Distributional Learning and Context/Substructure Enumerability in Nonlinear Tree Grammars

Makoto Kanazawa, National Institute of Informatics & SOKENDAI
Ryo Yoshinaka, Kyoto University

1

Generalizing Distributional Learning

MCFGs
Yoshinaka 2010

→ **Second-Order ACGs**
Yoshinaka & Kanazawa 2011

linear

nonlinear

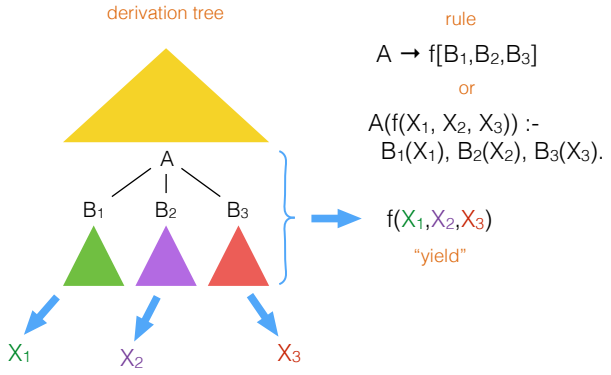
PMCFGs
Clark & Yoshinaka 2014

→ **Subclasses of Almost Linear Second-Order ACGs on Trees**
This talk

2

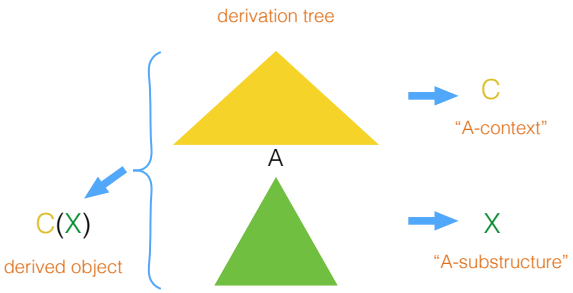
Almost linear second-order ACGs with bounded degree of nonlinearity either on the context side or the substructure side.
Will use the next 16 slides to give the background.

"Context-Free" Grammar Formalisms



3

Contexts and Substructures



4

Extraction of a subtree from a derivation tree induces a decomposition of the derived object into a context and a substructure.

Possible Contexts and Substructures

$\mathbb{G}$: class of grammars

$\mathbb{C}$: set of possible contexts for grammars in $\mathbb{G}$

$\mathbb{S}$: set of possible substructures for grammars in $\mathbb{G}$

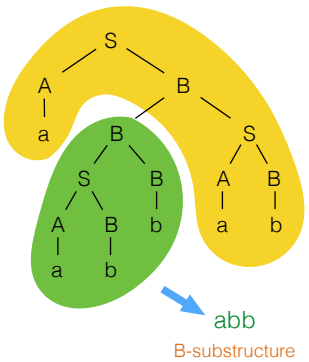
Each grammar G in $\mathbb{G}$ determines a relation:

$$\{ (C, X) \in \mathbb{C} \times \mathbb{S} \mid C(X) \in L(G) \}$$

Note: $C(X) \in L(G)$ does not imply G has a nonterminal A such that C is an A-context and X is an A-substructure.

5

Example: CFG in CNF



$S \to AB \mid SS$
$A \to a$
$B \to b \mid SB$

$x \mapsto a\ x\ ab$
B-context

abb
B-substructure

$\mathbb{G}$ = "CFG in CNF"

$\mathbb{C} \cong \Sigma^* \times \Sigma^*$

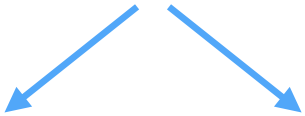
$\mathbb{S} = \Sigma^+$

6

A context is a function that takes a string and wraps two strings around it.

Distributional Learning

Hypothesis space:
"tractable" grammar class $\mathbb{G}$



Dual approach:
target grammars in $\mathbb{G}$ with
m-finite context property

Primal approach:
target grammars in $\mathbb{G}$ with
m-finite kernel property

7

"Tractable" means certain parameters of grammars are fixed.

