

D^3 AS A 2-MCFL

Orestis Melkonian, Konstantinos Kogkalidis

August 8, 2018

Utrecht University

2-MCFG

Generalization of the CFG over tuples of strings

2-MCFG

Generalization of the CFG over tuples of strings

N-DIMENSIONAL DYCK LANGUAGE D^N

Defined over an ordered alphabet of N symbols:
 $\{\alpha_1 < \dots < \alpha_N\}$ s.t. words satisfy two conditions:

1. Equal number of occurrences of all alphabet symbols
2. Any prefix of a word must contain at least as many α_i as $\alpha_{i+1} \quad \forall i \leq N - 1$

D^3 - SOME EXAMPLES

DYCK WORDS

- abc
- aabbcc
- abcabcabacbc

D^3 - SOME EXAMPLES

DYCK WORDS

- abc
- aabbcc
- abcabcabacbc

NON-DYCK WORDS

- aabb

D^3 - SOME EXAMPLES

DYCK WORDS

- abc
- aabbcc
- abcabcabacbc

NON-DYCK WORDS

- aabb
- aabbbcc

D^3 - SOME EXAMPLES

DYCK WORDS

- abc
- aabbcc
- abcabcabc

NON-DYCK WORDS

- aabb
- aabbbcc
- abcacb

D^3 - SOME EXAMPLES

DYCK WORDS

- abc
- aabbcc
- abcabcabc

NON-DYCK WORDS

- aabb
- aabbbcc
- abcacb

ababacbcabcc

First-match policy

D^3 - SOME EXAMPLES

DYCK WORDS

- abc
- aabbcc
- abcabcabc

NON-DYCK WORDS

- aabb
- aabbbcc
- abcacb

ababacbcabcc



First-match policy

D^3 - SOME EXAMPLES

DYCK WORDS

- abc
- aabbcc
- abcabcabc

NON-DYCK WORDS

- aabb
- aabbbcc
- abcacb

ababacbcabcc



First-match policy

D^3 - SOME EXAMPLES

DYCK WORDS

- abc
- aabbcc
- abcabcabc

NON-DYCK WORDS

- aabb
- aabbbcc
- abcacb



First-match policy

D^3 - SOME EXAMPLES

DYCK WORDS

- abc
- aabbcc
- abcabcabc

NON-DYCK WORDS

- aabb
- aabbbcc
- abcacb



First-match policy

NATURAL LANGUAGES

Free word order respecting linear order constraints

MOTIVATION

NATURAL LANGUAGES

Free word order respecting linear order constraints

PROGRAMMING LANGUAGES

Static Analysis of non-standard control flows (e.g. *yield*)

G_0 : GRAMMAR OF TRIPLE INSERTIONS

$$S(xy) \leftarrow W(x, y). \quad (1)$$

$$W(\epsilon, xy\mathbf{abc}) \leftarrow W(x, y). \quad (2)$$

$$W(\epsilon, x\mathbf{a}y\mathbf{bc}) \leftarrow W(x, y). \quad (3)$$

...

$$W(\mathbf{ab}x\mathbf{c}y, \epsilon) \leftarrow W(x, y). \quad (60)$$

$$W(\mathbf{abc}x\mathbf{y}, \epsilon) \leftarrow W(x, y). \quad (61)$$

$$W(\epsilon, \mathbf{abc}). \quad (62)$$

$$W(\mathbf{a}, \mathbf{bc}). \quad (63)$$

$$W(\mathbf{ab}, \mathbf{c}). \quad (64)$$

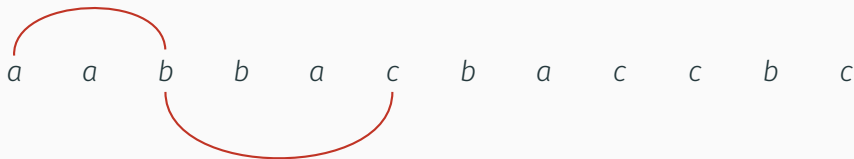
$$W(\mathbf{abc}, \epsilon). \quad (65)$$

G_0 : GRAMMAR OF TRIPLE INSERTIONS

$a \quad a \quad b \quad b \quad a \quad c \quad b \quad a \quad c \quad c \quad b \quad c$

