Multiple Context-Free Grammars: Basic Properties and Complexity

Hiroyuki Seki
NAIST

MCFG+2
Nara, Sept 9, 2011

Chomsky hierarchy

regular →

context-free(CF) → context-sensitive(CS)

→ phrase structure(PS)

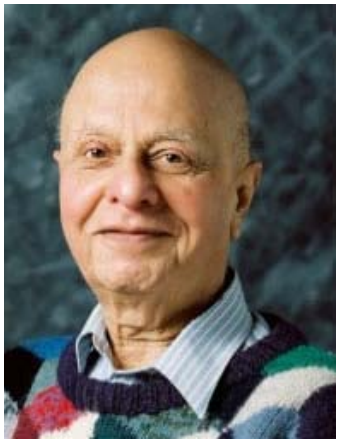
Noam Chomsky (1928-)

Mild context-sensitivity

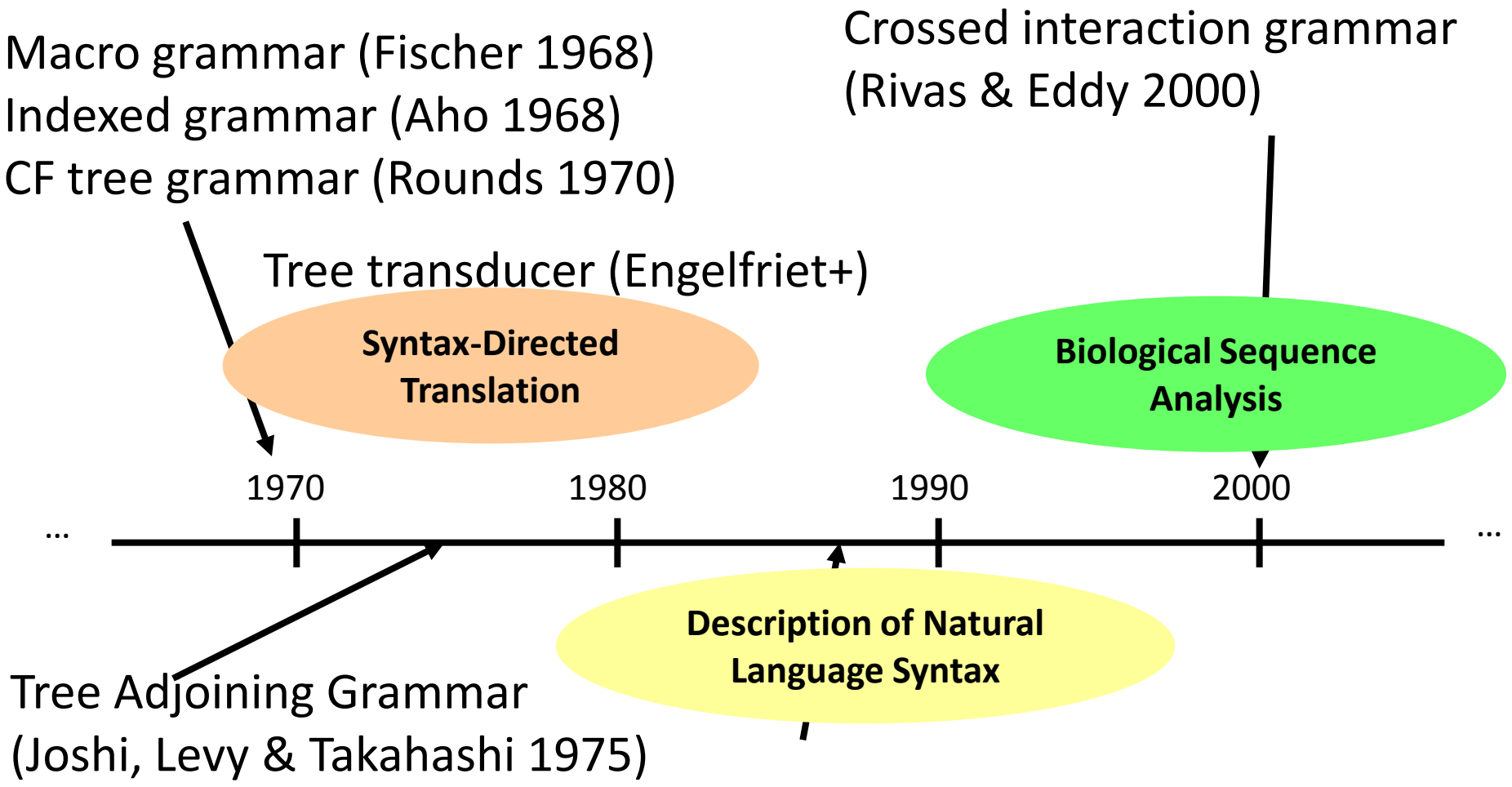
regular →

context-free(CF) → context-sensitive(CS)

→ phrase structure(PS)

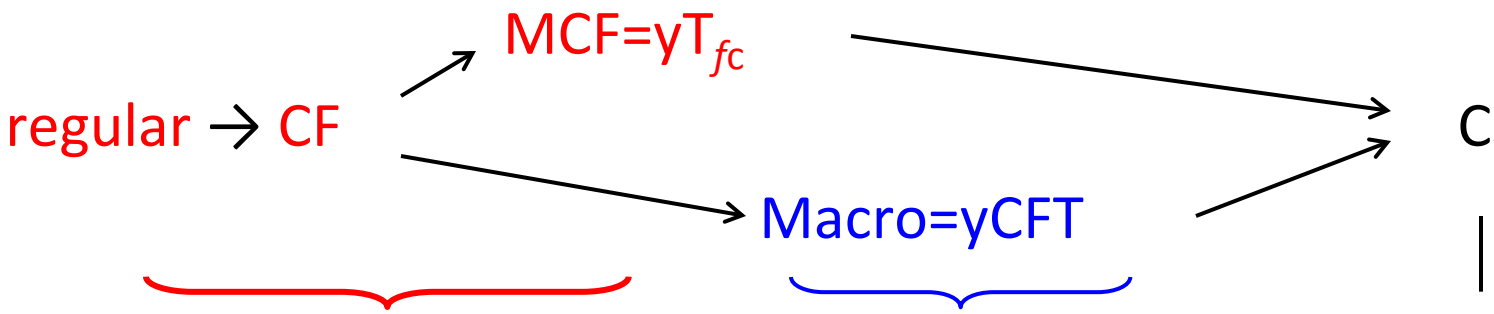


Arvind Joshi (1929-)



Linear CF rewriting system (Vijay-Shanker, Weir & Joshi 1987)
= Multiple CF grammar (Kasami, Seki & Fujii 1987),

Why MCFG ?



- Emptiness: linear time exp. time undecidable
- Membership: poly time NP-hard PSPACE

MCF: multiple CF languages
yT$_{fc}$: finite-copying tree to string transducers
Macro: macro languages
yCFT: yield of context-free tree languages

MCFG

- Kasami, Seki & Fujii, Tech. Rep., Osaka U. 1987, also in IEICE Trans., J71-D-I(5,6), 1988.
- Seki, Matsumura, Fujii & Kasami, TCS 88(2), 1991.
- Seki, Nakanishi, Kaji, Ando & Kasami, 31st ACL, 130-139, 1993.
- Kaji, Nakanishi, Seki & Kasami, Computational Intelligence, 10(4), 440-452, 1994., etc.

Tadao Kasami (1930-2007)

Contents

- Definitions
- Basic properties
- Recognition complexity
- Appendix

CFG as a set of clauses

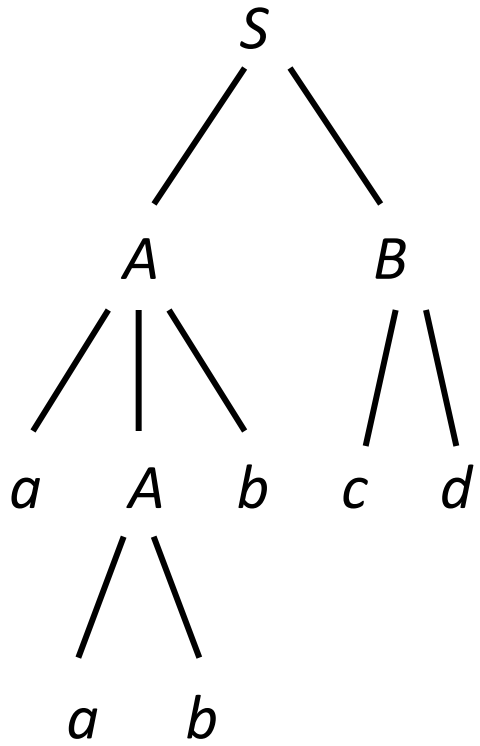
CFG		DHC (Definite Horn Clause)
Nonterminal symbol A	$\Leftrightarrow$	Unary predicate A $A(x)$ "A can derive x."
rules	$\Leftrightarrow$	clauses
$A \rightarrow B\ C$		$A(xy)$:- $B(x)$, $C(y)$.
$A \rightarrow a$		$A(a)$.
($B \Rightarrow^* u$ and $C \Rightarrow^* v$) implies $A \Rightarrow^* uv$.	$\Leftrightarrow$	($B(u)$ and $C(v)$) implies $A(uv)$.

Example

$S \rightarrow AB$

$A \rightarrow aAb \mid ab$

$B \rightarrow cBd \mid cd$

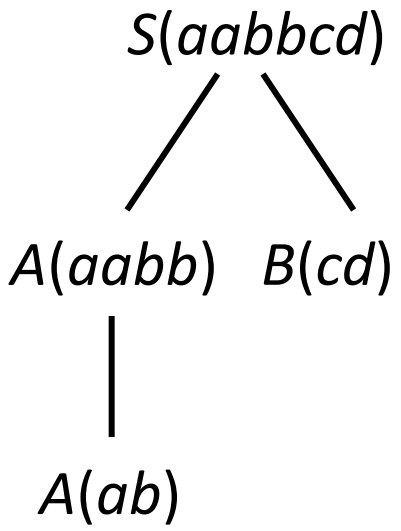


Example

$S(xy)$:- $A(x)$, $B(y)$.

$A(axb)$:- $A(x)$. $A(ab)$.

$B(cxd)$:- $B(x)$. $B(cd)$.



Derivation of *aabbcd* from *S*

= Proof of *S*(*aabbcd*).

From CFG to MCFG

CFG rule

$A \rightarrow B\ C$

$A \rightarrow a$

DHC (Unary predicates only)

$A(xy)$:- $B(x)$, $C(y)$.

$A(a)$.

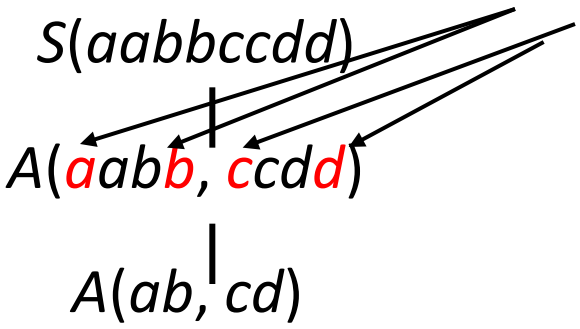
↓ Extension (arity≧1)

(Ex) $S(xy)$:- $A(x, y)$.