Finite Context Property

- G has the **m-FCP** ⇔ for each nonterminal A of G, there is a finite set $\mathbf{C}_A$ of A-contexts such that
  - for all $X \in \mathbb{S}$,
    $C(X) \in L(G)$ for all $C \in \mathbf{C}_A \Rightarrow X$ is an A-substructure
  - $|\mathbf{C}_A| \leq m$
- As a consequence, for each rule $A \rightarrow f[B_1,\ldots,B_n]$ of G and for all $X_1,\ldots,X_n \in \mathbb{S}$,

  $\bigwedge_{1\leq i\leq n} (C(X_i) \in L(G)$ for all $C \in \mathbf{C}_{B_i}) \Rightarrow$
  $C(f(X_1,\ldots,X_n)) \in L(G)$ for all $C \in \mathbf{C}_A$

8

Let's look at the dual approach, the simpler of the two approaches.

CFG with 1-FCP

S → AB | SS
A → a | AS | SA
B → b | SB | BS

$\mathbf{C}_S = \{ (\varepsilon, \varepsilon) \}$
$\mathbf{C}_A = \{ (\varepsilon, b) \}$
$\mathbf{C}_B = \{ (a, \varepsilon) \}$

x is an S-substructure ⇔ nf(x) = ε
x is an A-substructure ⇔ nf(x) = a
x is a B-substructure ⇔ nf(x) = b

ab ▻ ε

9

Each rule is valid under the given assignment of context sets to nonterminals.
(Removing some rules breaks 1-FCP, but validity of each rule is intact as long as the language stays the same.)

Distributional Learning Algorithm (Dual Approach)

D: finite positive data (received up to some point)

$\mathbb{C}|_D = \{ C \in \mathbb{C} \mid C(X) \in D \text{ for some } X \in \mathbb{S} \}$ observed contexts

Construct rules

$$\mathbf{C}_0 \rightarrow f[\mathbf{C}_1,\ldots,\mathbf{C}_n]$$

such that

- $\mathbf{C}_i$ is a nonempty finite subset of $\mathbb{C}|_D$ with $|\mathbf{C}_i| \leq m$
- f is a function "observed" in D

10

D is the positive data received up to the last moment when the hypothesis grammar was found to undergenerate.

Distributional Learning Algorithm (Dual Approach)

$E \supseteq D$: all positive data received so far

$\mathbb{S}|_E = \{ X \in \mathbb{S} \mid C(X) \in E \text{ for some } C \in \mathbb{C} \}$ observed substructures

Hypothesize grammar using those rules

$$\mathbf{C}_0 \rightarrow f[\mathbf{C}_1,\ldots,\mathbf{C}_n]$$

constructed from D that are **valid for** $\mathbb{S}|_E$:

for all $X_1,\ldots,X_n \in \mathbb{S}|_E$,

$\bigwedge_{1\leq i\leq n} (C(X_i) \in L(G) \text{ for all } C \in \mathbf{C}_i) \Rightarrow$

$C(f(X_1,\ldots,X_n)) \in L(G) \text{ for all } C \in \mathbf{C}_0$

use membership oracle

11

Membership queries are restricted to combinations of bits of observed positive data. (Not quite?)

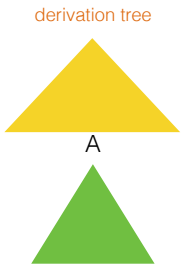
Context/Substructure Enumerability

- A grammar class $\mathbb{G}$ is
  - **context-enumerable** if $\mathbb{C}|_D$ can be enumerated in polynomial time in the size of D
  - **substructure-enumerable** if $\mathbb{S}|_D$ can be enumerated in polynomial time in the size of D
- Both are needed to achieve polynomial update time of distributional learning.

12

Clearly satisfied by $\mathbb{G}$ = "CFG in CNF".

