Multiple Context-Free Langauges and Non-Duplicating Macro Languages

Makoto Kanazawa, NII, Tokyo, Japan
Based on joint work with Sylvain Salvati

1956 Context-Free Grammar (Chomsky)

1968 Macro Grammar (Fischer)
1970 Context-Free Tree Grammar (Rounds)

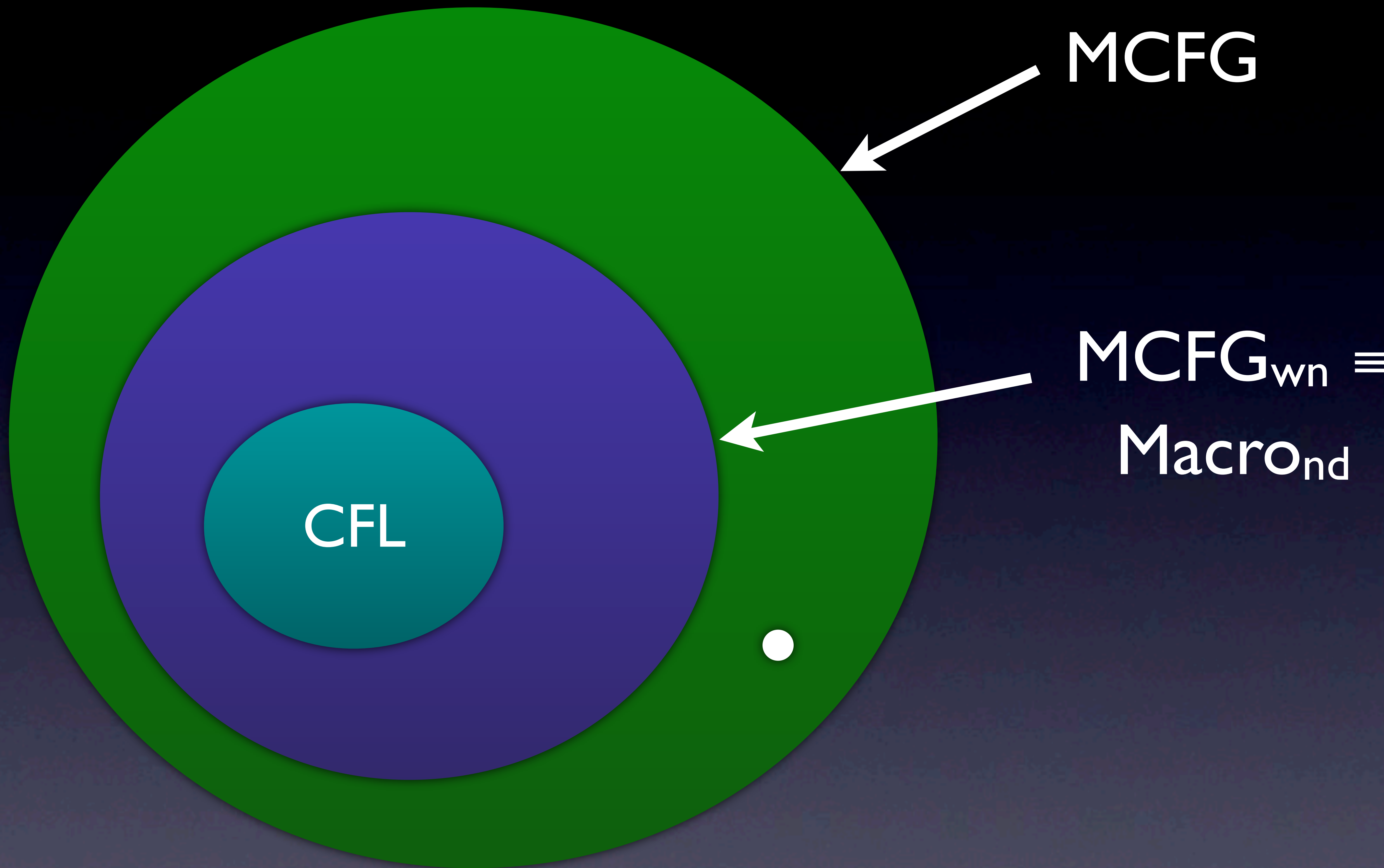
1975 Tree-Adjoining Grammar (Joshi et al.)

1984 Head Grammar (Pollard)

1991 Multiple Context-Free Grammar (Seki et al.)
1992 Coupled-Context-Free Grammar (Guan)

2007 Well-Nested Dependency Structures (Kuhlmann)

This talk is about multiple context-free grammars and non-duplicating macro grammars, which are equivalent to “well-nested” multiple context-free grammars.



$\{ L_0 \mid \{ w\#w \mid w \in L_0 \} \in \mathcal{L} \}$ copying power

We want to understand better the difference between the corresponding classes of languages.
We characterize the "copying power" of MCFG_{wn}.
First introduce MCFG, then motivate well-nestedness.

Context-Free Grammar

$$A \rightarrow B\ C$$

$$\beta\ A\ \gamma \Rightarrow \beta\ B\ C\ \gamma$$

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

MCFG is a natural extension of CFG.
The standard interpretation of CFG rules: rewriting instructions.

Nonterminals as Predicates

$$A \rightarrow B\ C$$

$$A(xy) \leftarrow B(x), C(y)$$

Horn clause

nonterminals = unary predicates on strings

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

This rule says “if B derives x and C derives y, then A derives xy”.
Nonterminals can be interpreted as unary predicates on strings.

Nonterminals as Predicates

$A(x_1 y_1, x_2 y_2) \leftarrow B(x_1, x_2), C(y_1, y_2)$ Horn clause

nonterminals = *k*-ary predicates on strings

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

In MCFG, nonterminals are k-ary predicates on strings.

$$S(x_1\#x_2) \leftarrow A(x_1, x_2)$$

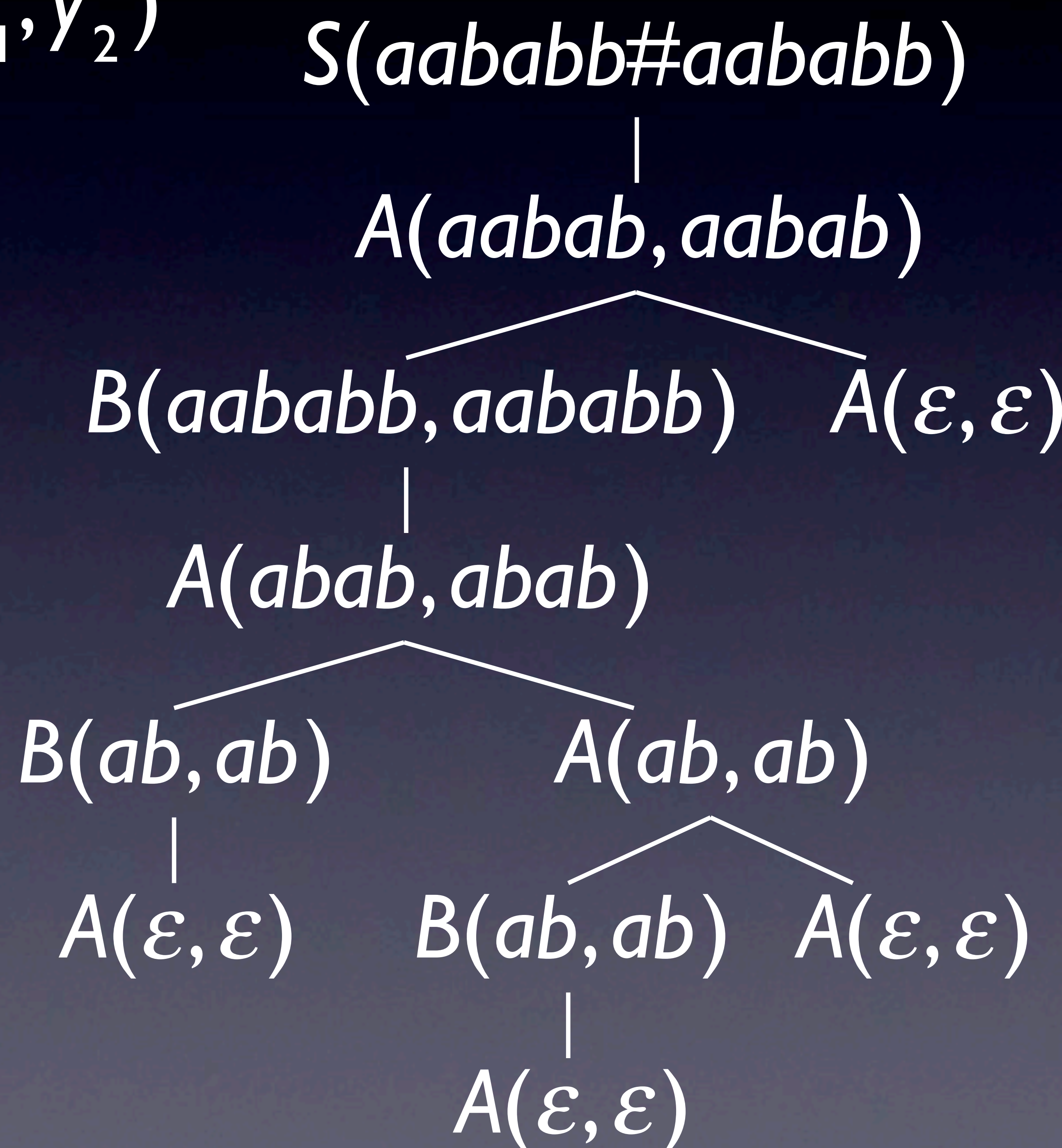
$$\{w\#w \mid w \in D_1^*\}$$

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

$$A(x_1y_1, x_2y_2) \leftarrow B(x_1, x_2), A(y_1, y_2)$$

$$B(ax_1b, ax_2b) \leftarrow A(x_1, x_2)$$

2-MCFG



This is an example of a 2-MCFG.
An m-MCFG allows nonterminals to take up to m arguments.
An example of a derivation tree.

Multiple Context-Free Grammar

$$A(\alpha_1, \ldots, \alpha_m) \leftarrow B(x_1, \ldots, x_p), \ldots, D(z_1, \ldots, z_r)$$

- Each variable occurs at most once in $\alpha_1 \ldots \alpha_m$
- nonterminal X = $\dim(X)$-ary predicate on strings
- $\dim(S) = 1$

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

restricted type of *elementary formal systems* (Smullyan 1961)

MCFGs are a natural extension of the Horn clause program view of CFGs.

Multiple Context-Free Grammar

- Introduced by Seki, Matsumura, Fujii, and Kasami (1987–1991)
- Independently by Vijay-Shanker, Weir, and Joshi (1987) under the name LCFRS
- Generalization of TAG (Joshi, Levy, and Takahashi 1975)

Vijay-Shanker et al. called an MCFG an “LCRFS”.
TAG = Tree Adjoining Grammar.

