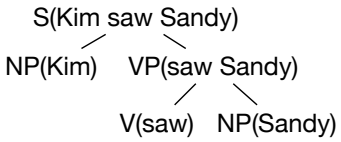
$$A \rightarrow w_0\, B_1\, w_1 \ldots B_n\, w_n$$

$$A(w_0\, \mathbf{x}_1\, w_1 \ldots \mathbf{x}_n\, w_n) \leftarrow B_1(\mathbf{x}_1),\ldots,B_n(\mathbf{x}_n)$$

Horn clause

$$L(G) = \{\, w \in \Sigma^* \mid G \vdash S(w) \,\}$$

Derivation = Proof Tree



$G \vdash S(\text{Kim saw Sandy})$

It's a small step to use n-ary predicates for $n \geq 2$.

Multiple Context-Free Grammars

$$A(\alpha_1,\ldots,\alpha_q) \leftarrow B_1(\mathbf{x}_{1,1},\ldots,\mathbf{x}_{1,q_1}),\ldots,B_n(\mathbf{x}_{n,1},\ldots,\mathbf{x}_{n,q_n})$$

$n \geq 0$, q, $q_i \geq 1$,
$\alpha_k \in (\Sigma \cup \{ \mathbf{x}_{i,j} \mid i \in [1,n], j \in [1,q_i] \})^*$
each $\mathbf{x}_{i,j}$ occurs exactly once in $(\alpha_1,\ldots,\alpha_q)$

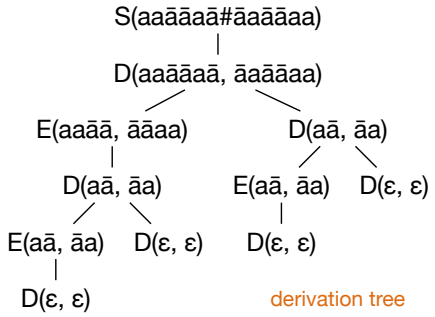
- $q = \dim(A)$ (**dimension** of A)
- $\dim(S) = 1$
- $L(G) = \{ w \in \Sigma^* \mid G \vdash S(w) \}$

It's best to think of an MCFG as a kind of logic program.
Each rule is a definite clause.
Nonterminals are predicates on strings.

$S(\mathbf{x}_1\#\mathbf{x}_2) \leftarrow D(\mathbf{x}_1, \mathbf{x}_2)$
$D(\varepsilon, \varepsilon) \leftarrow$
$D(\mathbf{x}_1\mathbf{y}_1, \mathbf{y}_2\mathbf{x}_2) \leftarrow E(\mathbf{x}_1,\mathbf{x}_2), D(\mathbf{y}_1,\mathbf{y}_2)$
$E(a\mathbf{x}_1\bar{a}, \bar{a}\mathbf{x}_2a) \leftarrow D(\mathbf{x}_1,\mathbf{x}_2)$

2-MCFG
2-ary branching

$\{ w\#w^R \mid w \in D_1^* \}$



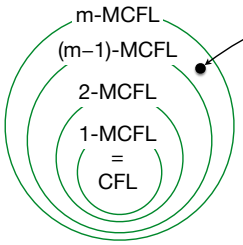
derivation tree

$$S(\mathbf{x}_1 \ldots \mathbf{x}_m) \leftarrow A(\mathbf{x}_1, \ldots, \mathbf{x}_m)$$

$$A(\varepsilon, \ldots, \varepsilon) \leftarrow$$

$$A(a_1\, \mathbf{x}_1\, a_2, \ldots, a_{2m-1}\, \mathbf{x}_m\, a_{2m}) \leftarrow A(\mathbf{x}_1, \ldots, \mathbf{x}_m)$$

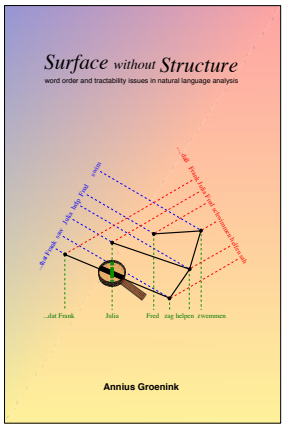
non-branching m-MCFG



$$\{\, a_1^n\, a_2^n \ldots a_{2m-1}^n\, a_{2m}^n \mid n \geq 0 \,\}$$

Seki et al. 1991

MCFGs as Logic Programs on Strings



s-rel:	$S(\ldots \texttt{dat}\ s\ do\ hv)$	:-	$NP(s),\ \overline{V}(d,\ o,\ h,\ v).$
s-decl:	$S(s\ h\ do\ v)$	:-	$NP(s),\ \overline{V}(d,\ o,\ h,\ v).$
s-inter:	$S(h\ s\ do\ v)$	:-	$NP(s),\ \overline{V}(d,\ o,\ h,\ v).$
s-topic:	$S(d\ h\ s\ o\ v)$	:-	$NP(s),\ \overline{V}(d,\ o,\ h,\ v).$
	$\overline{V}(d,\ \lambda,\ h,\ \lambda)$	:-	$VT(h),\ NP(d).$
	$\overline{V}(n,\ do,\ r,\ hv)$	:-	$VR(r),\ NP(n),\ \overline{V}(d,\ o,\ h,\ v).$

NP(Jan). NP(Marie). NP(koffie).
NP(jij). NP(wie).
VT(drinken). VR(zag).
...