MCFGs (of bounded dimension and rank)



$(x_1,\ldots,x_r) \mapsto v_0\, x_1\, v_1 \ldots x_r\, v_r$

regular r-variable pattern ≅
(r+1)-tuple of strings

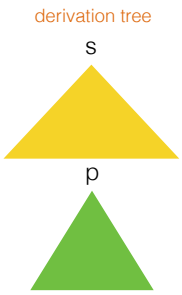
$(w_1,\ldots,w_r) \in (\Sigma^*)^r$

r-tuple of strings

context-enumerable	✓
substructure-enumerable	✓

13

Second-Order (Linear) ACGs



N

closed linear λ-term
of type $\mathcal{H}(p) \to \mathcal{H}(s)$

M

closed linear λ-term of type $\mathcal{H}(p)$

context-enumerable	✓
substructure-enumerable	✓

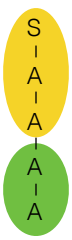
14

With a similar bound on certain parameters (possible object realizations of abstract types of constants limited to a fixed finite set). The type $\mathcal{H}(p)$ is taken from a fixed finite set.

k-copying PMCFGs

$S(x_1\, x_2\, a) \text{ :- } A(x_1,\, x_2)$
$A(x_1\, x_2\, a,\, x_2\, a\, a) \text{ :- } A(x_1,\, x_2)$ 2-copying
$A(\varepsilon, \varepsilon) \text{ :-}$

$(x_1,x_2) \mapsto$
$x_1x_2ax_2aaax_2aaaaa$
3-copying context



aaaaaaaaaaaaaaa
(aaaaaaaaaa, aaaaaa)
(aaaa, aaaa)
(a, aa)
(ε, ε)

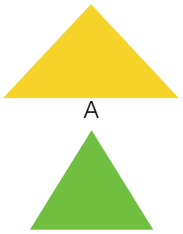
15

k-copying means each rule can make at most k copies of a single string. A context need not be k-copying.

k-copying PMCFGs (of bounded dimension and rank)

16

derivation tree



$(x_1,\ldots,x_r) \mapsto v_0\, x_{i_1}\, v_1 \ldots x_{i_n}\, v_n$

non-regular r-variable pattern (with no bound on n)

$(w_1,\ldots,w_r) \in (\Sigma^*)^r$

r-tuple of strings

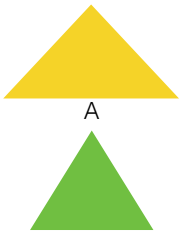
context-enumerable	×
substructure-enumerable	✓

There is no bound on the number of copies a context makes of a component of a substructure.

k-copying IO CFTGs

17

derivation tree



$t[x_1,\ldots,x_r] \mapsto u[t[v_1,\ldots,v_r]]$

one-variable tree pattern + r-tuple of trees

$t[x_1,\ldots,x_r] \in T_\Delta(X_r)$

r-variable tree pattern

context-enumerable	×
substructure-enumerable	×

Narrowing the Learning Target for k-copying PMCFGs (C&Y 2014)

18

The dual approach.

$\mathbb{G}$ = the class of k-copying PMCFGs bounded dimension, rank

$\mathbb{C}_k = \{ C \in \mathbb{C} \mid C \text{ is k-copying} \}$

$\mathbb{C}_k|_D = \{ C \in \mathbb{C}_k \mid C(X) \in D \text{ for some } X \in \mathbb{S} \}$ poly-time enumerable

Use finite subsets $\mathbf{C}$ of $\mathbb{C}_k|_D$ (with $|\mathbf{C}| \leq m$) as nonterminals

Validate candidate rules using $\mathbb{S}|_E$ (where $D \subseteq E$)

Works for k-copying PMCFGs with **(k,m)-FCP**:

- each nonterminal A is characterized by a finite set $\mathbf{C}_A$ of **k-copying** A-contexts (with $|\mathbf{C}_A| \leq m$)

Goal of This Work

- Define large classes of nonlinear grammars that are context- or substructure-enumerable.
- Show that PMCFG-type distributional learning works for these classes.

19

Use almost linear second-order ACGs on trees.

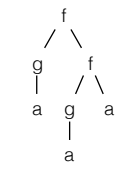
Almost Linear Second-Order ACGs on Trees

- Use **almost linear λ-terms** as functions in rules.
- A general formalism encompassing all known efficiently recognizable “context-free” formalisms on trees.
- Equivalent to
  - attribute grammars outputting trees
  - MSO-definable tree-to-term-graph transductions + unfolding

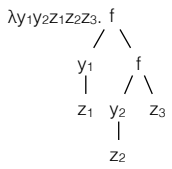
20

Generalizes IO CFTGs.