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

$\{ a^m b^n c^m d^n \mid m, n \geq 0 \}$

$S(x_1\#x_2) \leftarrow A(x_1, x_2)$
$A(x_1y_1, x_2y_2)$
$\quad \leftarrow B(x_1, x_2), A(y_1,y_2)$
$B(ax_1b, ax_2b) \leftarrow A(x_1, x_2)$
$A(\varepsilon, \varepsilon) \leftarrow$

$S(x_1x_2) \leftarrow A(x_1, x_2)$
$A(ax_1, cx_2) \leftarrow A(x_1, x_2)$
$A(x_1b, x_2d) \leftarrow A(x_1, x_2)$
$A(\varepsilon, \varepsilon) \leftarrow$

2-MCFG

2-MCFG
"non-branching"

The previous example and a new example.
The new example is "non-branching".

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

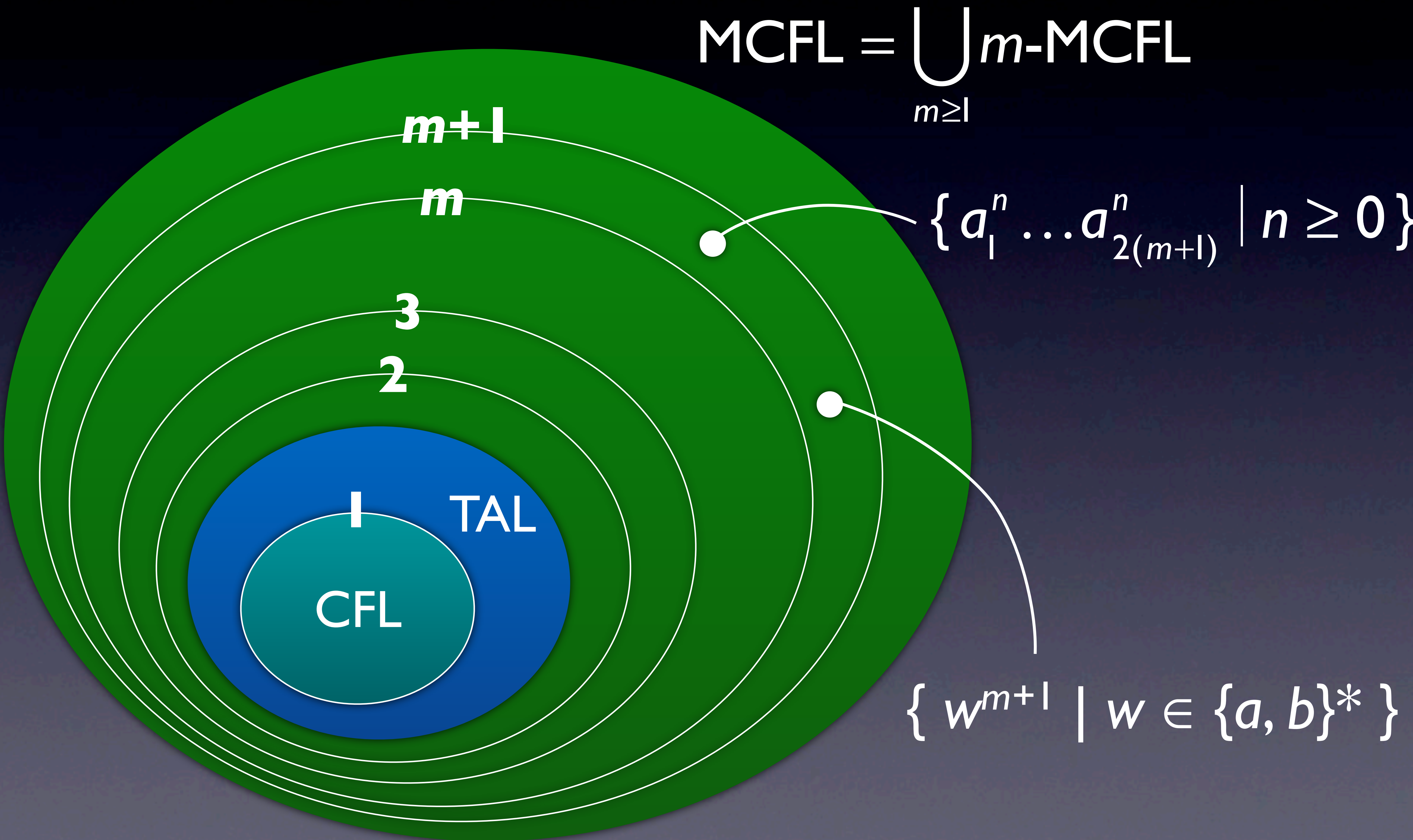
$$S(x_1 \ldots x_m) \leftarrow A(x_1, \ldots, x_m)$$
$$A(a_1 x_1 a_2, \ldots, a_{2m-1} x_m a_{2m}) \leftarrow A(x_1, \ldots, x_m)$$
$$A(\varepsilon, \ldots, \varepsilon) \leftarrow$$

m-MCFG

non-branching

Another non-branching MCFG.

MCFL Hierarchy



This is an infinite hierarchy.
TAL is the class of languages generated by Tree Adjoining Grammars.

Complexity of Recognition

	fixed language recognition	universal recognition
CFG	LOGCFL-complete	P-complete
TAG	LOGCFL-complete	P-complete
m-MCFG	LOGCFL-complete	NP-complete ($m \geq 2$)
MCFG	LOGCFL-complete	PSACE-complete/ EXPTIME-complete

Satta 1992, Kaji, Nakanishi, Seki, and Kasami 1992

Tabular recognition algorithms similar to CYK are easy to devise.

Mildly Context-Sensitive Grammar Formalism

- Properly extends CFG
- Polynomial-time parsable
- Semilinear

$\{ \psi(w) \mid w \in L \}$ Parikh image

$\psi(w) = (|w|_a, \ldots, |w|_z)$

- Exhibits limited cross-serial dependencies

Aravind K. Joshi

An informally defined notion of a "mildly context-sensitive grammar formalism".
The notion of "cross-serial dependency" comes from linguistics.

Cross-Serial Dependencies

that Charles lets Mary help Peter teach John to swim

daß der Karl die Maria dem Peter den Hans schwimmen lehren helfen läßt

dat Karel Marie Piet Jan laat helpen leren zwemmen

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

Dependencies between verbs and objects are nested in English and German, but are cross-serial in Dutch and Swiss German.

English/German

that Charles lets Mary help Peter teach John to swim

{ } ( ) { }

daß der Karl die Maria dem Peter den Hans schwimmen lehren helfen läßt

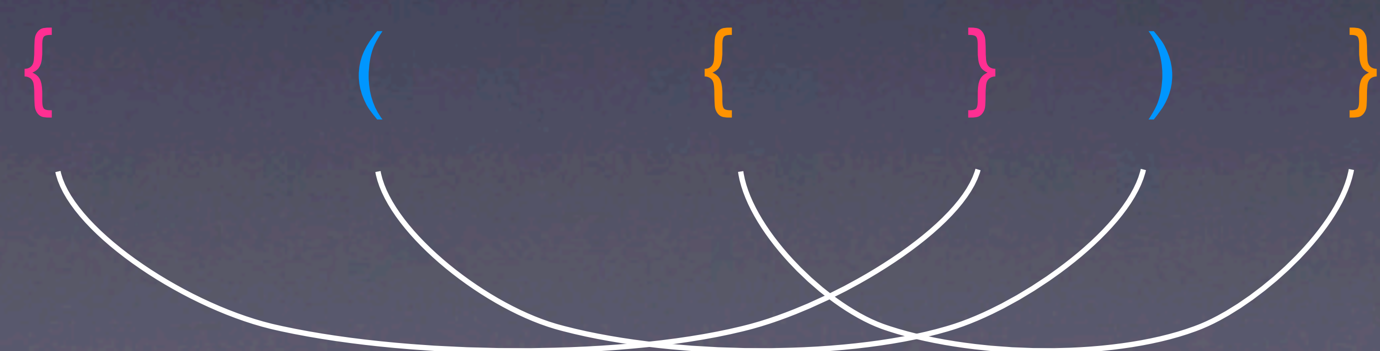
{ ({ }) }

Dependencies between verbs and objects are like pairs of (nested) parentheses in English and German.
This type of dependency is adequately handled by CFGs.

Dutch/Swiss German

dat Karel Marie Piet Jan laat helpen leren zwemmen

dass de Karl d’Maria em Peter de Hans laat hälfe lärne schwüme



Dependencies between verbs and objects are not like pairs of (nested) parentheses in Dutch and Swiss German.

Swiss German

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

Acc. Dat. Acc.

{ ({ }) }

$$\{ a^m b^n c^m d^n \mid m, n \geq 1 \} \notin \mathrm{CFL}$$

Shieber 1985

The pairing of verbs and objects can be clearly seen in Swiss German.
em Peter is dative, de Hans is accusative
Intersection with a regular set + homomorphism takes Swiss German to a non-CFL.

Limited Cross-Serial Dependencies

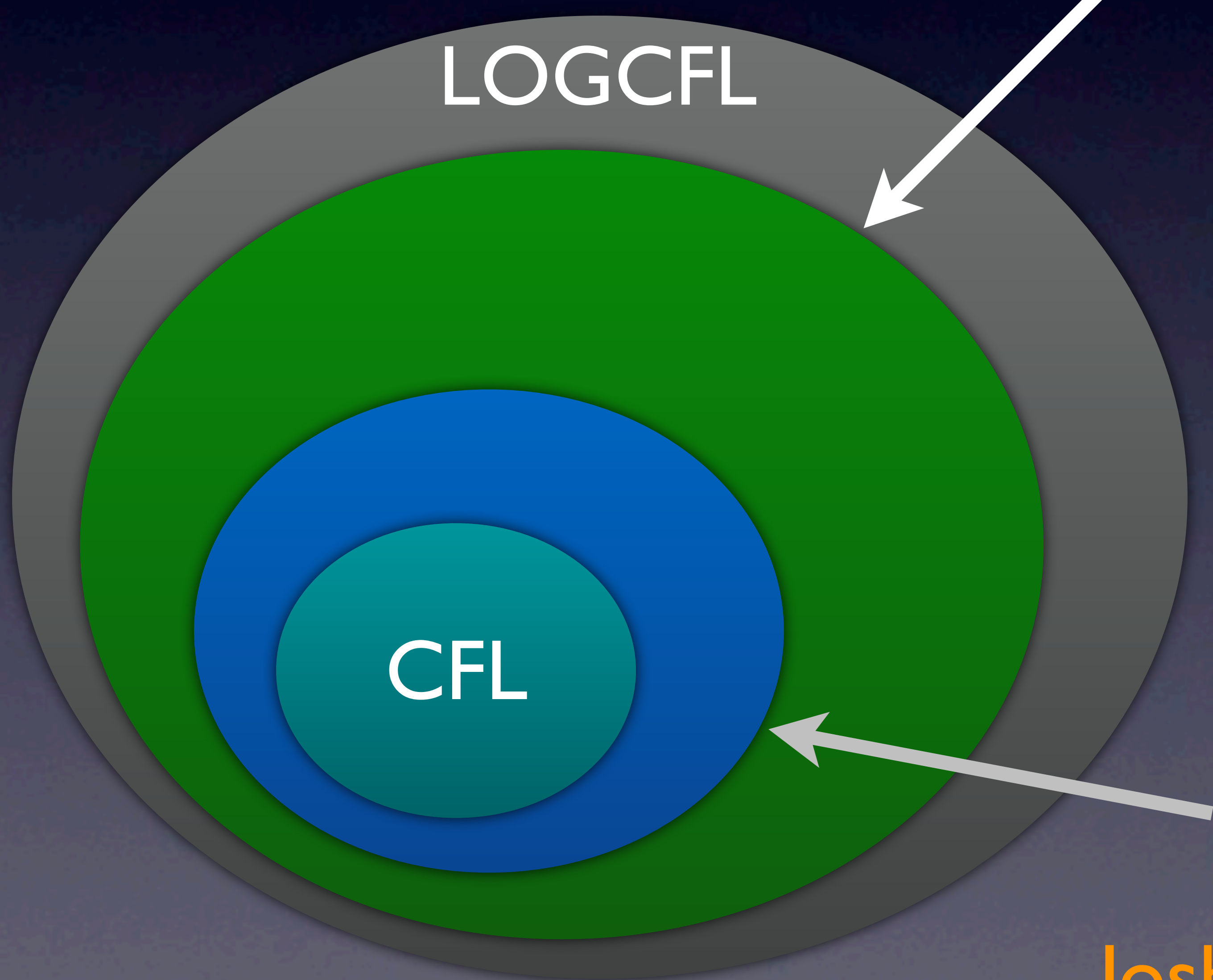
"MCSGs capture only certain kinds of dependencies, such as nested dependencies and certain limited kinds of crossing dependencies (for example, in subordinate clause constructions in Dutch or some variations of them, but perhaps not in the so-called MIX … language …)"