$A(axb, cyd)$:- $A(x, y)$.

$A(ab, cd)$.

synchronized



Example

G_1: $S(xy)$:- $A(x, y)$.

$A(axb, cyd)$:- $A(x, y)$.

$A(ab, cd)$.

$L(G_1) = \{ a^n b^n c^n d^n \mid n \geqq 1 \}$

$S(a^n b^n c^n d^n)$

|

$A(a^n b^n, c^n d^n)$

|

:

|

$A(aabb, ccdd)$

|

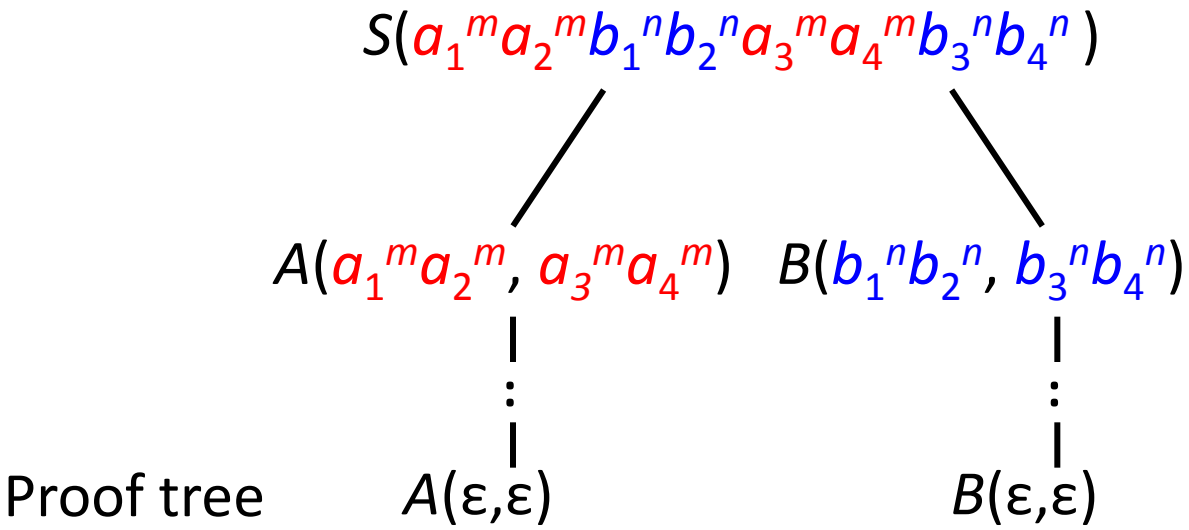
$A(ab, cd)$

Example

G_2: $S(x_1y_1x_2y_2)$:- $A(x_1, x_2)$, $B(y_1, y_2)$.

$A(a_1xa_2, a_3ya_4)$:- $A(x, y)$. $A(\varepsilon, \varepsilon)$.

$B(b_1xb_2, b_3yb_4)$:- $B(x, y)$. $B(\varepsilon, \varepsilon)$.



$L(G_2) = \{ a_1^m a_2^m b_1^n b_2^n a_3^m a_4^m b_3^n b_4^n \mid m, n \geqq 0 \}$

MCFG

$G=(N, T, V, P, S)$

N: predicates (nonterminals), T: terminals, V: variables,

P: rules, $S \in N$: start predicate

- A rule $\pi \in P$ is a definite Horn clause s.t.
  - Both head & body are linear (w. r. t. V),
  - Argument of predicate in body is variable,
  - Argument of predicate in head is

    string over T and variables in its body.

(Ex) $A(axb, cyd) :- A(x, y).$

$B(x_1y_1, x_2y_2) :- C(x_1, x_2), D(y_1, y_2).$

NG $A(x, ax) :- B(x).$

NG $A(x, y) :- B(x, y), C(x).$

Derivation

Let $G=(N, T, V, P, S)$ be an MCFG.

$A(w_1, \ldots, w_{\mathrm{arity}(A)})$ is *derivable* in G

if

$A(\alpha_1, \ldots, \alpha_{\mathrm{arity}(A)})$:-

$A_1(\ldots), \ldots, A_j(x_{j1}, \ldots, x_{j\ \mathrm{arity}(Aj)}), \ldots, A_n(\ldots) \in P,$

$A_j(u_{j1}, \ldots, u_{j\ \mathrm{arity}(Aj)})$ $(j \in [1..n])$ are derivable in G,

$w_i = \theta(\alpha_i)$ $(i \in [1..\deg(A)])$

where $\theta(x_{jk})=u_{jk}$. $(j \in [1..n],\ k \in [1..\mathrm{arity}(A_j)])$

MCFG Language

For an MCFG $G=(N, T, V, P, S)$,

$L(G) := \{ w \in T^* \mid S(w) \text{ is derivable in } G \}$

is the *multiple context-free language (mcfl)* generated by G.

Non-deleting MCFG

MCFG $G=(N, T, V, P, S)$

Non-deleting: $\forall \pi \in P, \mathrm{Var}_{\mathrm{head}}(\pi)=\mathrm{Var}_{\mathrm{body}}(\pi)$.

OK $B(x_1 y, x_2)$:- $C(x_1, x_2), D(y)$.

NG $B(y, x_2)$:- $C(x_1, x_2), D(y)$.

Lemma: $\forall$ MCFG G,

$\exists$ non-deleting MCFG G' s.t. $L(G') = L(G)$.

dim, rank, deg

MCFG $G=(N, T, V, P, S)$

- rule π: $A_0(\ldots)$:- $A_1(\ldots), A_2(\ldots), \ldots, A_n(\ldots)$.

  dim(π) := max { arity(A_i) | $0 \leqq i \leqq n$}, rank(π) := n,

  deg(π) := $\sum_{0 \leqq i \leqq n}$ arity(A_i).

  (deg $\leqq$ (rank + 1) * dim)

(Ex)

π_1: $A(axb, cyd)$:- $A(x, y)$.

dim(π_1)=2, rank(π_1)=1, deg(π_1)=4.

π_2: $B(x_1y_1, x_2y_2)$:- $C(x_1, x_2)$, $D(y_1, y_2)$.

dim(π_2)=2, rank(π_2)=2, deg(π_2)=6.

dim, rank, deg (cnt'd)

For MCFG $G=(N, T, V, P, S)$,

$\{\dim, \text{rank}, \deg\}(G) := \max_{\pi \in P} \{\dim, \text{rank}, \deg\}(\pi)$

- q-MCFG(r): MCFG with $\dim \leqq q$ and $\text{rank} \leqq r$
- q-MCFG: MCFG with $\dim \leqq q$
- MCFG(r): MCFG with $\text{rank} \leqq r$

Notation

(Class of) Grammars	CFG	MCFG	q-MCFG(r)	…
(Class of) Languages	CFL	MCFL	q-MCFL(r)	…

Properties of MCFG

(Also see [Vijay-Shanker, Weir & Joshi 1987])

- (Generative power) CFL $\subsetneq$ MCFL $\subsetneq$ CSL.
- Every MCFL is semilinear.
- (Closure property)
  - full AFL.
  - Not closed under intersection.
- (Decidability)
  - Emptiness ($L(G)=\Phi$?): $O(|G|)$-time decidable.
  - Recognition ($w \in L(G)$?): poly-time decidable.
  - Inclusion ($L(G_1) \subseteq L(G_2)$?) : Undecidable.

full AFL : a class of languages closed by homomorphism, inv. homomorphism, intersection with regular sets, union, Kleene closure

Proof example (exercise)

(Closure under intersection with regular language.)

CFL case:

For given CFG in Chomsky normal form $G=(N, T, P, S)$ &

FA (finite automaton) $M=(Q, T, \delta, p_0, F)$

(Q: set of states, δ: transition function, p_0: initial state, F: set of final states),

construct CFG $G'=(N \times Q \times Q \cup \{S'\}, T, P', S')$ where

$\cap$ R Closure: CFG Case

- $\forall A \rightarrow BC \in P,\ \forall q_1, q_2, q_3 \in Q$:

  $A[q_1,q_3] \rightarrow B[q_1,q_2]\ C[q_2,q_3] \in P'$

- $\forall A \rightarrow a \in P,\ \forall q_1, q_2 \in Q$ s.t. $\delta(q_1, a)=q_2$:

  $A[q_1,q_2] \rightarrow a \in P'$

Correctness:

$A[q_1,q_2] \Rightarrow^*_{G'} w$ iff

$A \Rightarrow^*_{G} w$ and $\delta^*(q_1, w)=q_2$

∩ R Closure: MCFG Case (1/2)

$A(x_1y_1, y_2x_2)$:- $B(x_1, x_2), C(y_1, y_2) \in P$

$A[?, ?, ?, ?]\ (x_1y_1, y_2x_2)$:-

$B[q_1,q_2,q_3,q_4]\ (x_1, x_2), C[r_1,r_2,r_3,r_4]\ (y_1, y_2) \in P'$

$A[q_1,r_2,r_3,q_4]\ (x_1y_1, y_2x_2)$:-

$B[q_1,q_2,q_3,q_4]\ (x_1, x_2), C[q_2,r_2,r_3,q_3]\ (y_1, y_2) \in P'$

q_1 $q_2=r_1$ r_2 r_3 $r_4=q_3$ q_4

∩ R Closure: MCFG Case (2/2)

Correctness:

$A[q_{1s},q_{1e},\ldots,q_{ns},q_{ne}](w_1,\ldots,w_n)$ provable in G' iff

$A(w_1,\ldots,w_n)$ derivable in G

and

$\delta^*(q_{js}, w)=q_{je}$ $(j \in [1..n])$

Properties of q-MCFG(r)

- q-MCFL(r), MCFL(r), q-MCFL ($r \geqq 2$, $q \geqq 1$):
  substitution closed full AFL.
- q-MCFL = $yT_{FIN(q)}$ [Weir92]
  (yield of tree transducers with copy bound q)
- q-MCFL(1) = $ET0L_{FIN(q)}$
  (DT0L with copy bound q)
- TAL $\subsetneqq$ 2-MCFL(2)

Pumping Lemma for CFL

$\forall L \in \text{CFL}\ \exists n \geqq 1\ \forall z \in L\ (|z| \geqq n)$

$\exists$ partition $z = uvwxy$ such that $|vx| \geqq 1$

$\forall i \geqq 0 : z_i = uv^i wx^i y \in L$.

Pumping Lemma for MCFL

For general MCFL :

$\forall L \in q\text{-MCFL}\ \exists n \geqq 1\ \exists z \in L\ (|z| \geqq n)$

$\exists$ partition $z = u_1 v_1 w_1 s_1 u_2 \ldots u_q v_q w_q s_q u_{q+1}$

$\Sigma |v_j s_j| \geqq 1$

$\forall i \geqq 0 : z_i = u_1 v_1^i w_1 s_1^i u_2 \ldots u_q v_q^i w_q s_q^i u_{q+1} \in L$.