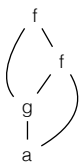
Almost Linear λ-terms



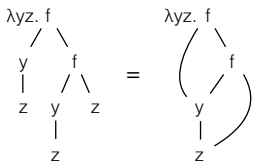
tree



linear λ-term



tree with sharing



almost linear λ-term

21

A nonlinear λ-term sometimes can be made to look linear by sharing of subterms (of atomic type).
In that case, the λ-term is said to be almost linear.

Almost Linear ACGs

22

s(X(λz^o.fzz)) :- p(X)
p(λy$^{o\to o}$.X(λz^o.y(fzz))) :- p(X)
p(λy$^{o\to o}$.ya) :-

f : o→o→o
a : o
$\mathcal{H}$(s) = o
$\mathcal{H}$(p) = (o→o)→o

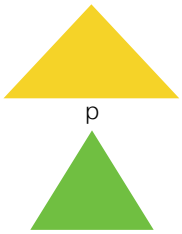
s — f(f(faa)(faa))(f(faa)(faa)))
p — λy$^{o\to o}$.y(f(faa)(faa))
p — λy$^{o\to o}$.y(faa)
p — λy$^{o\to o}$.ya

perfect binary trees

Almost Linear Second-Order ACGs

23

derivation tree



N
closed almost linear λ-term
of type $\mathcal{H}$(p) → o

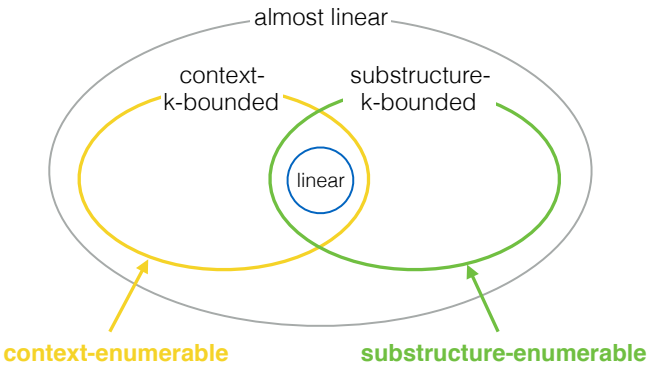
M
closed almost linear λ-term
of type $\mathcal{H}$(p)

context-enumerable	×
substructure-enumerable	×

Neither context- nor substructure-enumerable even with a bound on certain parameters.
Extends IO CFTGs.

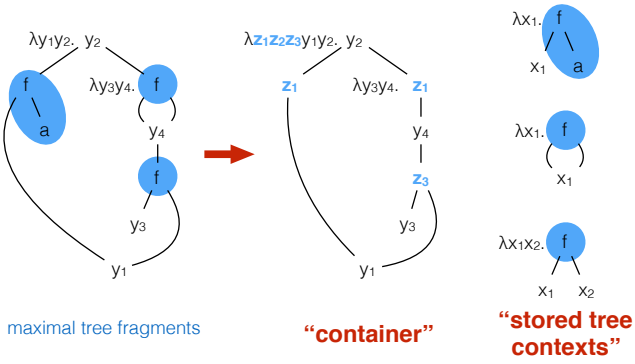
Almost Linear ACGs with Bounded Nonlinearity

24



Limit the “amount” of nonlinearity that shows up in substructures or in contexts.

Decomposition of Almost Linear λ-terms over a Tree Signature



maximal tree fragments

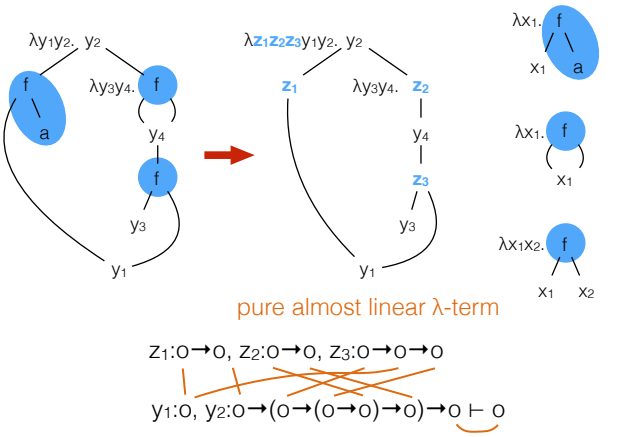
"container"

"stored tree contexts"

25

Preparation for the definition of k-boundedness of an almost linear λ-term.