Fig. 4. An LCFRS in predicate notation for Dutch sentential forms.

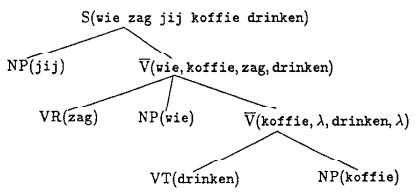
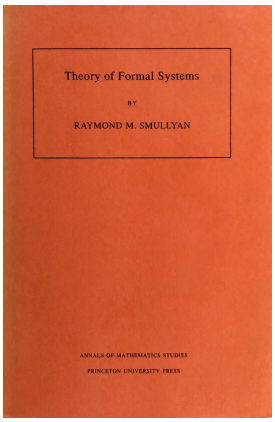


Fig. 5. LCFRS derivation of *Wie zag jij koffie drinken*?

Elementary Formal Systems



Smullyan 1961

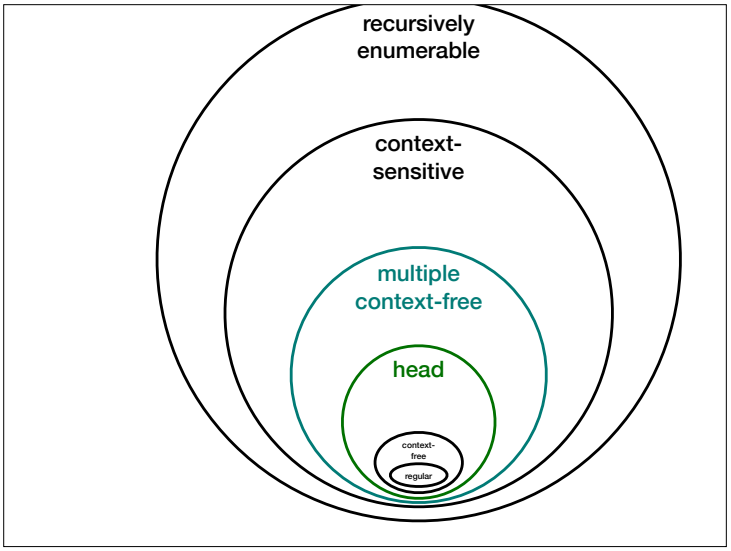
Elementary Formal Systems

Smullyan 1961

"Elementary Formal Systems can be looked at as variants of the canonical languages of Post …. The reason for our choice of elementary formal systems, in lieu of Post's canonical systems, is that their structure is more simply described, and their techniques are more easily applied. The general notion of "production" used in Post systems is replaced by the simpler logistic rules of substitution and detachment (modus ponens), which are more easily formalized."

Chomsky Hierarchy

Rewriting Systems	Machines	Logic Programs on Strings	Languages
Type 0	Turing	Elementary Formal Systems (Smullyan 1961)	r.e.
Type 1	LBA	Length-Bounded EFS (Arikawa et al. 1989)	CSL = NSPACE(n)
	Poly-time Turing	Simple LMG (Groenink 1997) / Hereditary EFS (Ikeda and Arimura 1997)	P
		MCFG	MCFL
Type 2	PDA	Simple EFS (Arikawa 1970)	CFL
Type 3	FA		Reg



Chomsky on Mathematical Linguistics (1979)

> … what happened is that there was a period of fairly fruitful contact between automata theory and linguistics in the late fifties and the early sixties. It mostly had to do with the properties of context-free grammars. Things turned up which are quite interesting.

> … Certainly context-free grammars represent some of the properties of languages. This seems to me what one would expect from applied mathematics, to see if you can find systems that capture some of the properties of the complex system that you are working with, and to ask whether those systems have any intrinsic mathematical interest, and whether they are worth studying in abstraction. And that has happened at exactly one level, the level of context-free grammar.

Chomsky (2004): *Generative Enterprise Revisited*

A long quote, from an interview held in the year when Hopcroft and Ullman's textbook was published.

Chomsky on Mathematical Linguistics (1979)

> … At any other level it has not happened. The systems that capture other properties of language, for example transformational grammar, hold no interest for mathematics. But I do not think that that is a necessary truth. It could turn out that there would be richer or more appropriate mathematical ideas that would capture other, maybe deeper properties of language than context-free grammars do. In that case you have another branch of applied mathematics which might have linguistic consequences. That could be exciting.

Chomsky (2004): *Generative Enterprise Revisited*

Summary

- The four levels of the Chomsky hierarchy are not of equal importance.
- The choice of semi-Thue systems as the most general form of grammar hampered investigation of some important language classes.
- Smullyan's elementary formal system provides the right level of generality and simplicity.
- Bottom-up semantics of rules is basic.
- Natural generalization of "inductive definitions".

What is the Significance of Grammar Formalisms?

- Abstract conception of grammars.
- Makes possible investigation of **algorithmic** properties of grammars.
- Should have good formal properties (like familiar closure properties).
- The choice of a grammar formalism should **not** be thought of as a tight characterization of possible human grammars.
- A grammar formalism does not have explanatory power.