For well-nested MCFL [Ka09] :

$\forall L \in q\text{-wnMCFL}\ \exists n \geqq 1\ \forall z \in L\ (|z| \geqq n)$

$\exists$ partition z = … (same as above) …

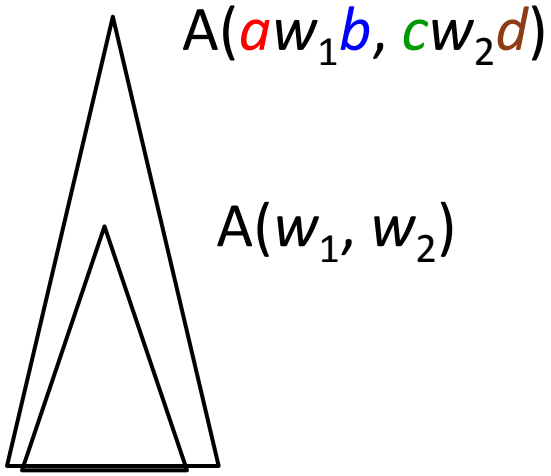
Recently, it was proved that the strong version does not hold for general MCFL.

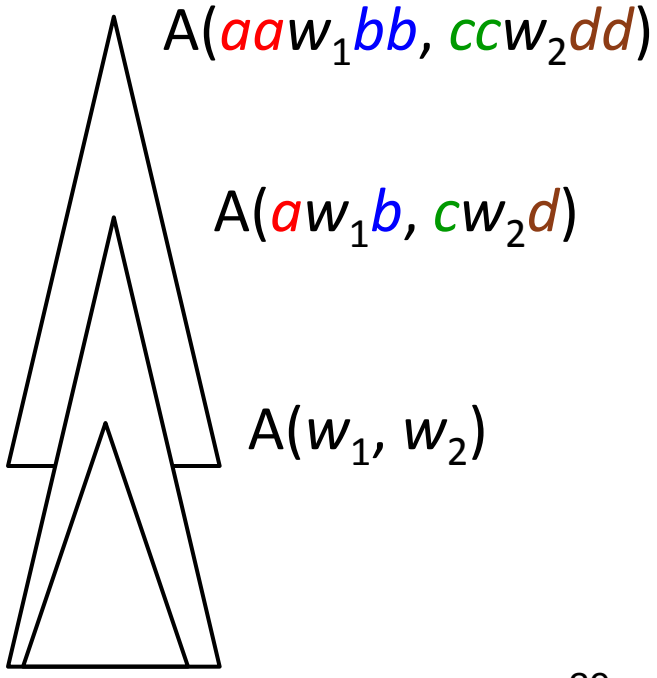
[Ka09] I
context-free languages, DLT09, LNCS 5583.

Pumping lemma

An easy case.

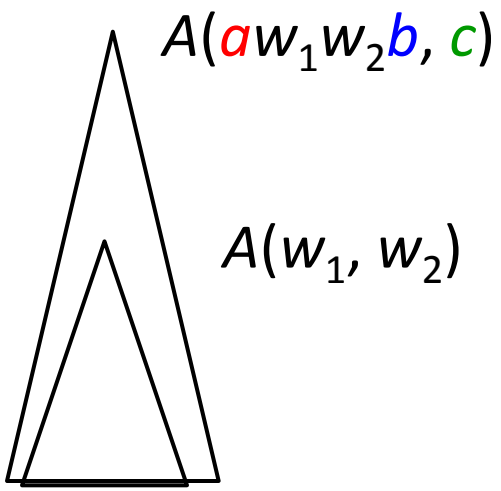
$$u_1 v_1^{\,i} w_1 s_1^{\,i} u_2 v_2^{\,i} w_2 s_2^{\,i} u_3 \in L$$

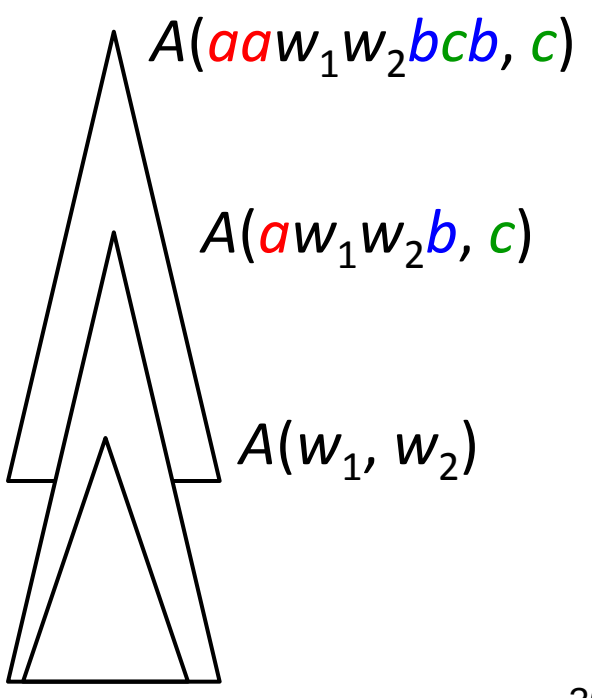




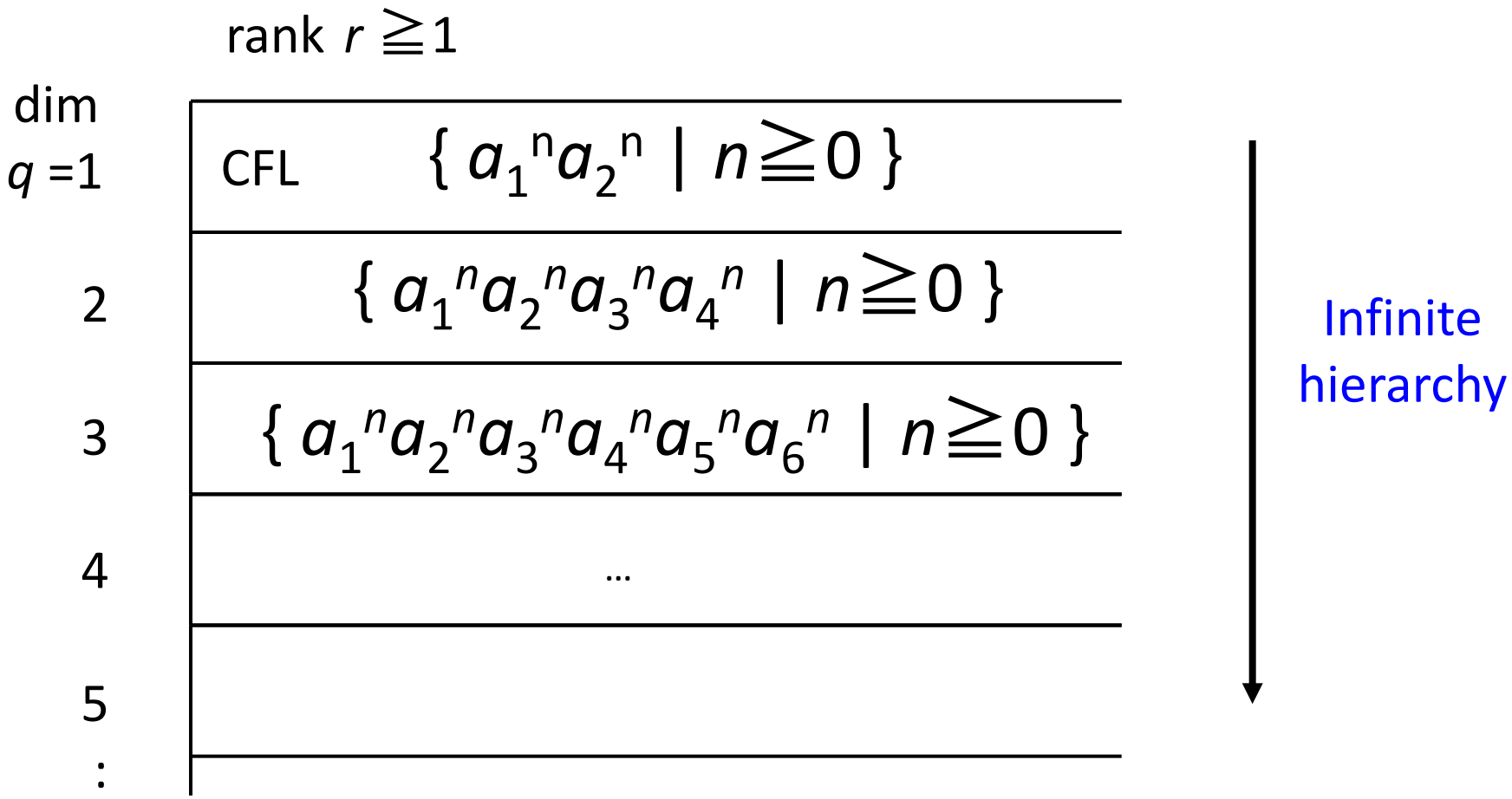
Pumping lemma

$u_1 v_1^{\,i} w_1 s_1^{\,i} u_2 v_2^{\,i} w_2 s_2^{\,i} u_3 \in L$??