Joshi, Vijay-Shanker, and Weir 1991

$$\mathrm{MIX} = \{ w \in \{a,b,c\}^* \mid |w|_a = |w|_b = |w|_c \}$$

The language MIX was supposed to be outside of the class of mildly context-sensitive languages.

Convergence of Mildly Context-Sensitive Grammar Formalisms



$\mathrm{MCFG} \equiv \mathrm{MCTAG} \equiv \mathrm{HR}$
$\equiv \mathrm{OUT(DTWT)}$
$\equiv \mathrm{yDT_{fc}(REGT)} \equiv \mathrm{LUSCG}$
$\equiv \mathrm{MG} \equiv \mathrm{ACG}_2$

$\mathrm{TAG} \equiv \mathrm{CCG} \equiv \mathrm{LIG} \equiv \mathrm{HG}$

Joshi, Vijay-Shanker, and Weir 1991

The “convergence of mildly context-sensitive ...” originally referred to TAG, CCG, LIG, HG, but in retrospect, the convergence at the level of MCFL is more robust.
A greater number of equivalent formalisms, more diverse.

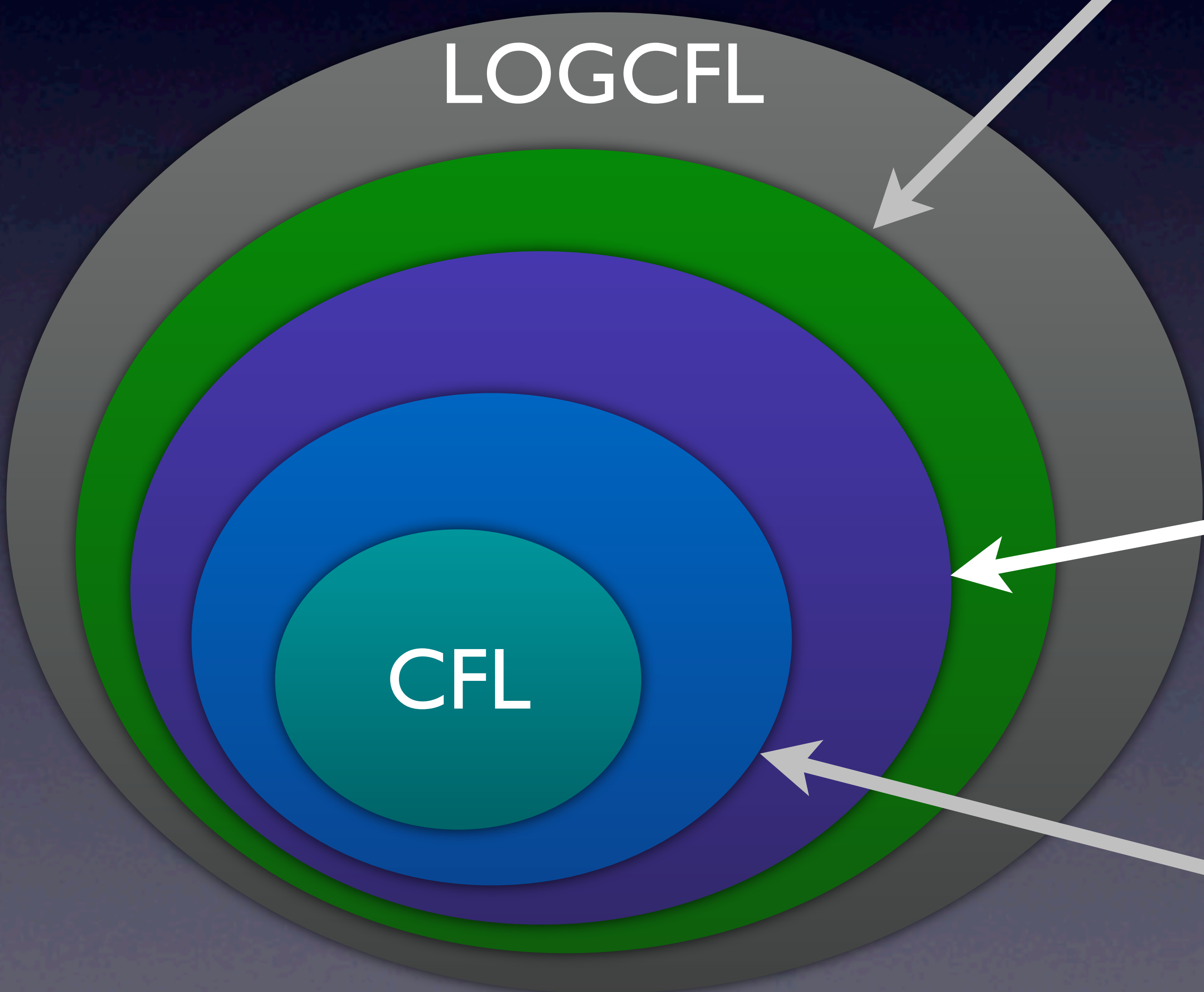
MCFG = “mildly context-sensitive”?

“The class of mildly context-sensitive languages seems to be most adequately approached by [MCFGs].”

Groenink 1997

A near-consensus has emerged, identifying “mildly context-sensitive” with MCFG.
But this consensus has recently been called into question.

Yet another point of convergence



$\mathrm{MCFG} \equiv \mathrm{MCTAG} \equiv \mathrm{HR}$
$\equiv \mathrm{OUT(DTWT)}$
$\equiv \mathrm{yDT_{fc}(REGT)} \equiv \mathrm{LUSCG}$
$\equiv \mathrm{MG} \equiv \mathrm{ACG}_{(2,4)}$

$\mathrm{MCFG}_{\mathrm{wn}} \equiv \mathrm{CCFG} \equiv \mathrm{Macro}_{\mathrm{nd}}$
$\equiv \mathrm{ACG}_{(2,3)}$

$\mathrm{TAG} \equiv \mathrm{CCG} \equiv \mathrm{LIG} \equiv \mathrm{HG}$

Well-nested MCFGs are equivalent to coupled-context-free grammars, non-duplicating macro grammars, and a subclass of second-order abstract categorial grammars

Well-nested MCFGs

$$S(x_1 y_1 x_2 y_2) \leftarrow A(x_1, x_2), B(y_1, y_2)$$

$$S(x_1 y_1 y_2 x_2) \leftarrow A(x_1, x_2), B(y_1, y_2)$$

$$C(x_1 y_1, y_2 z_1, z_2 x_2 z_3) \leftarrow A(x_1, x_2), B(y_1, y_2), C(z_1, z_2, z_3)$$

$$C(z_1 x_1, x_2 z_2, y_1 y_2 z_3) \leftarrow A(x_1, x_2), B(y_1, y_2), C(z_1, z_2, z_3)$$

Cf. Kuhlmann 2007

Assume all rules are "non-permuting" (x_i appears before x_j if $i < j$).

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

$S(x_1\#x_2) \leftarrow A(x_1, x_2)$

$\boxed{A(x_1y_1, x_2y_2) \leftarrow B(x_1, x_2), A(y_1,y_2)}$

$B(ax_1b, ax_2b) \leftarrow A(x_1, x_2)$

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

2-MCFG

non-well-nested

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

$S(x_1\#x_2) \leftarrow A(x_1, x_2)$

$A(x_1y_1, y_2x_2) \leftarrow B(x_1, x_2), A(y_1,y_2)$

$B(ax_1b, bx_2a) \leftarrow A(x_1, x_2)$

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

2-MCFG

well-nested

The first example is not well-nested.
A similar, but well-nested grammar.

$$\{\ a^m b^n c^m d^n \mid m, n \geq 0\ \}$$

$$S(x_1 x_2) \leftarrow A(x_1, x_2)$$
$$A(a x_1, c x_2) \leftarrow A(x_1, x_2)$$
$$A(x_1 b, x_2 d) \leftarrow A(x_1, x_2)$$
$$A(\varepsilon, \varepsilon) \leftarrow$$

2-MCFG

"non-branching"

well-nested

Non-branching MCFGs are well-nested in a trivial way.

$$\{ a_1^n \ldots a_{2m}^n \mid n \geq 0 \}$$

$$S(x_1 \ldots x_m) \leftarrow A(x_1, \ldots, x_m)$$
$$A(a_1 x_1 a_2, \ldots, a_{2m-1} x_m a_{2m}) \leftarrow A(x_1, \ldots, x_m)$$
$$A(\varepsilon, \ldots, \varepsilon) \leftarrow$$