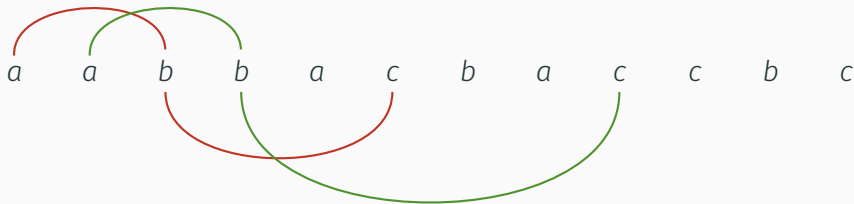
Straddling counter-example

G_0 : GRAMMAR OF TRIPLE INSERTIONS



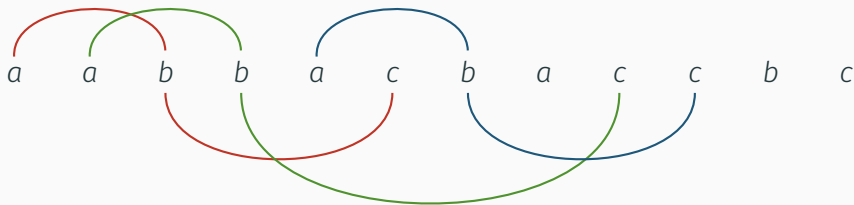
Straddling counter-example

G_0 : GRAMMAR OF TRIPLE INSERTIONS



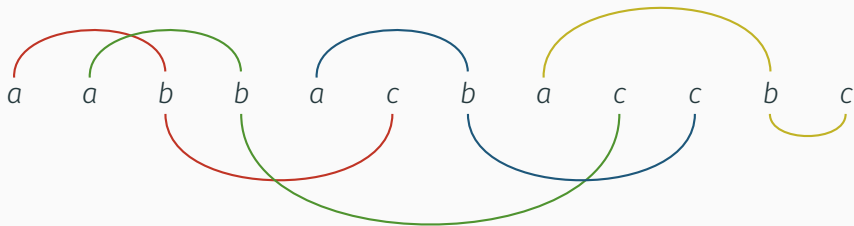
Straddling counter-example

G_0 : GRAMMAR OF TRIPLE INSERTIONS



Straddling counter-example

G_0 : GRAMMAR OF TRIPLE INSERTIONS



Straddling counter-example

NOTATION

$\mathcal{O}_m[\text{conclusion} \leftarrow \text{premises} \mid \text{partial orders}]$.

NOTATION

$\mathcal{O}_m[\text{conclusion} \leftarrow \text{premises} \mid \text{partial orders}]$.

META-GRAMMAR G_1

$\mathcal{O}_2[W \leftarrow \epsilon \mid \{a < b < c\}]$.	} TRIPLE INSERTION
$\mathcal{O}_2[W \leftarrow W_{xy} \mid \{x < y, a < b < c\}]$.	

META-GRAMMARS: INTRODUCTION

NOTATION

$\mathcal{O}_m[\text{conclusion} \leftarrow \text{premises} \mid \text{partial orders}]$.

META-GRAMMAR G_1

$\mathcal{O}_2[W \leftarrow \epsilon \mid \{a < b < c\}].$	}	TRIPLE INSERTION
$\mathcal{O}_2[W \leftarrow W_{xy} \mid \{x < y, a < b < c\}].$		
$+$		
$\mathcal{O}_2[W \leftarrow W_{xy}, W_{zw} \mid \{x < y, z < w\}].$	}	INTERLEAVING WORDS

G₂: ADDING STATES

$$\left. \begin{array}{l} \mathcal{O}_2[A^+ \leftarrow \epsilon \mid \{a\}]. \\ \mathcal{O}_2[B^+ \leftarrow \epsilon \mid \{b\}]. \\ \mathcal{O}_2[C^+ \leftarrow \epsilon \mid \{c\}]. \end{array} \right\} \text{BASE CASES}$$

$$\left. \begin{array}{l} \mathcal{O}_2[C^- \leftarrow A^+, B^+ \mid \{x < y < z < w\}]. \\ \mathcal{O}_2[B^- \leftarrow A^+, C^+ \mid \{x < y < z < w\}]. \\ \mathcal{O}_2[A^- \leftarrow B^+, C^+ \mid \{x < y < z < w\}]. \\ \mathcal{O}_2[A^+ \leftarrow C^-, B^- \mid \{x < y < z < w\}]. \\ \mathcal{O}_2[B^+ \leftarrow C^-, A^- \mid \{x < y < z < w\}]. \\ \mathcal{O}_2[C^+ \leftarrow B^-, A^- \mid \{x < y < z < w\}]. \\ \forall K \in \mathcal{S} \setminus W : \mathcal{O}_2[K \leftarrow K_{xy}, W_{zw} \mid \{x < y, z < w\}]. \end{array} \right\} \text{COMBINATIONS}$$

$$\left. \begin{array}{l} \mathcal{O}_2[W \leftarrow A^+, A^- \mid \{x < y < z < w\}]. \\ \mathcal{O}_2[W \leftarrow C^-, C^+ \mid \{x < y < z < w\}]. \end{array} \right\} \text{CLOSURES}$$