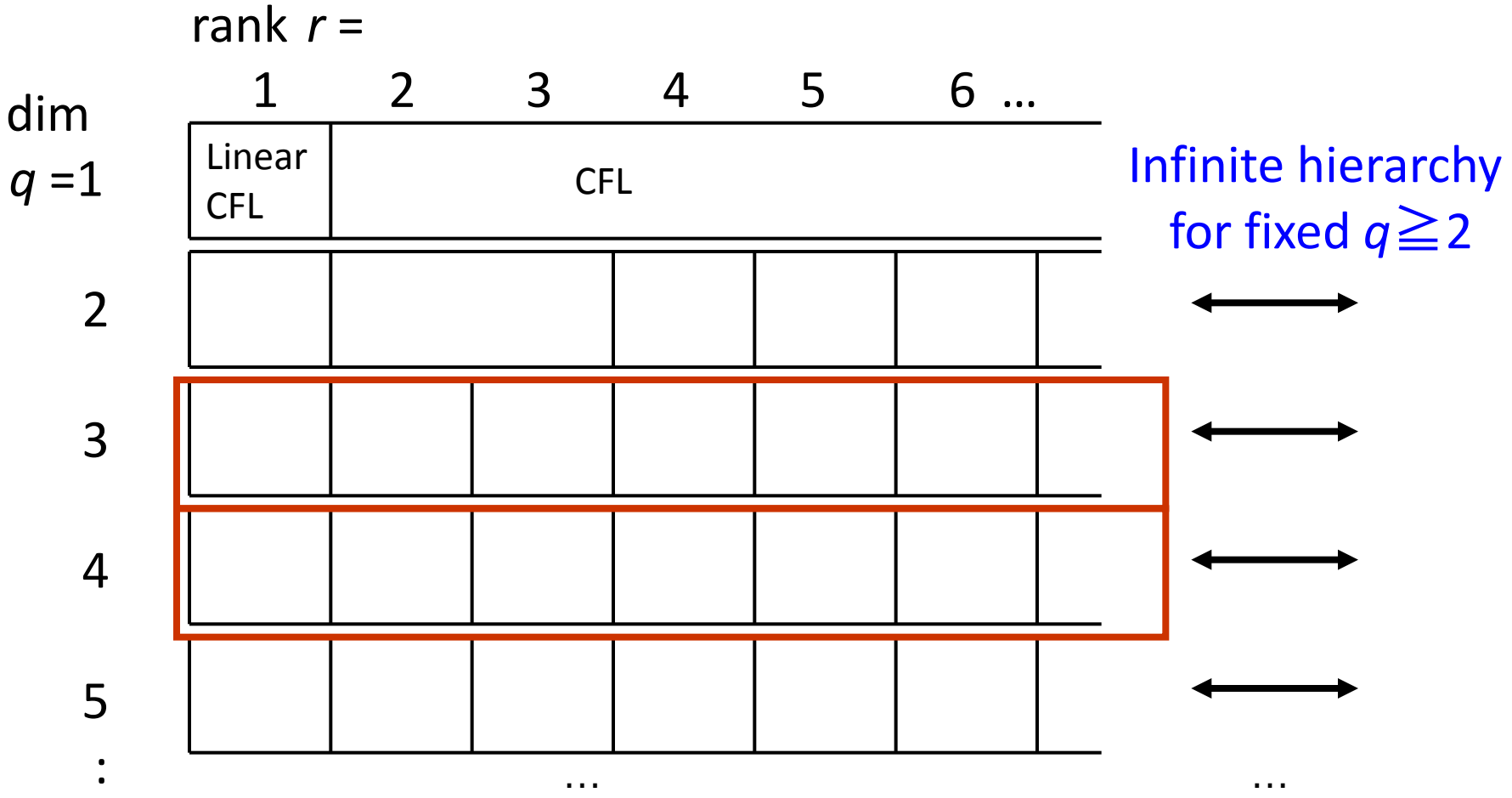




Hierarchy on dimensions

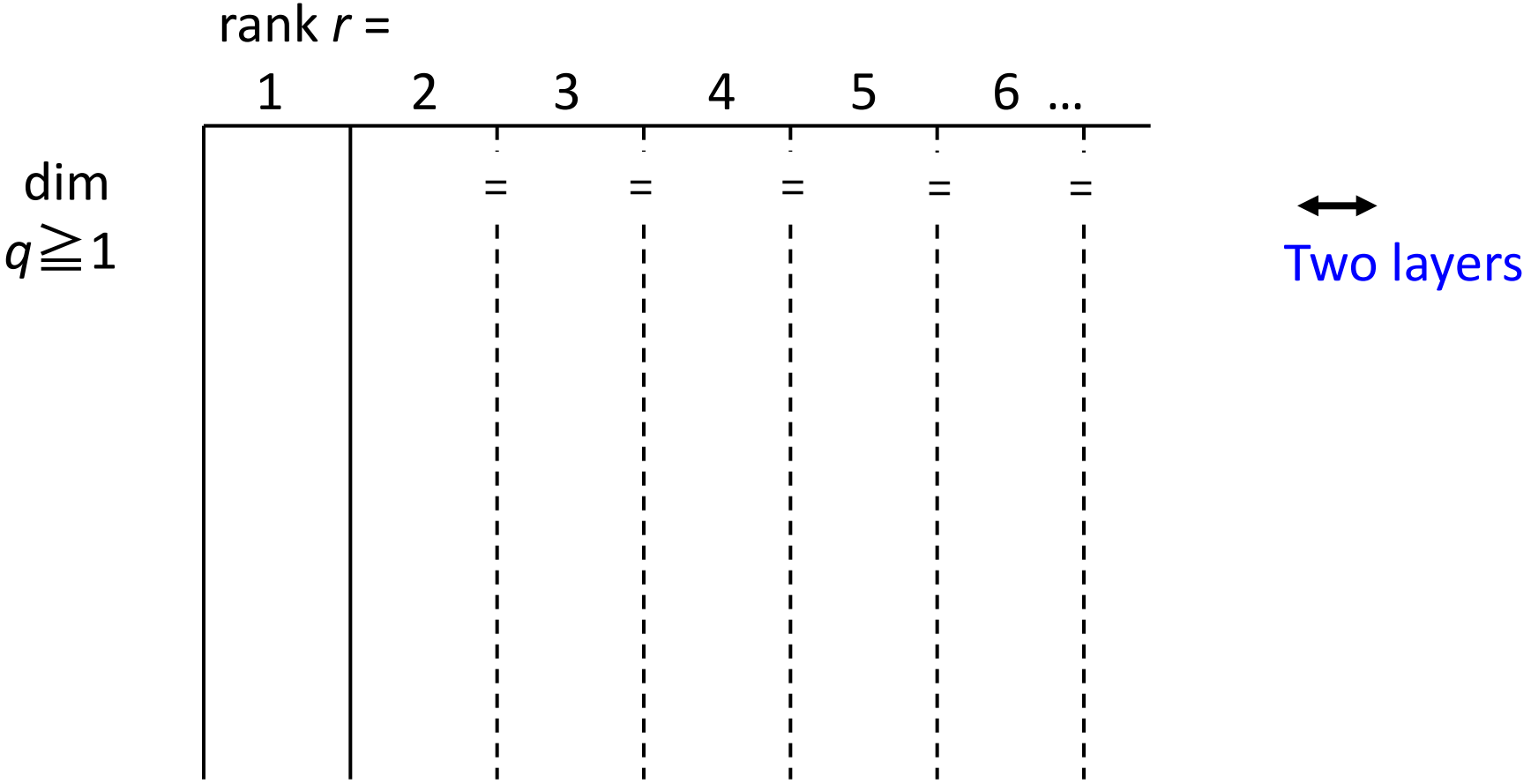


Hierarchy on ranks for fixed dim [RS99]



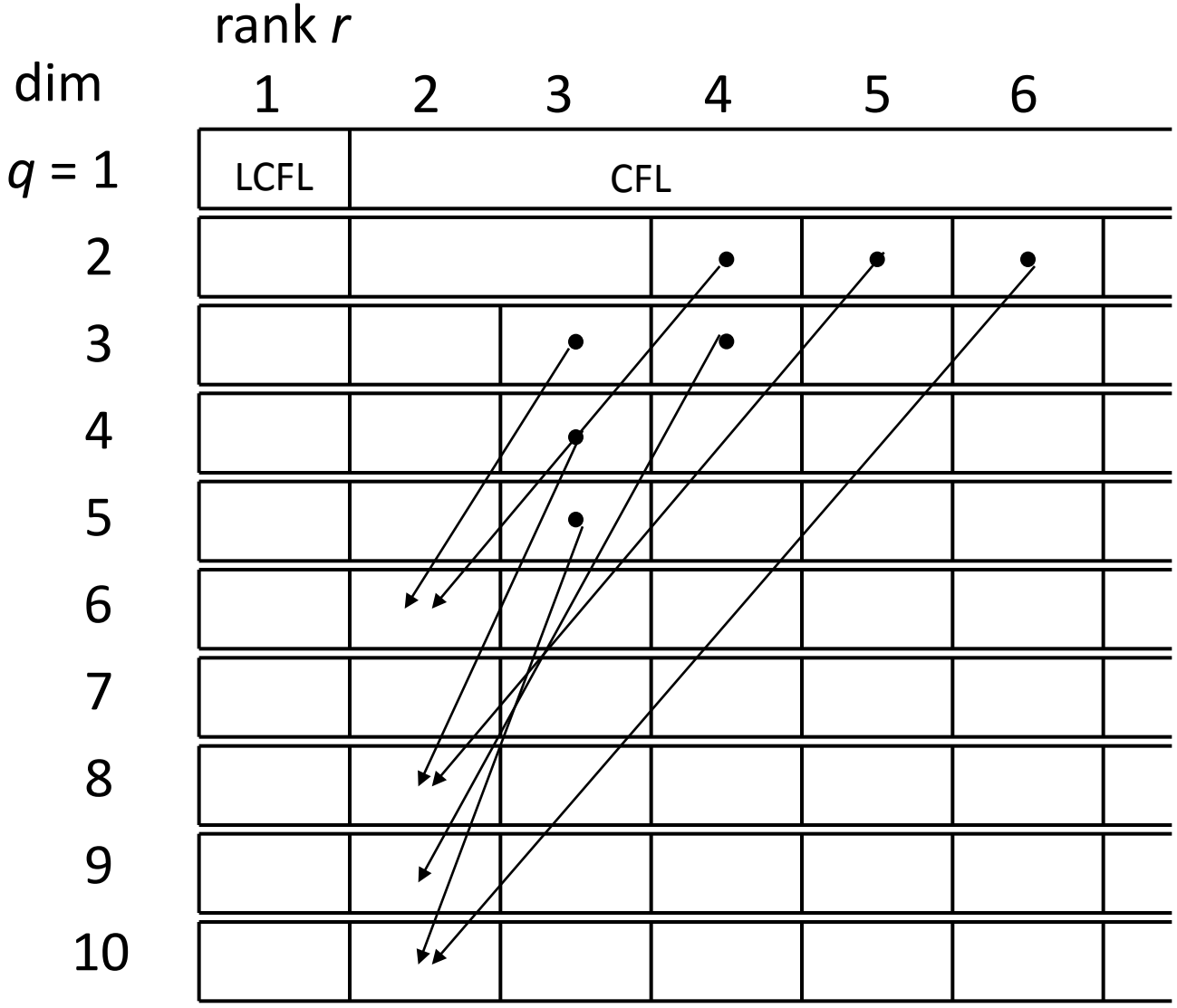
[RS99] Rambow and Satta, TCS 223, 1999.

Hierarchy on ranks [RS99][Matsumura89]



MCFL=MCFL(2) ⇒ It suffices to consider
MCFL(1) and MCFL(2) as far as ranks are concerned.

Trade-offs [RS99]



q-MCFL(*r*) ⊆
(*k*+1)*q*-MCFL(*r*-*k*)

Corollary:
q-MCFL(*r*) ⊆
(*r*-1)*q*-MCFL(2)

Beyond MCFG

$A(x, x)$:- $B(x)$, $C(x)$.

$A(x)$:- $B(x)$, $C(x)$.

Closed under intersection

$A(x, ax)$:- $B(x)$.

$B(x_1y_1, x_2y_2)$:- $C(x_1, x_2)$, $D(y_1, y_2)$.

MCFG ⊂ PMCFG ⊂ simpleLMG [Groenink95]

MCFL ⊊ PMCFL ⊊ simpleLML = P

$\{ a^{2^n} \mid n \geqq 0\} \in$ PMCFL – MCFL

$S(xx)$:- $S(x)$, $S(a)$.

Other topics on MCFG (Also refer to the other talks.)