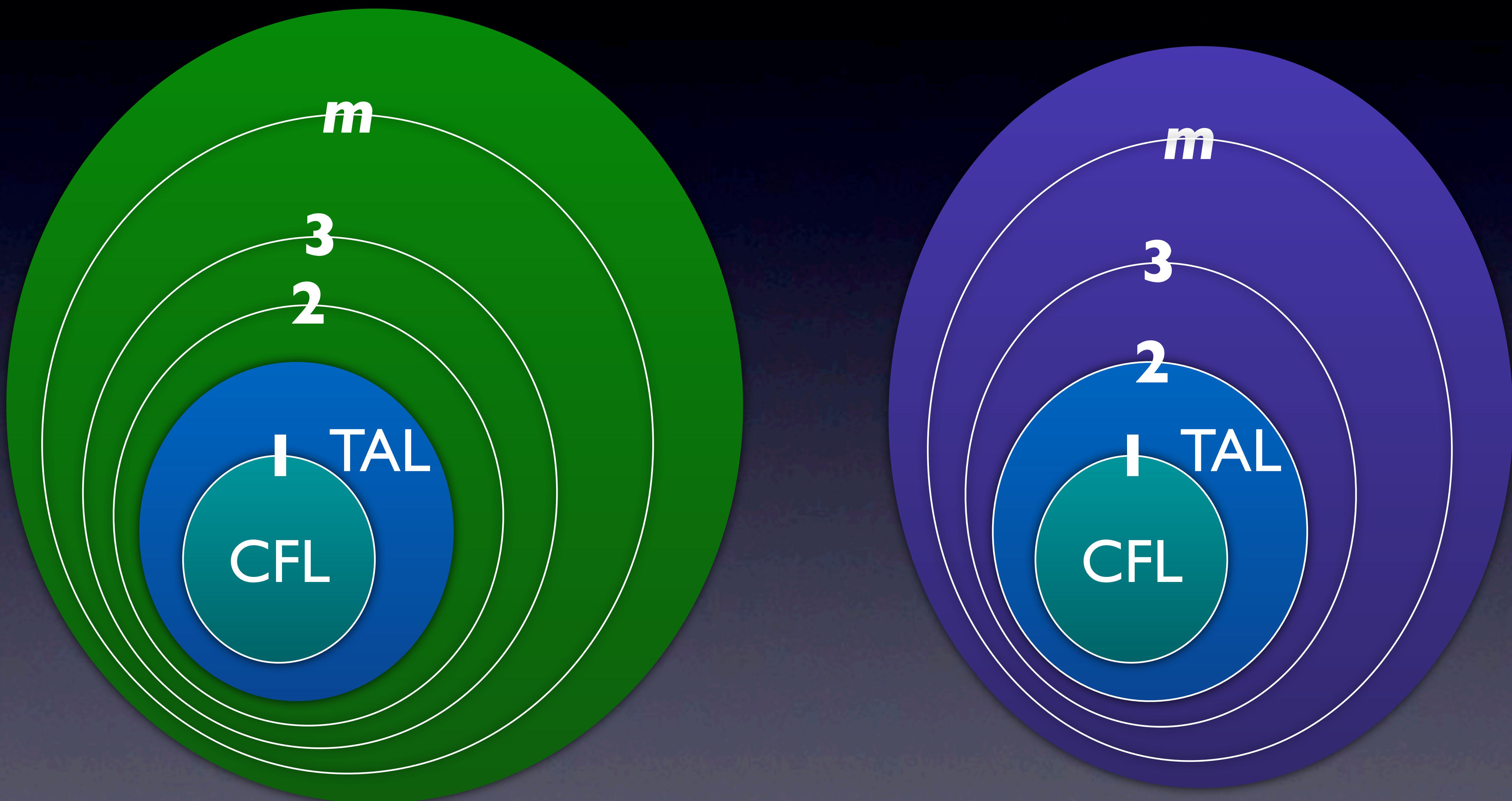
m-MCFG

non-branching

well-nested

Another non-branching, well-nested MCFG.

Two infinite hierarchies



$$\mathrm{MCFL} = \bigcup_{m \geq 1} m\text{-}\mathrm{MCFL} \qquad \mathrm{MCFL}_{\mathrm{wn}} = \bigcup_{m \geq 1} m\text{-}\mathrm{MCFL}_{\mathrm{wn}}$$

2–MCFL$_{\mathrm{wn}}$ coincides with TAL.

m-MCFL vs. m-MCFL$_{\text{wn}}$

$\mathrm{RESP}_2 = \{ a_1^i a_2^i b_1^j b_2^j a_3^i a_4^i b_3^j b_4^j \mid i, j \geq 0 \}$ Weir 1989

$\mathrm{RESP}_2 \in$ 2-MCFL – 2-MCFL$_{\text{wn}}$ Seki et al. 1991

$\mathrm{RESP}_m = \{ a_1^i a_2^i b_1^j b_2^j \ldots a_{2m-1}^i a_{2m}^i b_{2m-1}^j b_{2m}^j \mid i, j \geq 0 \}$

$\mathrm{RESP}_m \in m$-MCFL – m-MCFL$_{\text{wn}}$ for $m \geq 2$

Seki and Kato 2008

$\mathrm{RESP}_m \in 2m$-MCFL$_{\text{wn}}$

m–MCFL and m–MCFL$_{\text{wn}}$ have many languages in common, but are of course different. Separation is easy at each level.

Complexity of Recognition

	fixed language recognition	universal recognition
CFG	LOGCFL-complete	P-complete
m-MCFG$_{wn}$	LOGCFL-complete	P-complete
m-MCFG	LOGCFL-complete	NP-complete ($m \geq 2$)
MCFG$_{wn}$	LOGCFL-complete	PSPACE-complete
MCFG	LOGCFL-complete	PSACE-complete/ EXPTIME-complete

The complexity of universal recognition doesn't go up for well-nested m-MCFGs.

Binarization

$$m\text{-MCFL}_{wn} = m\text{-MCFL}_{wn}(2)$$

Kanazawa and Salvati 2010
Gómez-Rodríguez, Kuhlmann, and Satta 2010

Well-nested m-MCFGs are "binarizable", unlike general m-MCFGs.
A generalization of Chomsky Normal Form.

An m-MCFL$_{\text{wn}}$ is $2m$-iterative

$L \in m$-MCFL$_{\text{wn}}$

⬇

For all but finitely many $z \in L$,

$$z = u_0 v_1 u_1 \ldots v_{2m} u_{2m}$$

$$|v_1 \ldots v_{2m}| \geq 1$$

$$u_0 v_1^i u_1 \ldots v_{2m}^i u_{2m} \in L$$

Kanazawa 2009

A natural generalization of the Pumping Lemma for CFL.
It is not known whether every m-MCFL is 2m-iterative.
An m-MCFL is weakly 2m-iterative in the sense of having an infinite 2m-iterative subset (if infinite itself).

Chomsky-Schützenberger Theorem

$$\Delta = \Delta^{(1)} \cup \cdots \cup \Delta^{(m)}$$

$$\tilde{\Delta} = \bigcup \{ \{ [_{a,1},]_{a,1}, \ldots, [_{a,r},]_{a,r} \} \mid a \in \Delta^{(r)} \}$$

$$g : [_{a,1} \mapsto [_{a,1}$$

$$]_{a,1} \mapsto]_{a,r}$$

$$[_{a,i+1} \mapsto]_{a,i}$$

$$]_{a,i+1} \mapsto [_{a,i+1}$$

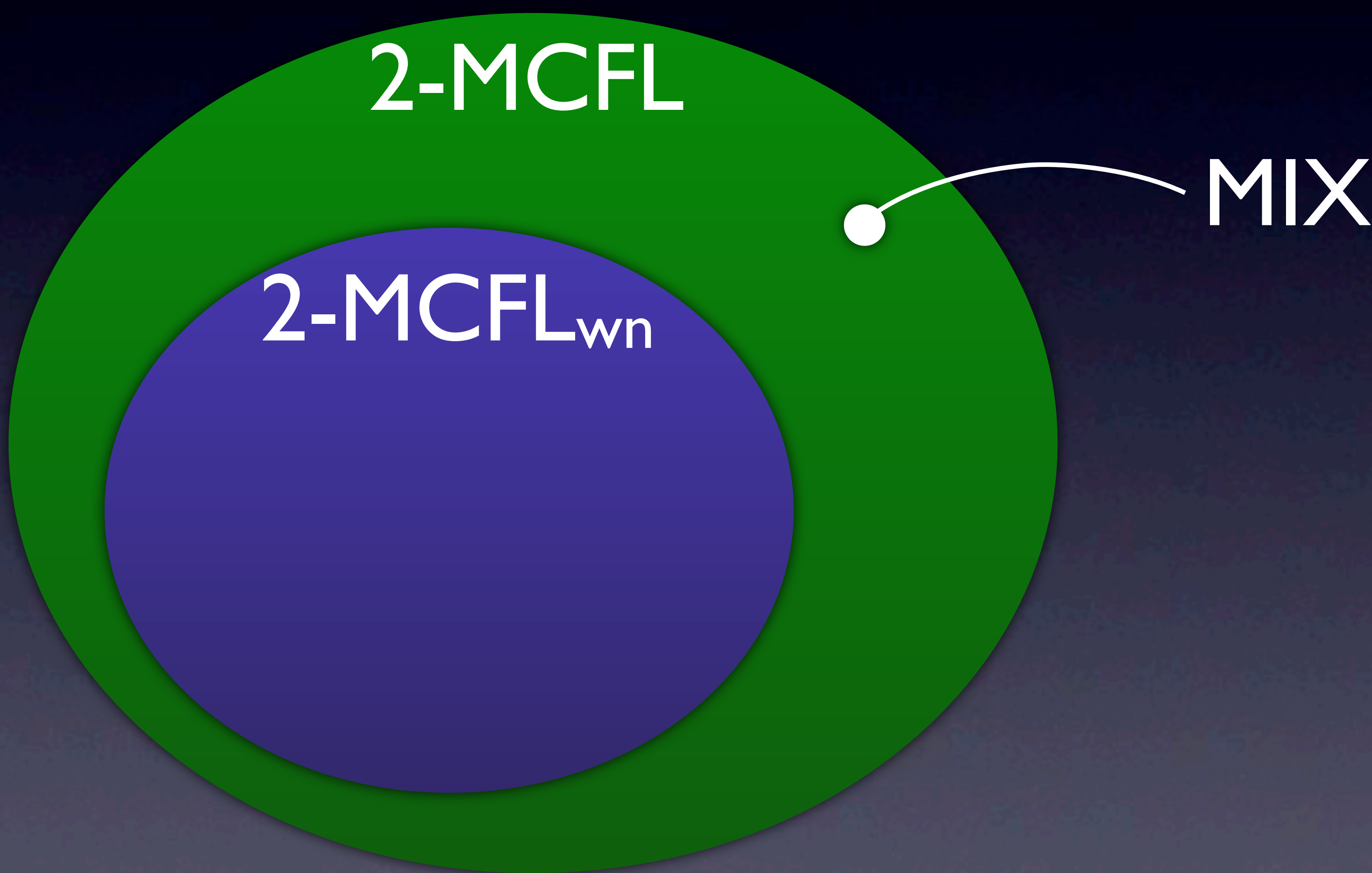
Dyck language

$$L \in m\text{-MCFL}_{\text{wn}} \Rightarrow L = h(D^*_{\tilde{\Delta}} \cap g^{-1}(D^*_{\tilde{\Delta}}) \cap R)$$

homomorphism local set

Yoshinaka, Seki, and Kaji (2010) present a Chomsky-Schützenberger theorem for m-MCFL(q). A stronger analogy with the Chomsky-Schützenberger theorem for CFL in the case of m-MCFL_{wn}, using non-duplicating context-free tree grammars.

MCFG_{wn} = “mildly context-sensitive”?