Extract maximal tree fragments from a given almost linear λ-term in such a way that the remaining pure λ-term is almost linear.



pure almost linear λ-term

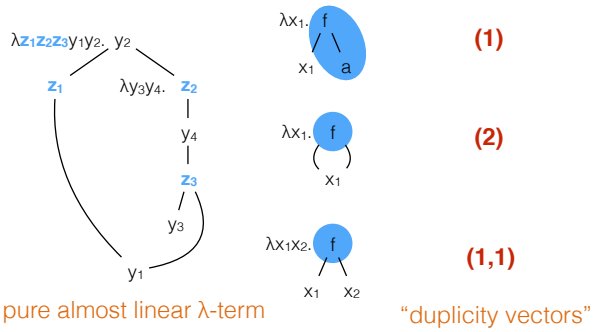
finitely many possible containers for each type

26

The number of stored tree contexts is bounded by the number of positive atomic type occurrences in the type of the input λ-term.

The arity of each stored tree context is bounded by the number of negative atomic type occurrences in the type of the input λ-term.

Degree of Nonlinearity



pure almost linear λ-term

(1)

(2)

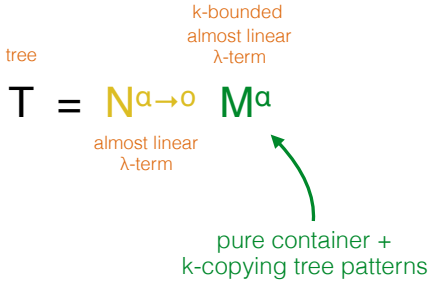
(1,1)

"duplicity vectors"

M is **k-bounded** ⇔ M's duplicity vectors ∈ {1,…,k}*

27

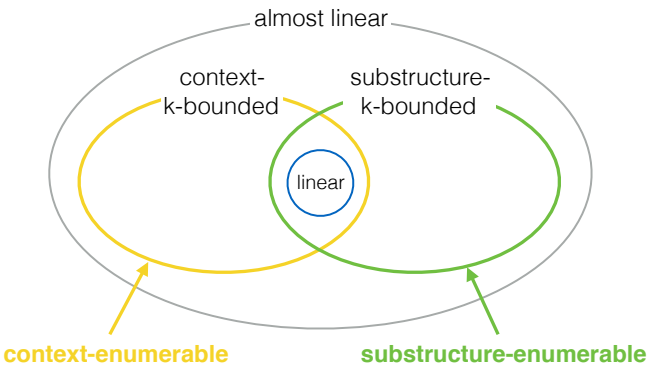
Extracting k-bounded λ-terms from a Tree



28

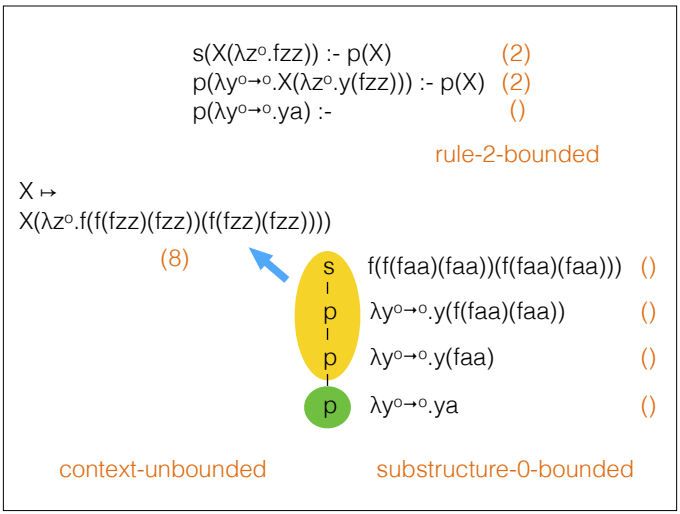
Such M are polynomial-time enumerable.