- Parsing
  - Earley type parser (Matsumura+89), (Kanazawa 08)
  - 'Unambiguous' MCFG (O(n^2) time recognizable) (Nakanishi92)
  - LL(k) MCFG (Nii96)
  - Stochastic CYK parser (Kato+06)
- An extension of Chomsky-Schuzenberger theorem for CFL (Kaji+91, Yoshinaka+10)

Fixed vs. universal recognition

Recognition problem for *fixed* language L (FRP)

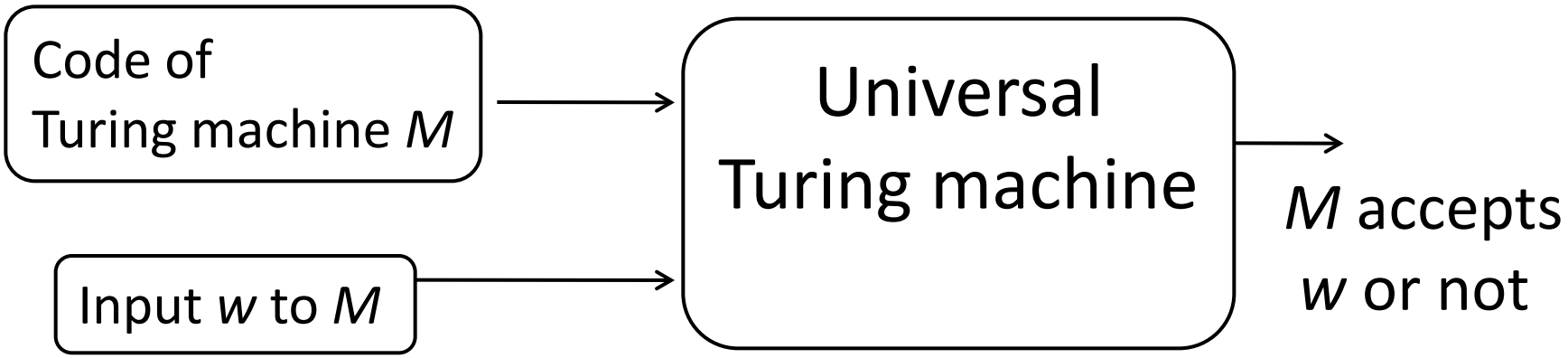
Input: string w

Problem: $w \in L$?

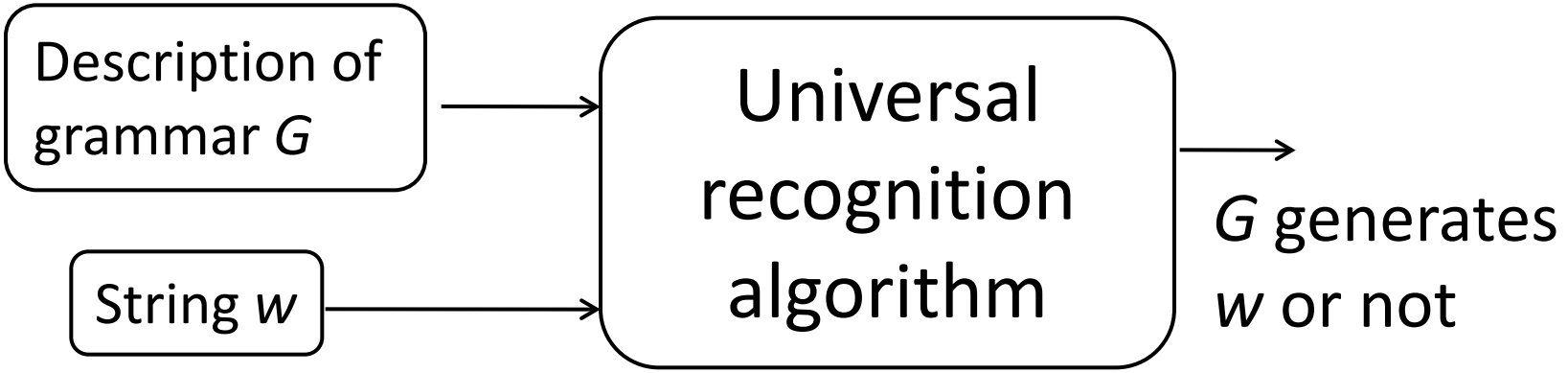
Universal recognition problem (URP)

Input: grammar G, string w

Problem: $w \in L(G)$?


Code of
Turing machine *M*
Input *w* to *M*
Universal
Turing machine
M accepts
w or not

URP


Description of grammar *G*
String *w*
Universal recognition algorithm
G generates *w* or not

Example

Recognition problem for *fixed* language L:

For a string w, decide $w \in L$.

$O(|w|^3)$ time when L is CFL.

Universal recognition problem (URP):

For a grammar G & string w, decide $w \in L(G)$.

$O(|G|\ |w|^3)$ time when G is CFG.

DEXP time-hard when G is GPSG†.

† GPSG (Generalized Phrase Structure Grammar[GKPS85])

GPSG = CFG in generative power

⇒ Generative power ≠ Complexity of URP

FRP vs. URP

- Generative power can be measured by

  Complexity of FRP
- Generative power ≠ Complexity of URP

  Rather, complexity of URP for grammar class *C*

  ∼ Succinctness of *C*

Grammar (especially MCFG) as computation device with bounded resource

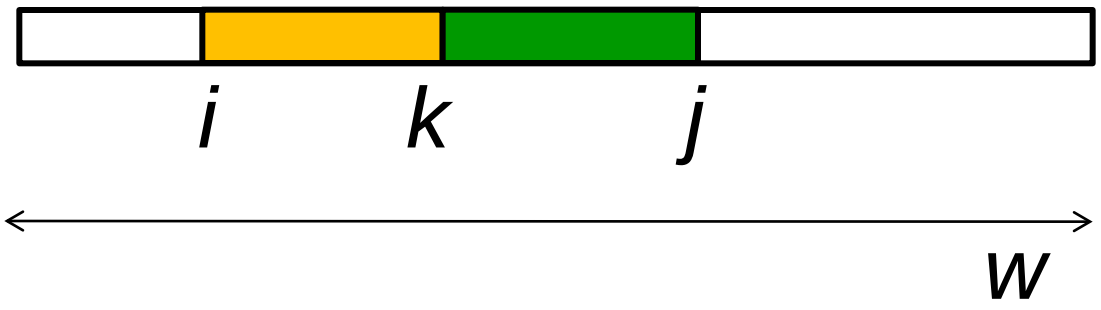
Let's start with FRP.

CYK algorithm for CFG

If rule: $A(xy)$:- $B(x)$, $C(y)$ ($A \rightarrow BC$)

$B \in \text{table}(i, k) \wedge C \in \text{table}(k, j)$ for $\exists\ k\ (i \leqq u < j)$

then add A to table(i, j) ;



Time complexity: $O(n^3)$ n = length of input w

CYK algorithm for MCFG

If rule: $A(y_1z_1, z_2y_2)$:- $B(y_1, y_2)$, $C(z_1, z_2)$,

$B \in \text{table}(i, u, v{+}1, l) \wedge C \in \text{table}(u{+}1, j, k, v)$

for $\exists\ u, v\ (i \leqq u < j,\ k \leqq v < l)$

then add A to table(i, j, k, l) ;

- *Time complexity:* $O(n^{\deg(G)})$ $(\deg(G) \leqq (r{+}1)q)$

  ($\deg(G)$=6 in this example)

URP for MCFG

- General case — DEXP-complete
- Non-deleting MCFG — PSPACE-complete
- q-MCFG with fixed q — NP-complete
- q-MCFG(r) with fixed q, r — P-complete

General case

- DEXP-complete
  - Upper bound: Simulation of MCFG by DEXP time bounded Turing machine.
  - Lower bound: Simulation of PSPACE bounded *alternating* Turing machine (APSPACE) by MCFG.

    Since general MCFG can use deleting rules, it can simulate computation for a universal state (of alternating TM) with poly-length ‘sentential form.’

Alternating Turing machine

ATM is $M = (Q, \Sigma, \Gamma, B, \delta, q_s, Q_F, Q_U, Q_E)$

$Q = Q_U \cup Q_E \cup Q_F \cup \{q_s\}$: state set

Σ ($\subseteq \Gamma$ - {B}): input symbols

Γ: tape symbols B: blank symbol

$\delta \subseteq (Q \times \Gamma) \times (Q \times \Gamma \times \{\rightarrow, \leftarrow\})$: transition relation

q_s: initial state Q_F: final (accepting) states

Q_U: set of *universal* states

Q_E: set of *existential* states

Alternating Turing machine

ATM $M = (Q, \Sigma, \Gamma, B, \delta, q_s, Q_F, Q_U, Q_E)$

ID (instantaneous description) is $\alpha q \beta$ where $q \in Q$ & $\alpha, \beta \in \Gamma^*$.

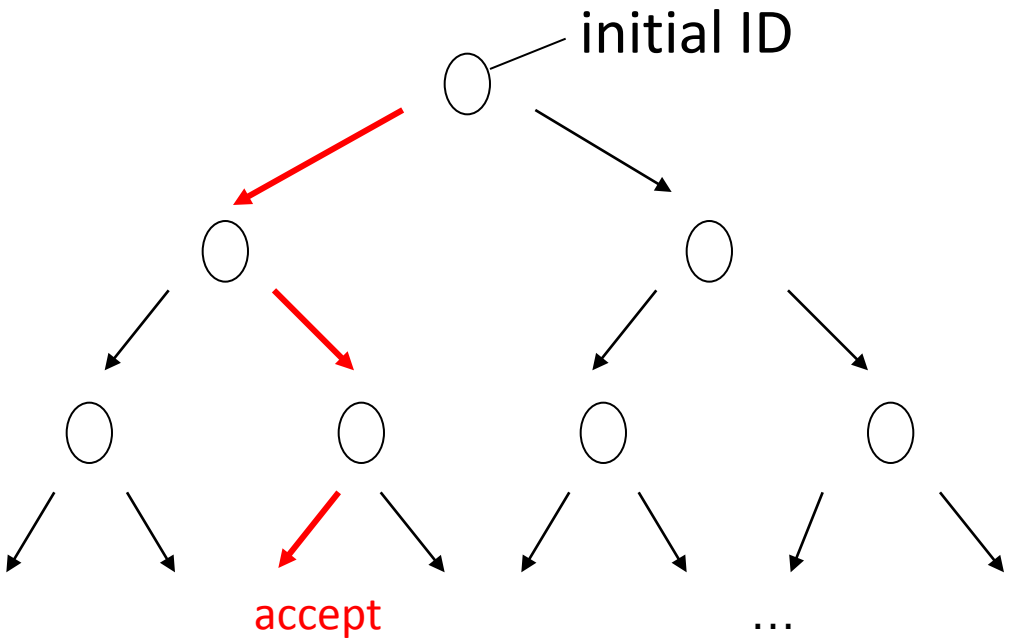
Move relation $\alpha q \beta \rightarrow^*_M \alpha' q' \beta'$ over IDs is defined in a usual way.

For an input w, ID $q_s w$ is the <u>initial</u> ID for w.

ID $\alpha q \beta$ is <u>accepting</u> if $q \in Q_F$.

Nondeterministic Turing machine

Input *w* is accepted if there is a run that reaches an accepting ID from the initial ID for *w*.