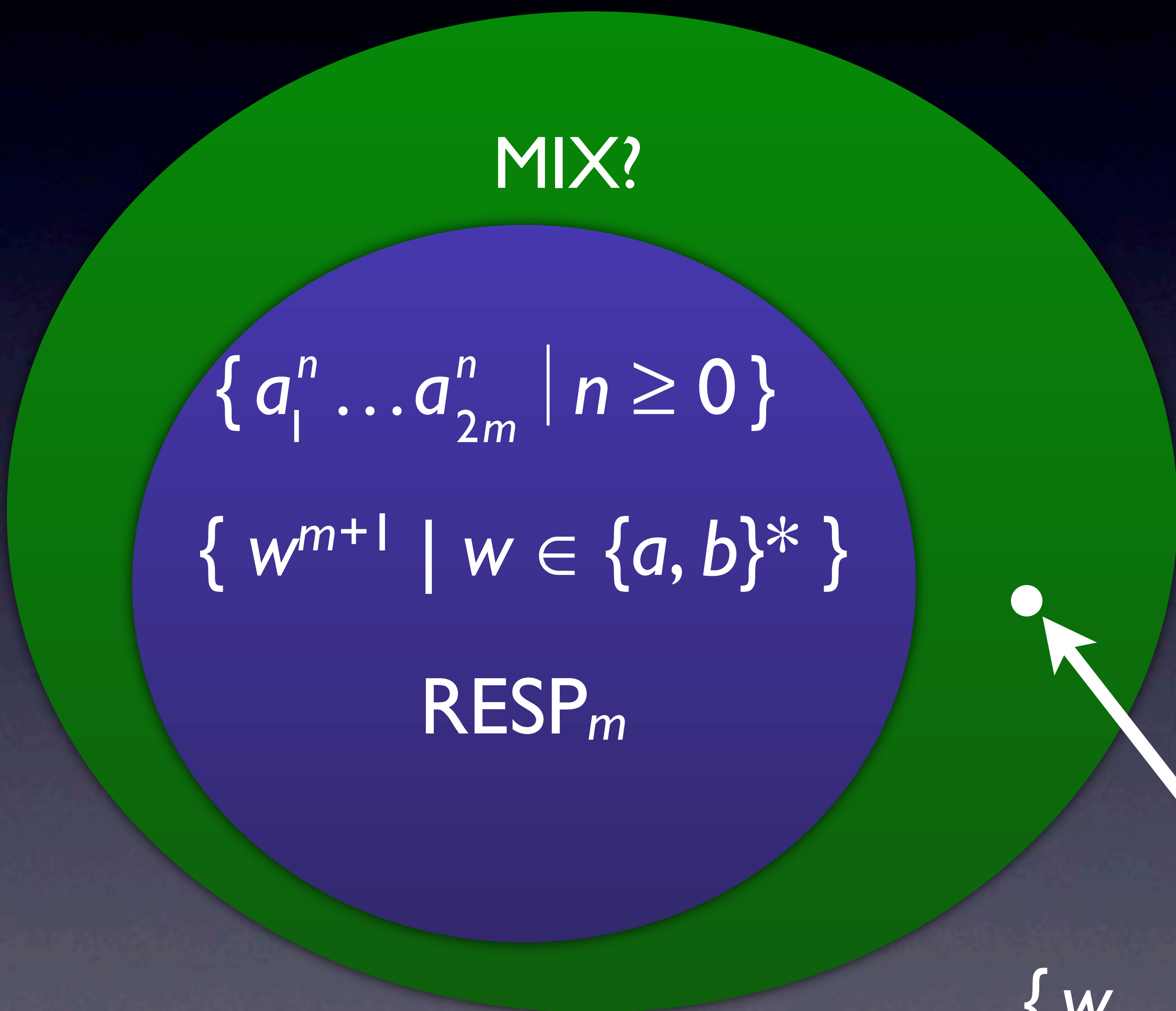


Salvati, next talk

$$\mathrm{MIX} = \{\, w \in \{a,b,c\}^* \mid |w|_a = |w|_b = |w|_c \,\}$$

MIX was supposed to be not mildly context-sensitive.
There is a simple 2-MCFG for MIX (not at all obvious).
Probably not a 2-MCFL_{wn}, but not known.

MCFL vs. MCFL$_{wn}$

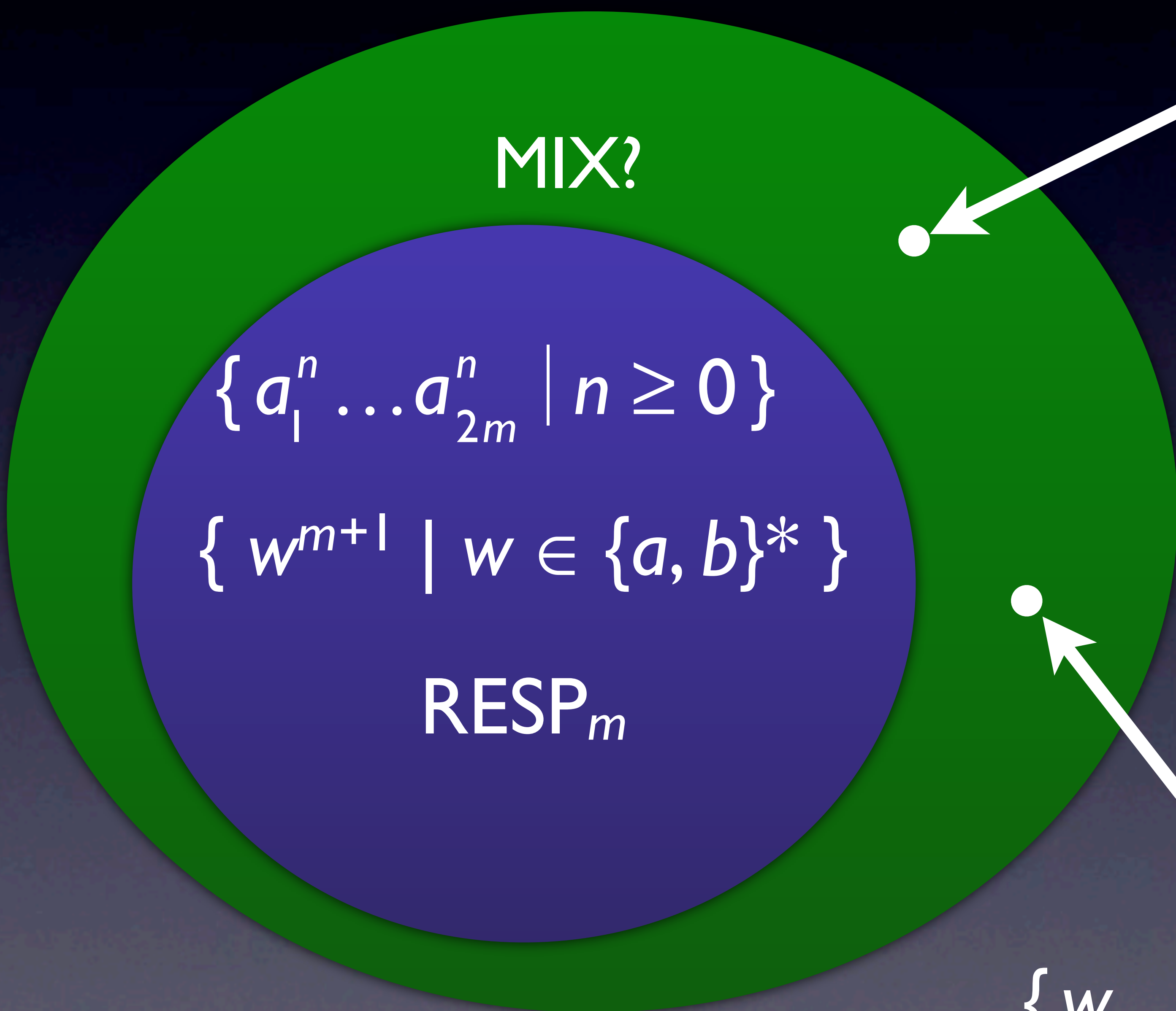


Staudacher 1993
Michaelis 2005

$$\{ w_1 \dots w_n z_n w_n z_{n-1} \dots z_1 w_1 z_0 w_1^R \dots w_n^R \mid n \in \mathbb{N}, w_i \in \{c, d\}^+, z_n \dots z_0 \in D_1^* \}$$

MCFL – MCFL$_{wn}$ was not well-understood before.
Only one example in the literature shown to be in MCFL – MCFL$_{wn}$.

MCFL vs. MCFL_{wn}



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

Engelfriet and Skyum 1976

Staudacher 1993
Michaelis 2005

$$\{ w_1 \ldots w_n z_n w_n z_{n-1} \ldots z_1 w_1 z_0 w_1^R \ldots w_n^R \mid n \in \mathbb{N}, w_i \in \{c, d\}^+, z_n \ldots z_0 \in D_1^* \}$$

One more example, using Engelfriet and Skyum's theorem (thanks to Uwe Mönnich for the pointer).
These languages are in 3-MCFL.

Copying Theorem for OI

Engelfriet and Skyum 1976

$\{ w\#w\#w \mid w \in L_0 \} \in \mathrm{OI}$

$\Updownarrow$

$\{ w\#w\#w \mid w \in L_0 \} \in \mathrm{EDT0L}$

$\Updownarrow$

$L_0 \in \mathrm{EDT0L}$

$D_1{}^* \notin \mathrm{EDT0L} \Rightarrow \{ w\#w\#w \mid w \in D_1{}^* \} \notin \mathrm{OI} \supset \mathrm{MCFL}_{wn}$

Rozoy 1987

OI = OI macro languages = indexed languages
MCFL_{wn} = non-duplicating macro $\subseteq$ IO $\cap$ OI
EDT0L = output languages of certain type of string-to-string transducers

Copying power of MCFG

For every $k \geq 1$,

$L_0 \in m$-MCFL $\Rightarrow$ $\{ w(\#w)^{k-1} \mid w \in L_0 \} \in km$-MCFL

$\{ w\#w\#w \mid w \in D_1^* \} \in$ 3-MCFL – MCFL_{wn}

Copying power of MCFG

For every $k \geq 1$,

$L_0 \in m$-MCFL $\Rightarrow$ $\{ w(\#w)^{k-1} \mid w \in L_0 \} \in km$-MCFL

$\{ w\#w\#w \mid w \in D_1{}^* \} \in$ 3-MCFL – $\mathrm{MCFL}_{\mathrm{wn}}$

?

$\{ w\#w \mid w \in D_1{}^* \} \in$ 2-MCFL – $\mathrm{MCFL}_{\mathrm{wn}}$

$$\{\, w\#w\#w \mid w \in L_0 \,\} \in \mathrm{OI}$$

$$\Updownarrow$$

$$\{\, w\#w\#w \mid w \in L_0 \,\} \in \mathrm{EDT0L}$$

$$\Updownarrow$$

$$L_0 \in \mathrm{EDT0L}$$

Engelfriet and Skyum's "Triple Copying Theorem" for OI

$$\{\, w\#w \mid w \in L_0 \,\} \in \mathrm{OI}$$

$$\Uparrow \qquad \Downarrow ?$$

$$\{\, w\#w \mid w \in L_0 \,\} \in \mathrm{EDT0L}$$

$$\Uparrow \qquad \Downarrow ?$$

$$L_0 \in \mathrm{EDT0L}$$

“Double Copying Theorem” for OI?