Almost Linear ACGs with Bounded Nonlinearity



29

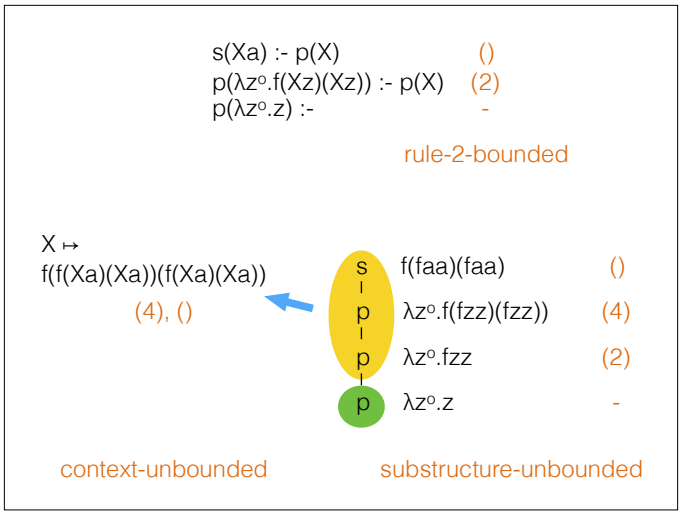
Substructure-k-bounded: Every substructure the grammar provides as the yield of a subtree of a derivation tree is a k-bounded almost linear λ-term.
Context-k-bounded: Every context the grammar provides as the yield of a derivation tree with a hole is a k-bounded almost linear λ-term.



30

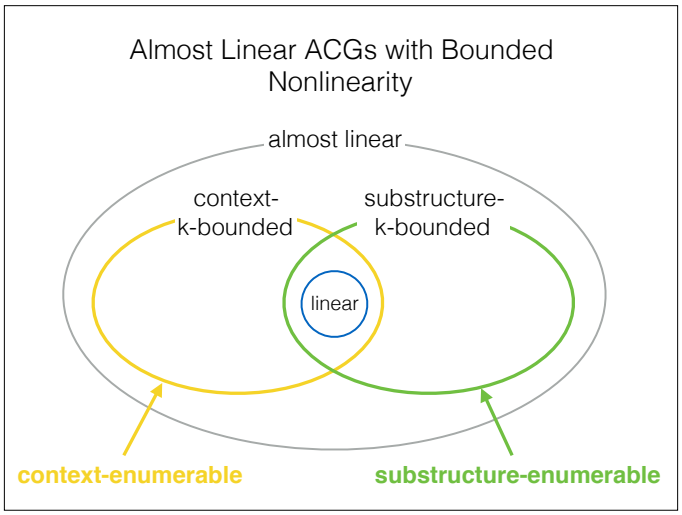
An example of a substructure-k-bounded almost linear ACG.

31



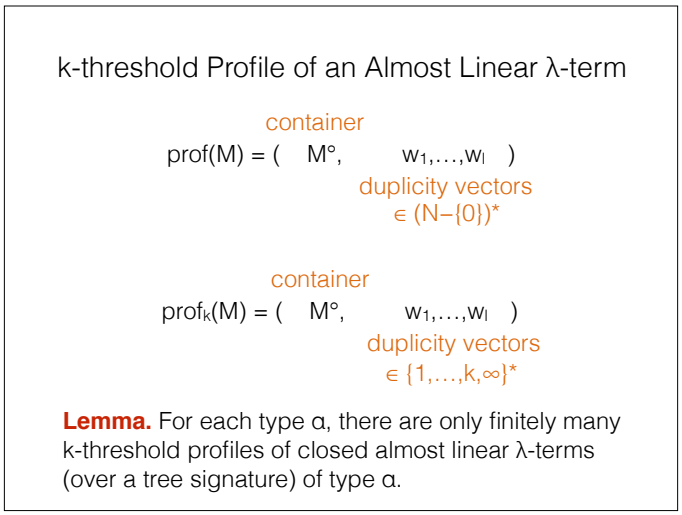
An example that is unbounded on both sides.

32



Are these classes decidable?

33



Deciding k-boundedness

Lemma. If $\mathrm{prof}_k(M_i) = \mathrm{prof}_k(M_i')$ for $i = 1,\ldots,n$, then $\mathrm{prof}_k(NM_1\ldots M_n) = \mathrm{prof}_k(NM_1'\ldots M_n')$.

Lemma. The derivation trees of an almost linear ACG that involve only k-bounded substructures form a regular set.