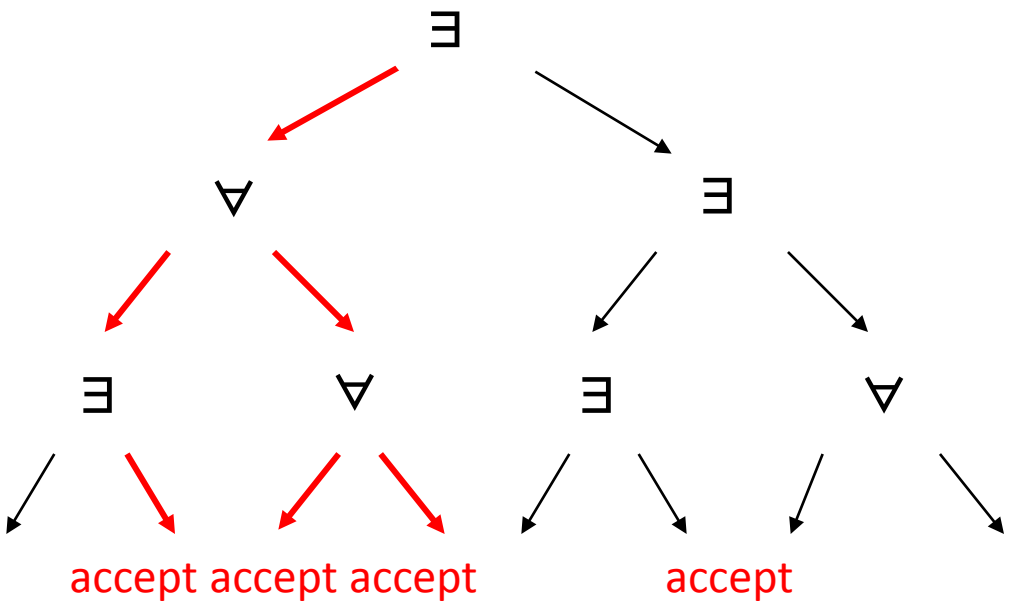


Alternating Turing machine

For $q \in Q_F$, ID $\alpha q \beta$ (trivially) is accepting.

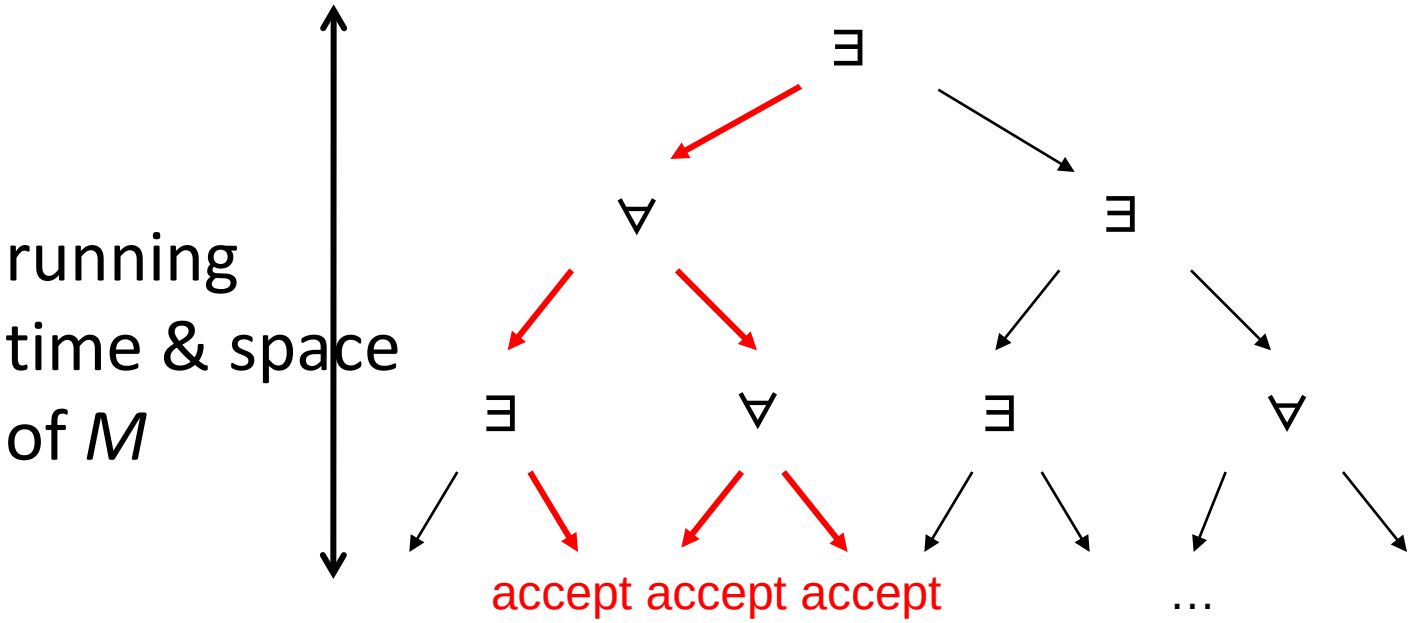
For $q \in Q_E$, ID $\alpha q \beta$ is accepting if *there is* γ such that $\alpha q \beta \rightarrow^*_M \gamma$ and γ is accepting.

For $q \in Q_U$, ID $\alpha q \beta$ is accepting if *every* γ satisfying $\alpha q \beta \rightarrow^*_M \gamma$ is accepting.



Accepting language

$L(M) = \{ w \mid$ Initial ID $q_s w$ is accepting $\}$



TM and ATM

Propositon[CS80]:

PSPACE=APTIME, DEXPTIME=APSPACE

APTIME := class of problems solvable by poly time-bounded ATM

APSPACE := class of problems solvable by poly space-bounded ATM

Proof of DEXP completeness

For given $p(n)$-space bounded ATM $M = (Q, \Sigma, \Gamma, B, \delta, q_s, Q_F, Q_U, Q_E)$ and $w \in \Sigma^*$,

construct MCFG $G\ (N, \{1\}, V, P, S)$ such that

M accepts w if and only if $\varepsilon \in L(G)$.

APSPACE $\leqq$ URP MCFG

- Nonterminals $A[q, k]$ ($q \in Q$, $k \in [1..p(n)]$)

$$\dim(\ .\ ) = p(n)\ |\Gamma|$$

$$A[q, k] \left(\begin{array}{ccccc} 1 & \ldots & j & \ldots & p(n) \\ & & \varepsilon & & \end{array} \right) c$$

derivable intuitively means that

$\exists\, \alpha, \beta$: $\alpha q \beta$ is accepting,

$k = |\alpha| + 1$, the j-th symbol of $\alpha\beta$ is c.

Construction

(0 step acceptance)

$\forall q_f \in Q_F, \forall\ k \in [1..p(n)]$

$A[q_f, k](\varepsilon, \ldots, \varepsilon)$.

(Initial ID)

$\forall q \in Q, \forall\ k \in [1..p(n)]$

$S(x_{<1,a1>} x_{<2,a2>} \ldots x_{<n,an>} x_{<n+1,B>} \ldots x_{<p(n),B>})$

:- $A[q, k](\ldots)$

for given input $w = a_1 a_2 \ldots a_n$.

Construction (existential state)

- $q \in Q_E$, $(b, p, \rightarrow) \in \delta(q, a)$ (move right)

$A[q,2]\ (x_{<1,a>}, x_{<2,b>}, x_{<3,a>},$

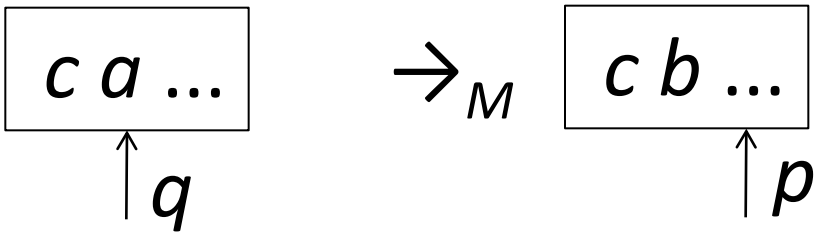
$x_{<1,b>}, 1, x_{<3,b>},$

$x_{<1,c>}, 1, x_{<3,c>})$:-

$A[p, 3]\ (x_{<1,a>}, x_{<2,a>}, x_{<3,a>},$

$x_{<1,b>}, x_{<2,b>}, x_{<3,b>},$

$x_{<1,c>}, x_{<2,c>}, x_{<3,c>})$



Construction (universal state)

- $q \in Q_U$, , $(b, p, \rightarrow) \in \delta(q, a)$, $(c, r, \leftarrow) \in \delta(q, a)$

$A[q,2]$ ($x_{<1,a>}\ y_{<1,a>}$, $x_{<2,b>}\ y_{<2,c>}$, $x_{<3,a>}\ y_{<3,a>}$,

$x_{<1,b>}\ y_{<1,b>}$, 1 , $x_{<3,b>}\ y_{<3,b>}$,

$x_{<1,c>}\ y_{<1,c>}$, 1 , $x_{<3,c>}\ y_{<3,c>}$) :-

$A[p\ , 3]$ ($x_{<1,a>}$, $x_{<2,a>}$, $x_{<3,a>}$,

$x_{<1,b>}$, $x_{<2,b>}$, $x_{<3,b>}$,

$x_{<1,c>}$, $x_{<2,c>}$, $x_{<3,c>}$),

$A[r, 3]$ ($y_{<1,a>}$, $y_{<2,a>}$, $y_{<3,a>}$,

$y_{<1,b>}$, $y_{<2,b>}$, $y_{<3,b>}$,

$y_{<1,c>}$, $y_{<2,c>}$, $y_{<3,c>}$),

Recent result

Theorem: URP for poly depth-bounded MCFG is PSPACE-complete.

Corollary: URP for poly depth-bounded MCFG(1) is NP-complete.

Summary

non-deletion	dimension	depth	Universal recognition	
-	-	-	DEXP-complete	[KNSK94]
required	-	-	PSPACE-complete	
-	bounded	-	NP-complete	
-	-	poly	PSPACE-complete	[Seki10]
-	-	poly, (*i*-1)-bounded alternation	Σ_i/π_i-complete	
-	-	poly, rank 1	NP-complete	

Appendix

Tadao Kasami Wins the 1999 Claude E. Shannon Award

The Information Theory Society's highest honor, the Claude E. Shannon Award, is awarded annually to an individual who has achieved consistent and profound contributions to the field of information theory. The recipient is chosen by a selection committee consisting of Society officers and two former Shannon Award recipients.

Prof. Tadao Kasami of Hiroshima City University, Japan, has been selected as the 1999 Claude E. Shannon Award recipient. The award was announced at the 1998 International Symposium on Information Theory and will be presented to Prof. Kasami at the 2000 International Symposium on Information Theory.

Tadao Kasami

Hideki Imai held the following interview with Tadao Kasami in honor of his receipt of the Award.