Double Copying Theorem for $\mathrm{MCFL}_{\mathrm{wn}}$

$$\{ w\#w \mid w \in L_0 \} \in \mathrm{MCFL}_{\mathrm{wn}}$$

$$\Updownarrow$$

$$\{ w\#w \mid w \in L_0 \} \in \mathrm{EDT0L}_{\mathrm{FIN}}$$

$$\Updownarrow$$

$$L_0 \in \mathrm{EDT0L}_{\mathrm{FIN}}$$

Double Copying Theorem for $\mathrm{MCFL}_{\mathrm{wn}}$

$\{\, w\#w \mid w \in L_0 \,\} \in \mathrm{MCFL}_{\mathrm{wn}}$

$\Updownarrow$

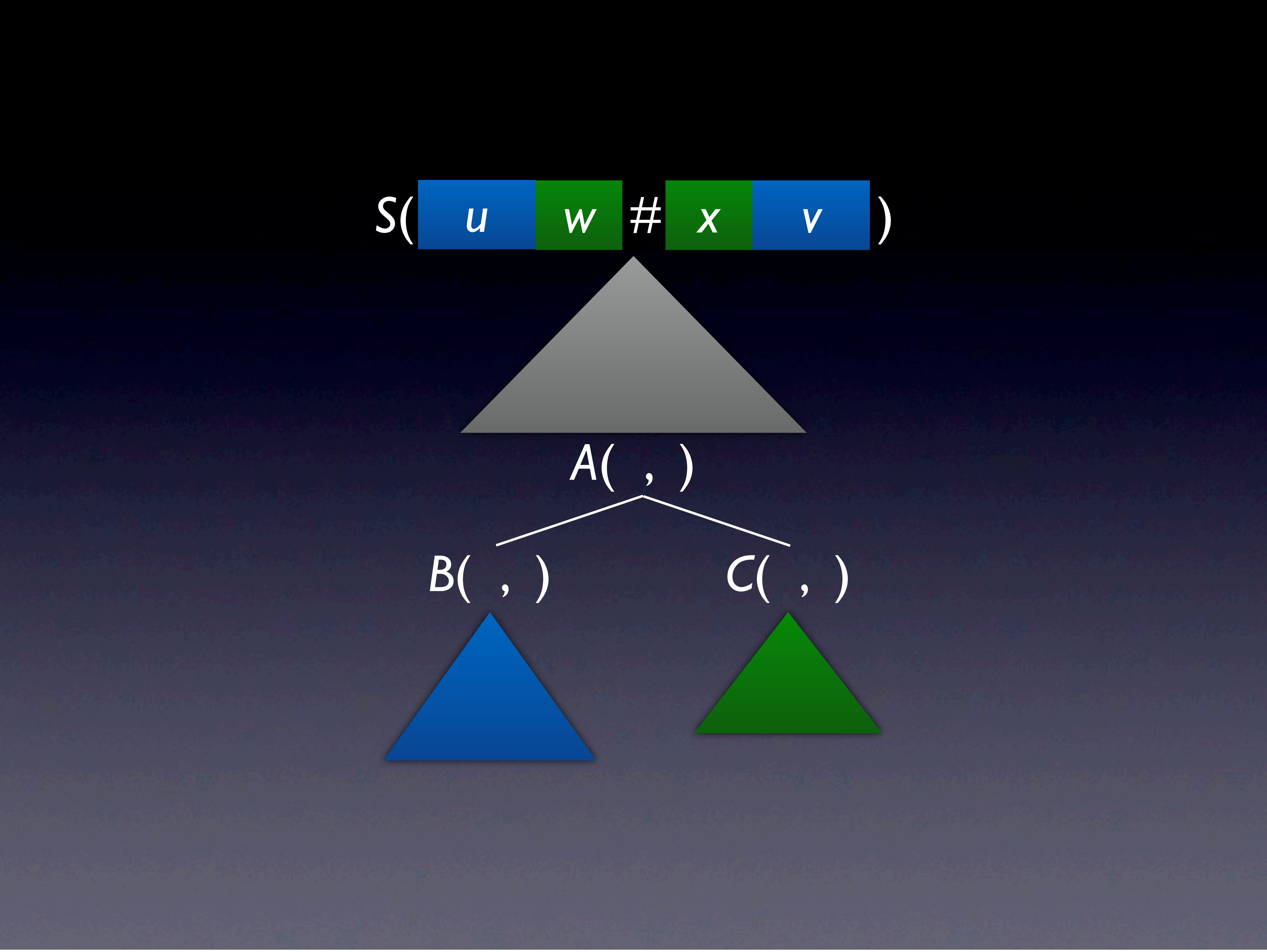
$\{\, w\#w \mid w \in L_0 \,\} \in \mathrm{MCFL}(1)$