Lemma. The derivation trees of an almost linear ACG that involve only k-bounded contexts form a regular set.

34

The class of substructure-k-bounded almost linear ACGs is a decidable class.
So is the class of context-k-bounded almost linear ACGs.
Can take either of these classes (with certain parameters fixed) as hypothesis space of distributional learning.

PMCFG-Type Dual Distributional Learning of Substructure-k-bounded Almost Linear ACGs

$\mathfrak{G}$ = the class of almost linear ACGs that are rule- and substructure-k-bounded (with certain parameters fixed)

$\mathfrak{C}_k = \{ C \in \mathfrak{C} \mid C \text{ is k-bounded} \}$

$\mathfrak{C}_k|_D = \{ C \in \mathfrak{C}_k \mid C(X) \in D \text{ for some } X \in \mathfrak{S} \}$ (poly-time enumerable)

Use finite subsets $\mathbf{C}$ of $\mathfrak{C}_k|_D$ (with $|\mathbf{C}| \leq m$) as "nonterminals"

Validate candidate rules using $\mathfrak{S}|_E$ (where $D \subseteq E$)

Target grammars in $\mathfrak{G}$ with **(k,m)-FCP**:

- each "nonterminal" p is characterized by a finite set $\mathbf{C}_p$ of **k-bounded** λ-terms of type $\mathcal{H}(p) \to o$ (with $|\mathbf{C}_p| \leq m$)

35

There is no guarantee that the hypothesized grammar is substructure-k-bounded.
This can cause overgeneration.

Ensuring Substructure-k-boundedness

- Use $(\mathbf{C},\pi)$ as nonterminals where
  - $\mathbf{C}$ is a finite subset of $\mathfrak{C}_k|_D$ consisting of λ-terms of type α (with $|\mathbf{C}| \leq m$)
  - π is a k-threshold profile for type α
- Construct rules
  $(\mathbf{C}_0,\pi_0)(P^{\alpha_1\to\cdots\to\alpha_n\to\alpha_0}\ X_1\ldots X_n) :- (\mathbf{C}_1,\pi_1)(X_1),\ldots,(\mathbf{C}_1,\pi_n)(X_n)$
  such that
  - P is a k-bounded almost linear λ-term "observed" in D
  - $\pi_0 = P\pi_1\ldots\pi_n$.
- Validate for $X_i \in \mathfrak{S}|_E \cap \mathrm{prof}_k^{-1}(\pi_i)$.

36

The second bullet point guarantees substructure-k-boundedness.

Profile-Sensitive (k,m)-FCP

- Each "nonterminal" p of G is characterized by finite sets $\mathbf{C}_{p,\pi}$ of $\mathbb{C}_k$ (with $|\mathbf{C}_{p,\pi}| \leq m$), one for each profile π for $\mathcal{H}(p)$:

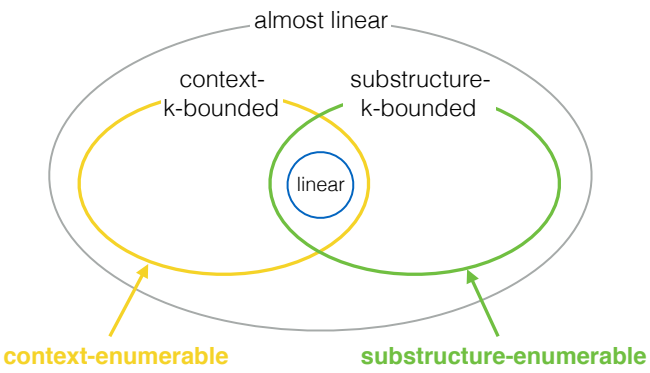
  for all $X \in \mathbb{S} \cap \text{prof}_k^{-1}(\pi)$,

  $C(X) \in L(G)$ for all $C \in \mathbf{C}_{p,\pi} \Rightarrow X$ is a p-substructure

37

This is a more natural notion of FCP corresponding to the validation in the learning algorithm.

Almost Linear ACGs with Bounded Nonlinearity



38

"PMCFG-type" distributional learning is possible for both these classes (more complicated for the context-k-bounded grammars).
Leads to a "profile-sensitive" notion of FCP (and of FKP).

39