Interview with Tadao Kasami

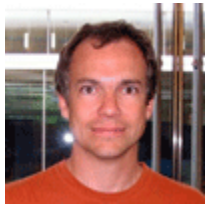
Hideki: Congratulations on winning 1999 Claude E.Shannon Award. You certainly deserve this award on the basis of your outstanding contributions in coding theory. I know that you started your career as a re-

theory in which there remained only very hard long-standing problems. Almost everyday, I scanned papers in IRE Transactions and so on at the library. For instance, I found a paper [2] by N. Abramson and read it with a strong interest. There continued a rush of epoch-making papers by Bose-Chaudhuri, Peterson and so on. I had to study hard in order to catch up with the progress of coding theory. Finally, I finished my Ph. D. thesis (1963) based on published papers [3-8].

Hideki: How did you meet Prof. W. Wesley Peterson and what did you learn from him?

Tadao: I exchanged letters with Prof. Peterson (Wes) who had been an associate editor of IEEE Trans. on Information Theory. I first had the chance to meet him in 1963 when he visited Japan to attend a URSI meeting in Tokyo. In November 1964, my sincere hope to study under Wes came true by his kind arrangements. I was deeply impressed with his profound knowledge which ran extensively from quantum communication to software engineering. He encouraged me to continue my study [9] on a syntax analysis algorithm for context-free grammar. Wes advised me to select a research subject of main interest in the field. When

- Original paper on CYK algorithm (July, 1965).
- Dr. S. Eddy, a system biologist, Washington U. asked Dr. Kasami to send a copy (Apr. 2002).



- Dr. Kasami searched his house for a week, found one at last and sent it to Dr. Eddy.

AFCRL-65-558

AN EFFICIENT RECOGNITION AND SYNTAX-ANALYSIS ALGORISM FOR CONTEXT-FREE LANGUAGES

TADAO KASAMI

CONTRACT NO. AF19(628)-4379
PROJECT NO. 5628
TASK NO. 562801

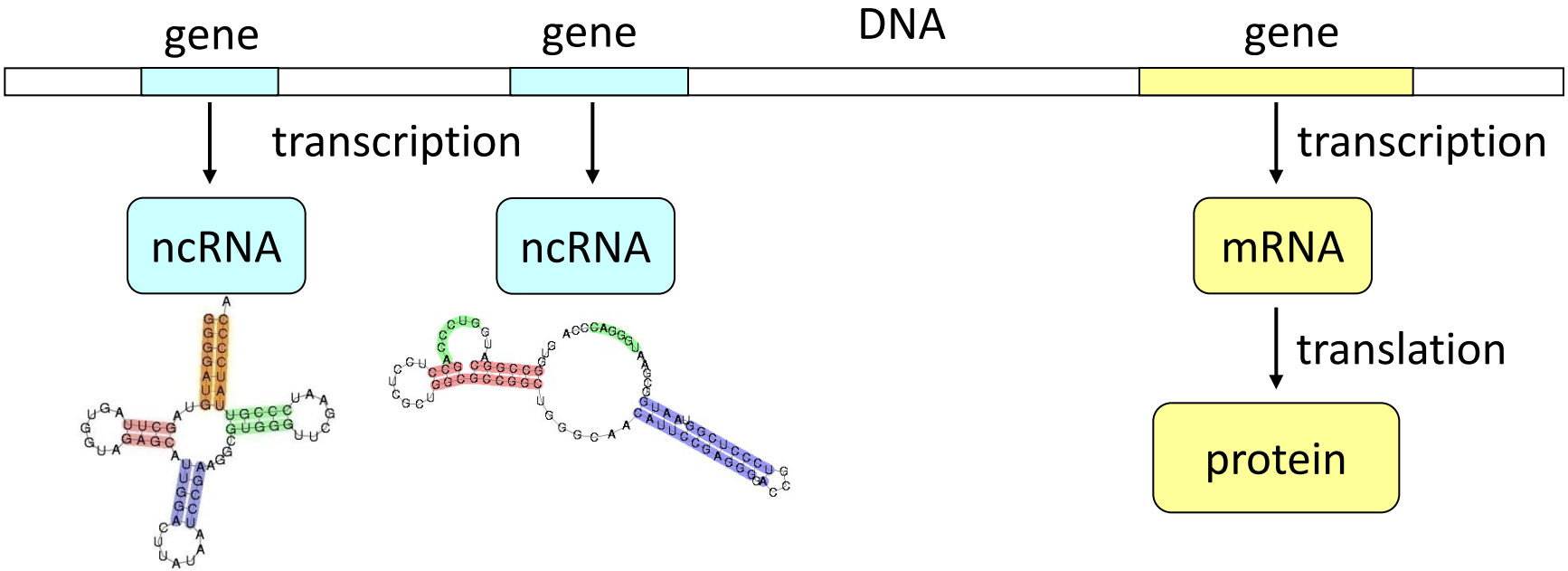
SCIENTIFIC REPORT NO. 2

JULY 11, 1965

PREPARED FOR
AIR FORCE CAMBRIDGE RESEARCH LABS.
OFFICE OF AEROSPACE RESEARCH
UNITED STATES AIR FORCE
BEDFORD, MASSACHUSETTS

DEPARTMENT OF
ELECTRICAL ENGINEERING
UNIVERSITY OF HAWAII
HONOLULU, HAWAII

noncoding RNA



- ncRNA (noncoding RNA)
  - RNA not translated to protein, some of them play important roles in bio-chemical reactions (functional RNA).

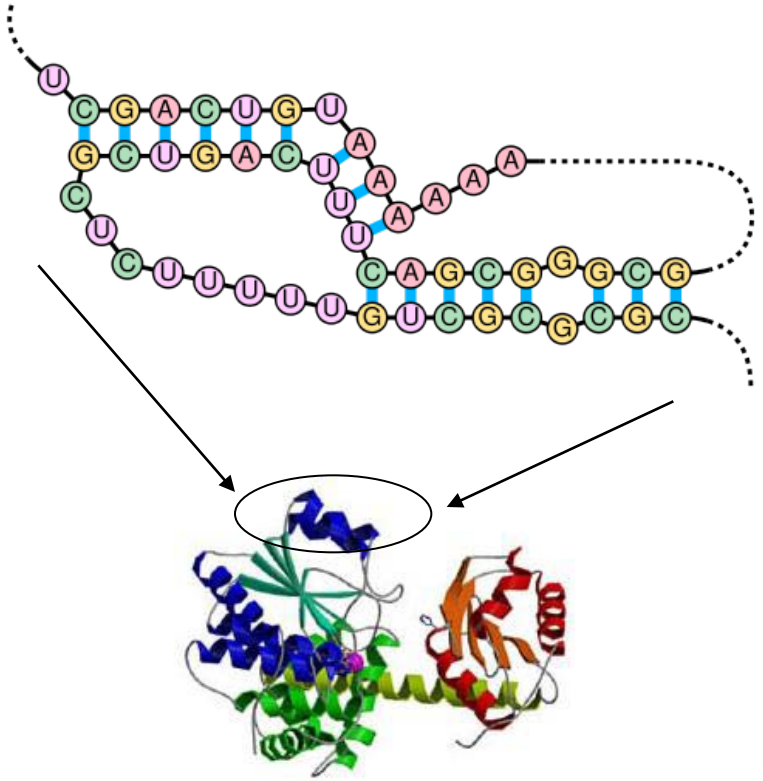
Structures of ncRNA

- Primary structure: sequence of bases A, U, C, G

  C U U C A U C A G A A A A U G A C

  (easy to obtain by next-generation sequencer)
- Secondary structure:

  folding structure by hydrogen bonding between bases
- Ternary (3D) structure

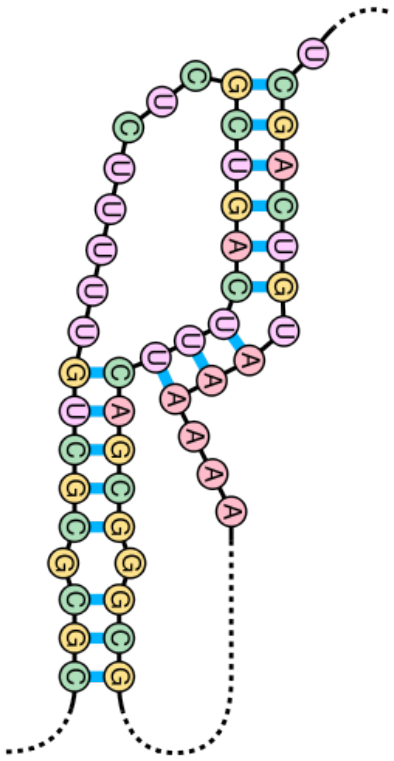


RNA secondary structure prediction

Co-relation between

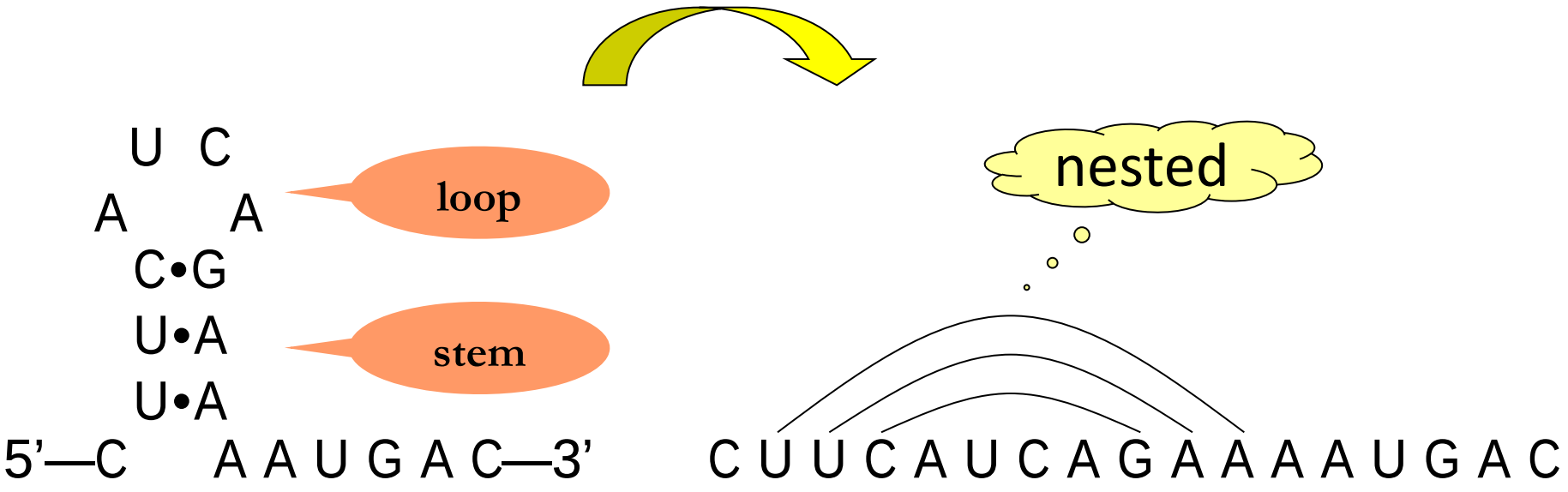
structure & function

⇒ Genomic drug discovery, etc.