$\Updownarrow$

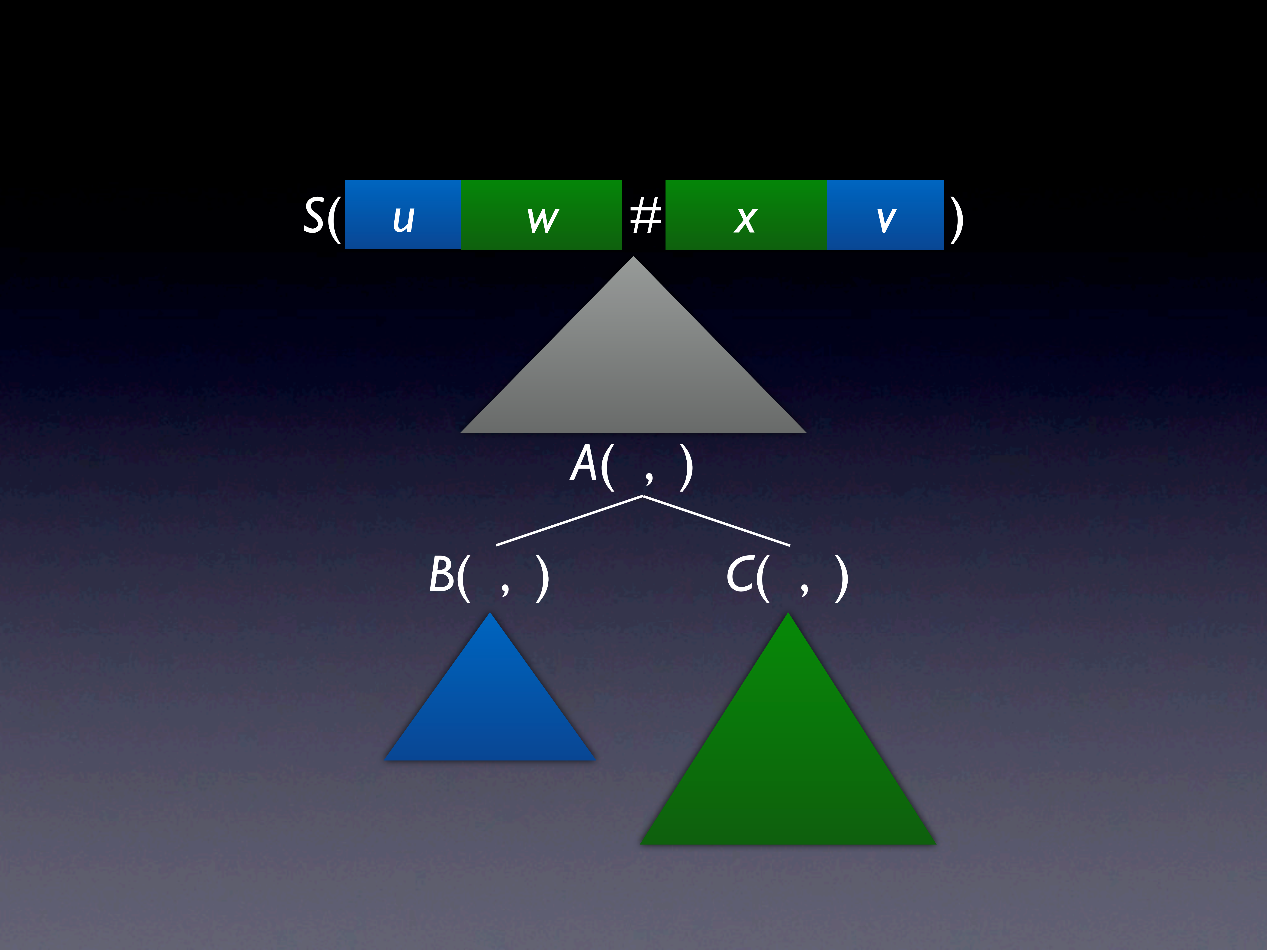
$L_0 \in \mathrm{MCFL}(1)$

$\mathrm{EDT0L}_{\mathrm{FIN}} = \mathrm{MCFL}(1)$

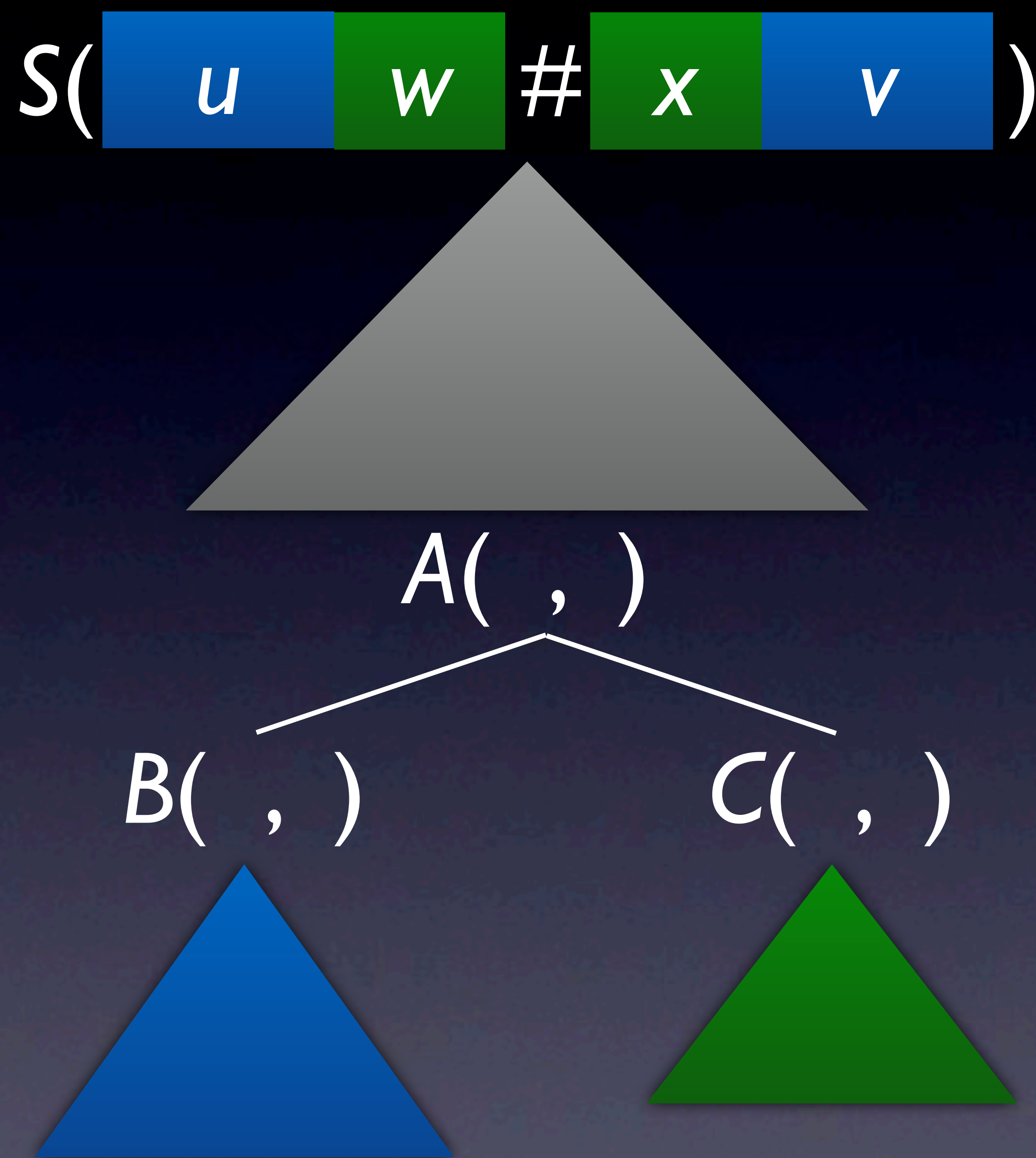
non-branching



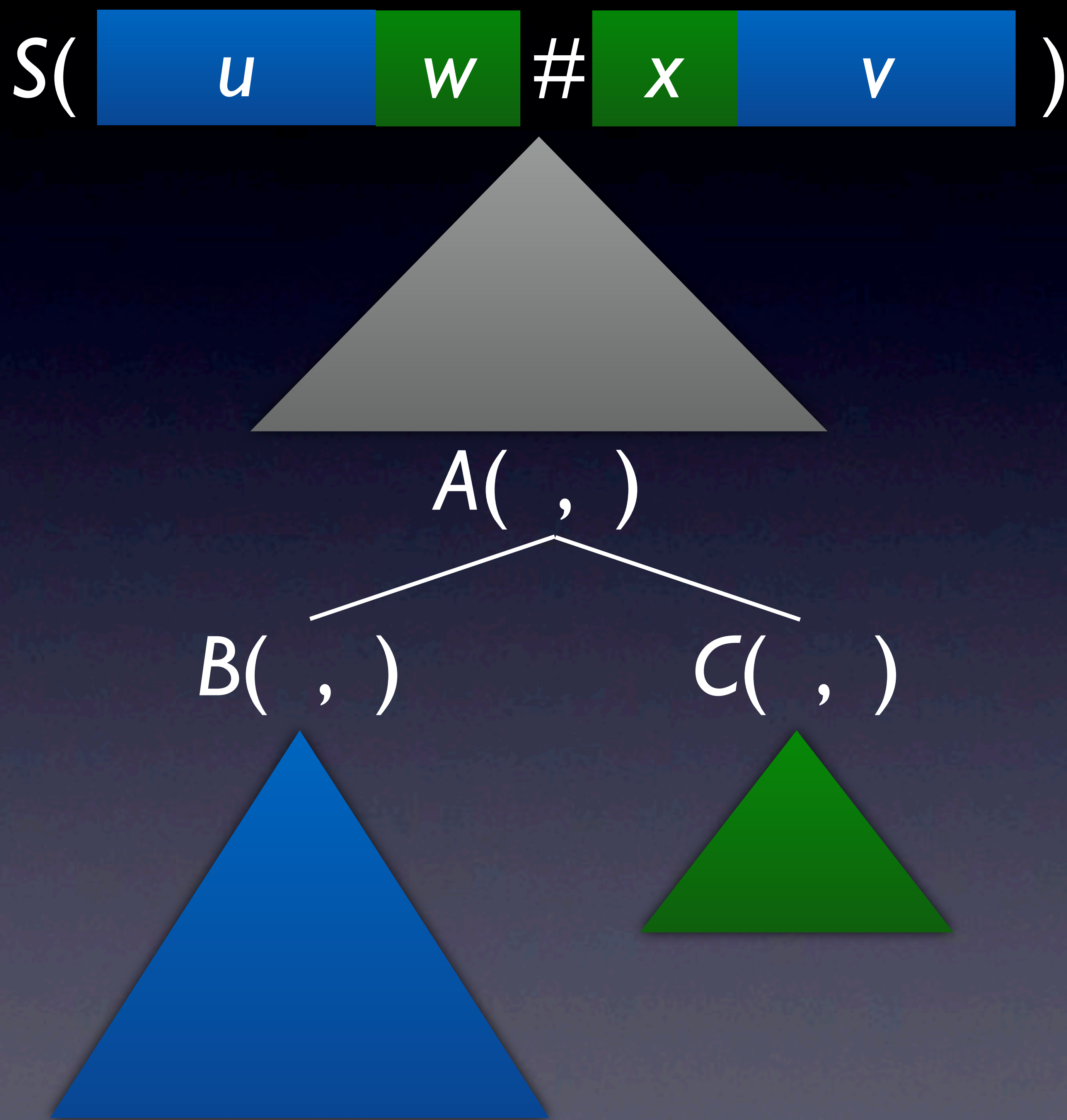
A derivation tree with an instance of a branching rule.
The blue regions and the green regions can vary independently.



S(u w # x v)
A(,)
B(,)
C(,)

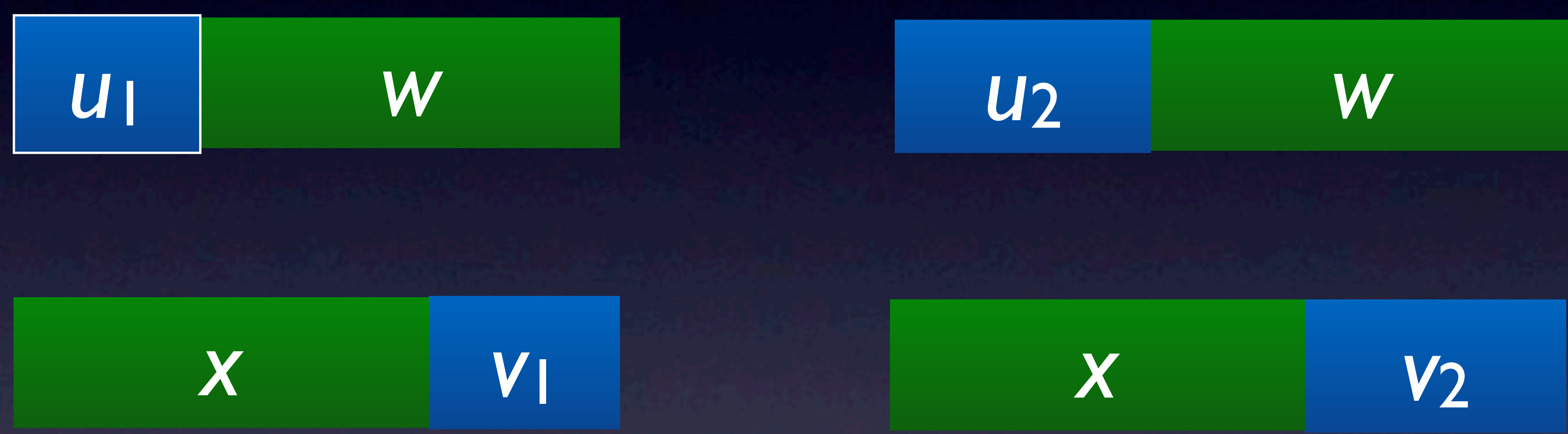


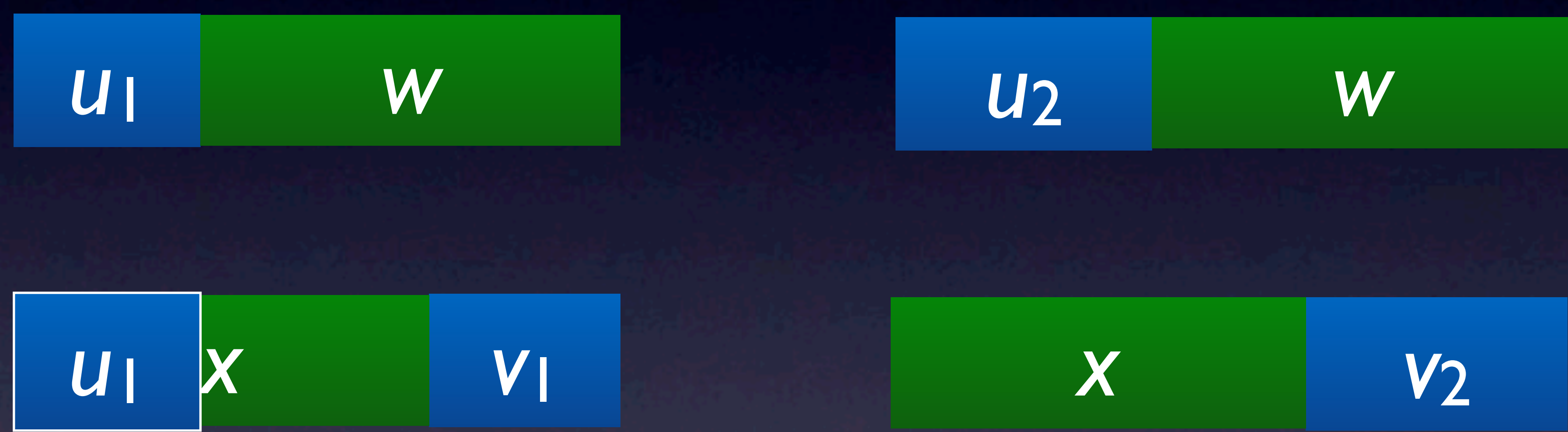
S(u w # x v)
A(,)
B(,)
C(,)

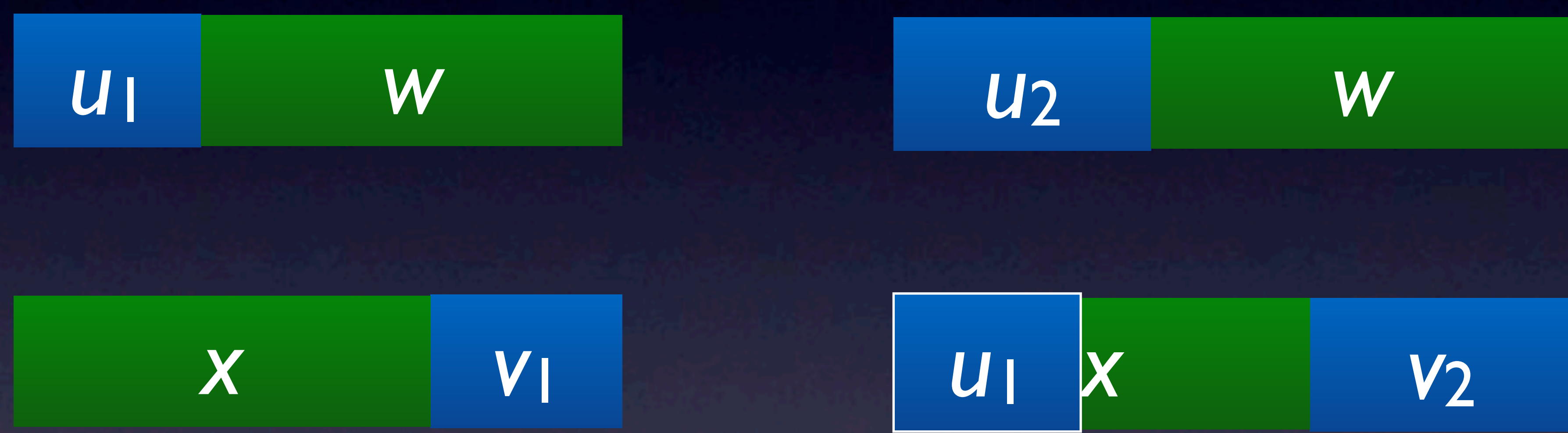

S(u w # x v)
A(,)
B(,)
C(,)