Problem: Given an RNA primary sequence, predict its secondary structure

Stem-loop structure



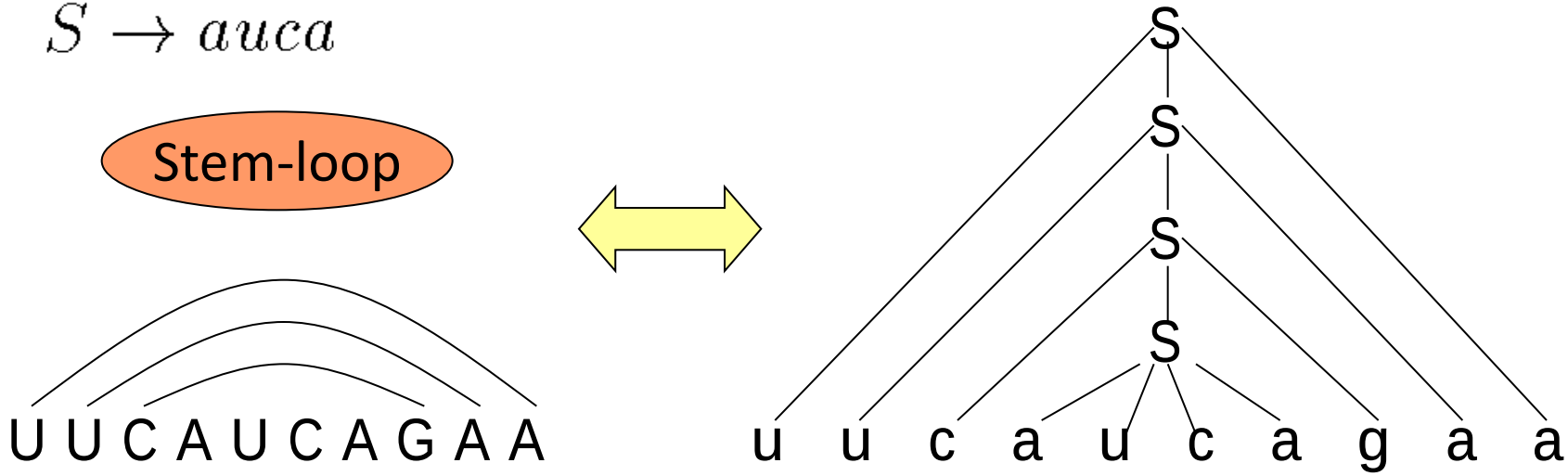
Secondary structure and grammar

- Modeling biological regularity by a grammar *G*
- Secondary structure prediction

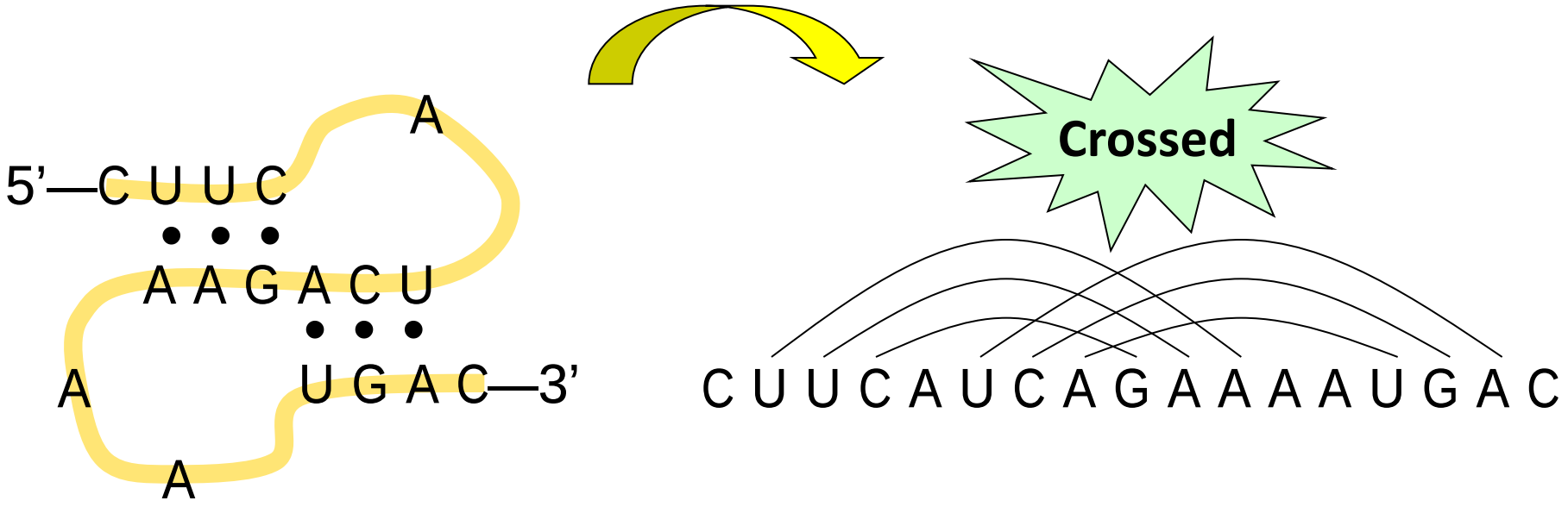
  = Parsing primary sequence with *G*

$$S \rightarrow aSu \mid uSa \mid cSg \mid gSc$$

$$S \rightarrow auca$$



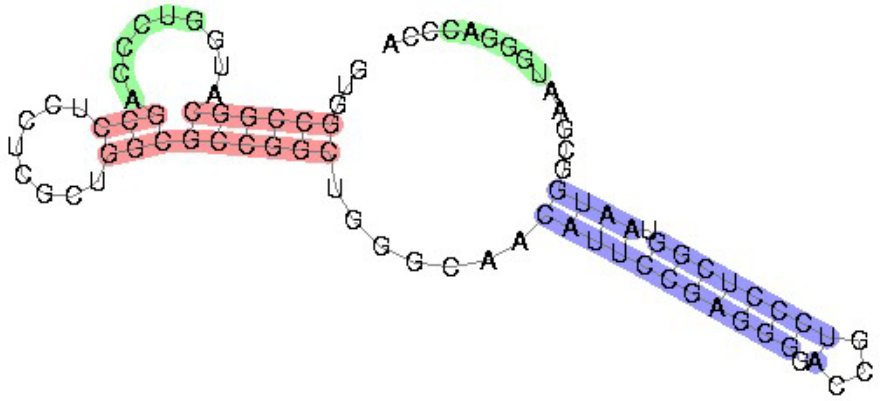
Pseudoknot



- CFG cannot represent **pseudoknot.**

HDV_ribozyme

- Hepatitis delta virus ribozyme
- length: 87–91



Existing methods using comparative approach for pseudoknots

- **hxmatch** [Witwer+04]
- ILM [Ruan+04]
- Simulfold [Meyer&Miklós07]

Not grammar based

- **Pair-SMCFG** [Mizoguchi+09]
- **Proposed method** [Mizoguchi+11]

[Witwer+04] Witwer , Hofacker & Stadler, IEEE Trans. Computational Biology and Bioinformatics, 2(2), 2004
[Meyer&Miklós07] Meyer & Miklós, PLoS Computational Biology, 3(8), 2007
[Ruan+04] Ruan, Stormo & Zhang, Bioinformatics 20(1), 2004.
[Mizoguchi+09] Mizoguchi, Kato & Seki, 20th International Conference on Genome Informatics poster, 2009
[Mizoguchi+11] Mizoguchi, Kato & Seki, A grammar-based approach to RNA pseudoknotted structure prediction for aligned sequences, *IEEE 2011 ICCABS*.

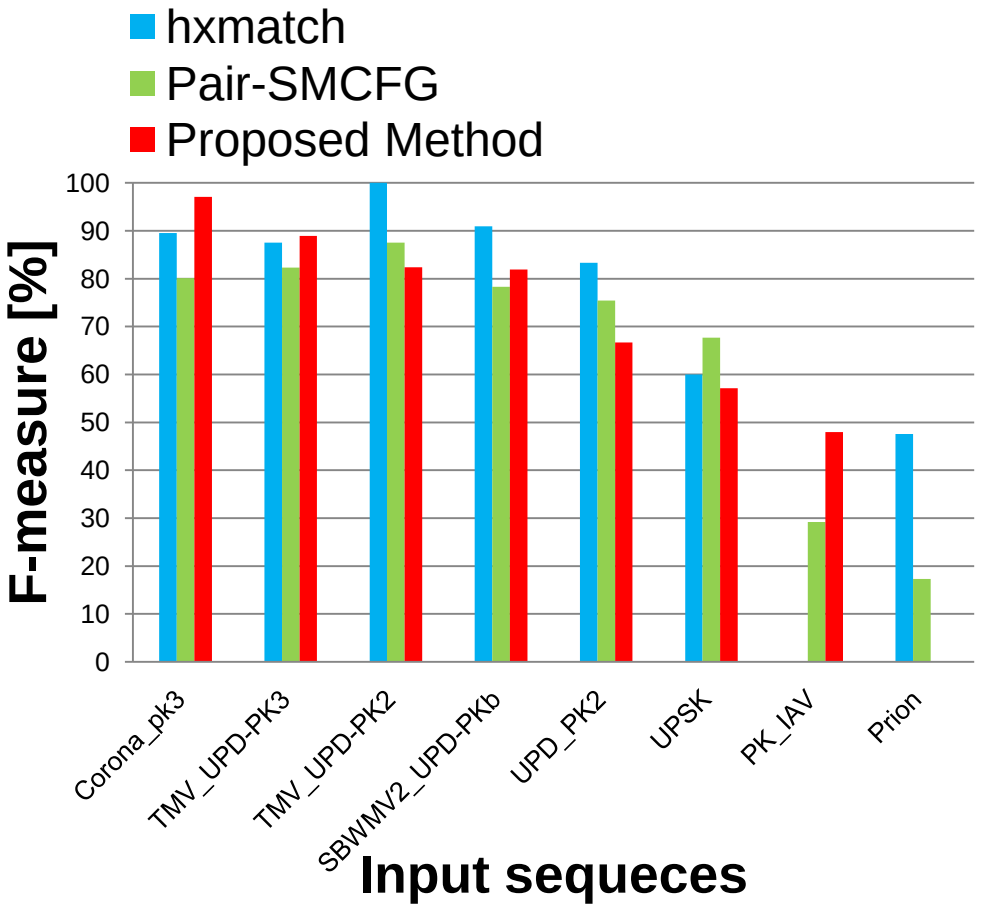
Prediction accuracy

Data set:

8 families from Rfam database [Griffiths-Jones+05].

Average F-measure:

- **hxmatch:** 69.85 %
- **Pair-SMCFG:** 64.73 %
- **Proposed Method:** 65.26 %



[Griffiths-Jones+05] Griffiths-Jones, Moxon, Marshall, Khanna, Eddy & Bateman, Nucleic Acids Res., 33, 2005

Thank you!