$u_1 \; w \; \# \; x \; v_1$

$u_2 \; w \; \# \; x \; v_2$


u_1
w
u_2
w
x
v_1
x
v_2


u_1
w
u_2
w
u_1
x
v_1
x
v_2


u_1
w
u_2
w
x
v_1
u_1
x
v_2


u_1
w
x
v_1
u_1
w
x
v_2

u_1
w
u_1
$\hat{u}$
w
x
v_1
x
v_2


u_1
w
x
v_1
u_1
$\hat{u}$
w
$\hat{u}$
v_2


u_1
w
u_1
$\hat{u}$
w
$\hat{u}$
v_1
x
v_2

u_1
$\hat{u}$
w
x
v_1
u_1
$\hat{u}$
w
x
v_2


u_1
w
x
v_1
u_1
$\hat{u}$
$\hat{u}$
w
x
v_2


u1
w
u1
û
û
w
x
v1
x
v2

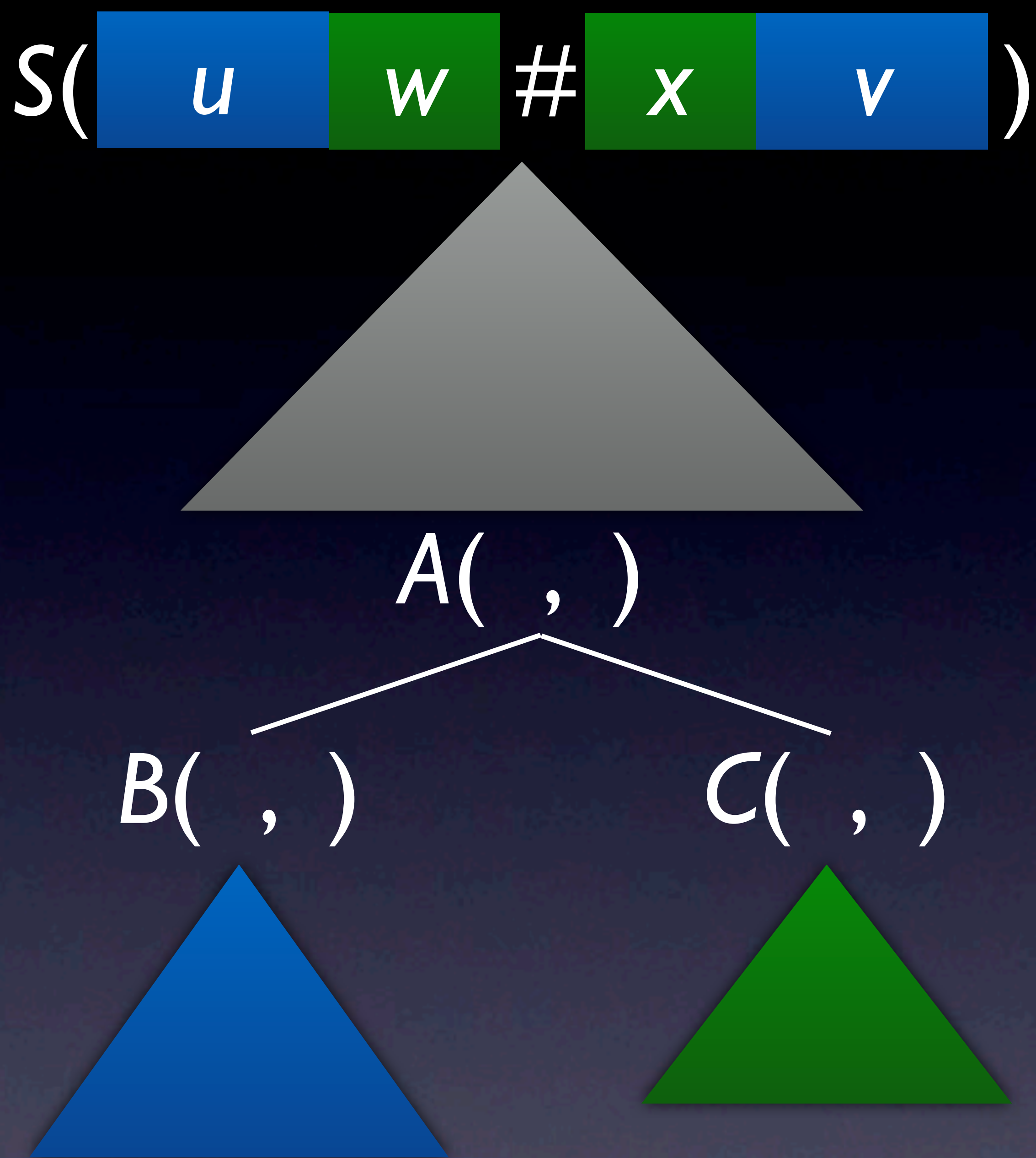

u_1
w
x
v_1
u_1
$\hat{u}$
$\hat{u}$
w
x
$\hat{u}$
v_2

u_1
w
u_1
$\hat{u}$
$\hat{u}$
w
x
$\hat{u}$
v_1
x
v_2


u1
û
w
u1
û
û
w
x
v1
x
v2


u1
w
u1
û
û
û
w
x
v1
x
v2


u_1
w
x
v_1
u_1
$\hat{u}$ $\hat{u}$ $\hat{u}$ $\hat{u}$ v
x
v_2



$$A(r_1 t^{n_1} s_1, \ldots, r_q t^{n_q} s_q)$$

$$A(r_1 t^{n_1} s_1, \ldots, r_{j-1} t^{n_{j-1}} s_{j-1},$$
$$r_j t^{n_j} s_j \# r_{j+1} t^{n_{j+1}} s_{j+1},$$
$$r_{j+2} t^{n_{j+2}} s_{j+2}, \ldots, r_{q+1} t^{n_{q+1}} s_{q+1})$$

$$uw = xv \in r\, t^* s$$

Whenever A(...) is derived using this branching rule, it has one of these special forms.
Can easily write a non-branching MCFG deriving these.

Double Copying Theorem for MCFL_{wn}

$\{\ w\#w \mid w \in L_0\ \} \in \mathrm{MCFL}_{wn}$

$\Updownarrow$

$\{\ w\#w \mid w \in L_0\ \} \in \mathrm{MCFL}(1)$

$\Updownarrow$

$L_0 \in \mathrm{MCFL}(1)$

$\mathrm{EDT0L}_{\mathrm{FIN}} = \mathrm{MCFL}(1)$

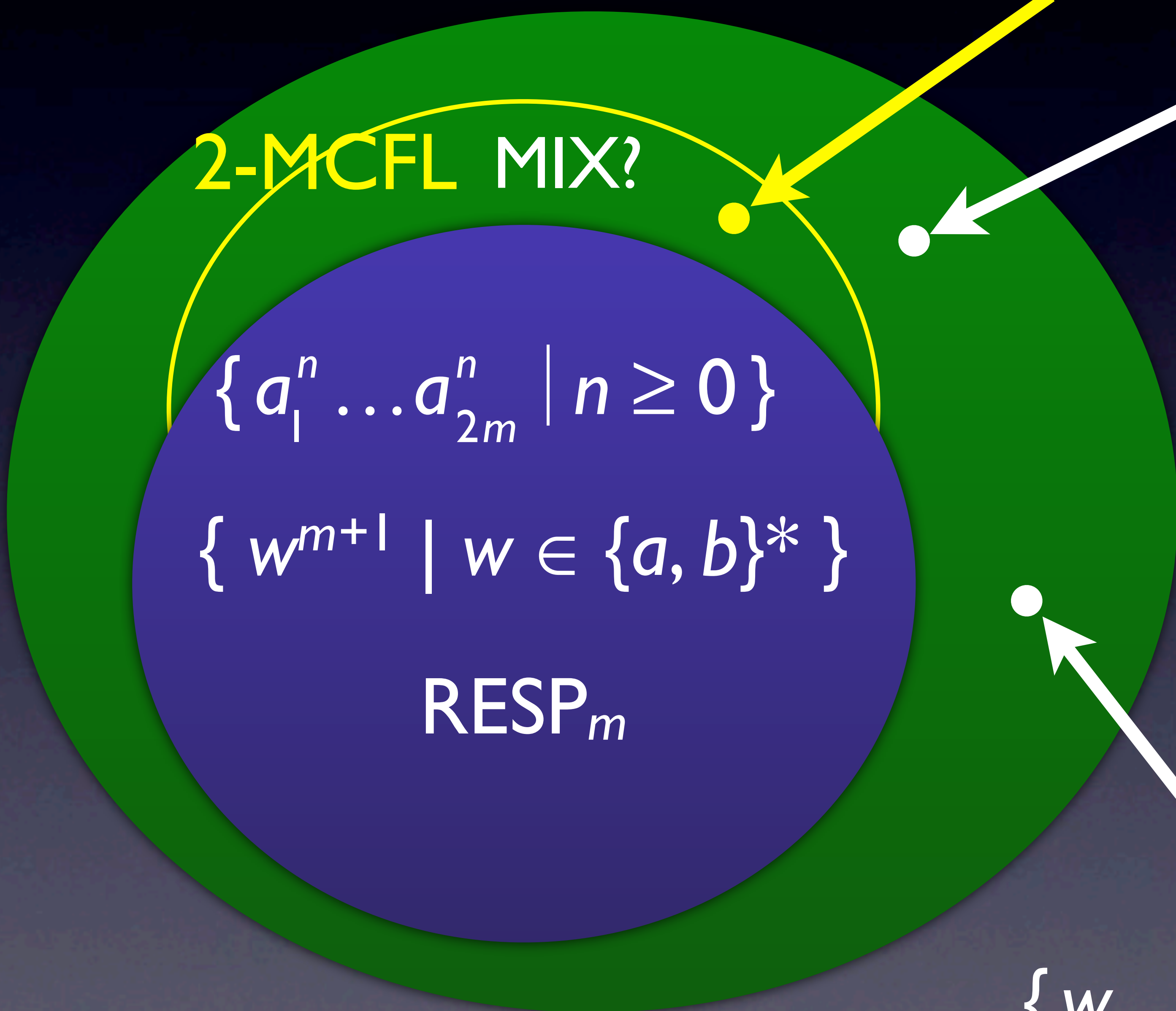
non-branching

MCFL vs. MCFL_{wn}

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

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

Engelfriet and Skyum 1976



Staudacher 1993
Michaelis 2005

$$\{ w_1 \ldots w_n z_n w_n z_{n-1} \ldots z_1 w_1 z_0 w_1^R \ldots w_n^R \mid n \in \mathbb{N}, w_i \in \{c, d\}^+, z_n \ldots z_0 \in D_1^* \}$$

With our theorem, we can see 2-MCFL − $\mathrm{MCFL}_{wn} \neq \emptyset$.
Improves known results.