G_3 : $G_2 + \text{UNIVERSAL TRIPLE INSERTION}$

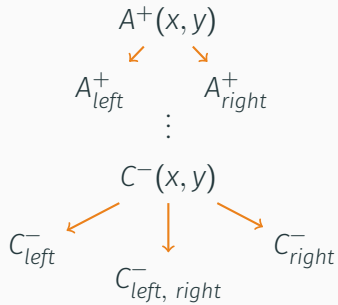
$$G_3 = G_2 + \forall K \in \mathcal{S} \setminus W :$$

$$\mathcal{O}_2 \llbracket K \leftarrow K_{xy} \mid \{x < y, a < b < c\} \rrbracket.$$

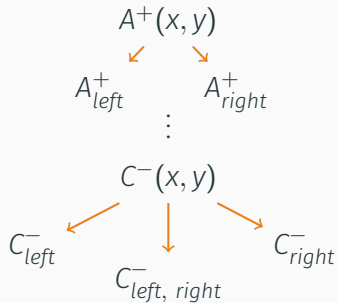


YOU SHALL NOT PARSE!

EXAMPLE



EXAMPLE

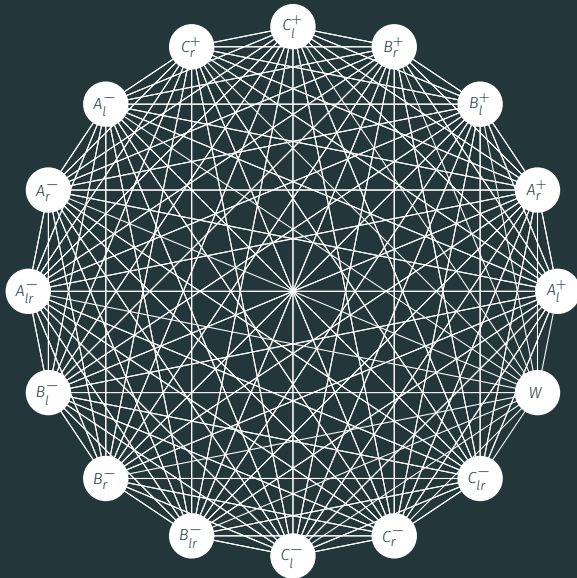


WHY?

NEW ORDERS IN INTERACTIONS

$$C^-(xz, wy) \leftarrow A^+_{left}(x, y), B^+(z, w).$$

REFINING STATES: INTERACTIONS



STATE DESCRIPTORS \mathcal{D}

$$W \mapsto (\epsilon, \epsilon)$$

$$A_l^+ \mapsto (a, \epsilon)$$

$$A_r^+ \mapsto (\epsilon, a)$$

$$B_l^+ \mapsto (b, \epsilon)$$

$$B_r^+ \mapsto (\epsilon, b)$$

$$C_l^+ \mapsto (c, \epsilon)$$

$$C_r^+ \mapsto (\epsilon, c)$$

$$A_l^- \mapsto (bc, \epsilon)$$

$$A_r^- \mapsto (\epsilon, bc)$$

$$A_{lr}^- \mapsto (b, c)$$

$$B_l^- \mapsto (ac, \epsilon)$$

$$B_r^- \mapsto (\epsilon, ac)$$

$$B_{lr}^- \mapsto (a, c)$$

$$C_l^- \mapsto (ab, \epsilon)$$

$$C_r^- \mapsto (\epsilon, ab)$$

$$C_{lr}^- \mapsto (a, b)$$

AUTOMATIC RULE INFERENCE: EXAMPLE

CASE OF $A_{lr}^-(x_b, y_c) + B_{lr}^-(z_a, w_c)$

PERMUTATION

\vdots

(zxw, y)

\vdots

$(xz w, y)$

\vdots

AUTOMATIC RULE INFERENCE: EXAMPLE

CASE OF $A_{lr}^-(x_b, y_c) + B_{lr}^-(z_a, w_c)$

PERMUTATION

DESCRIPTOR

\vdots

\vdots

$(zxw, y) \longrightarrow (abc, c)$

\vdots

\vdots

$(xzw, y) \longrightarrow (bac, c)$

\vdots

\vdots

AUTOMATIC RULE INFERENCE: EXAMPLE

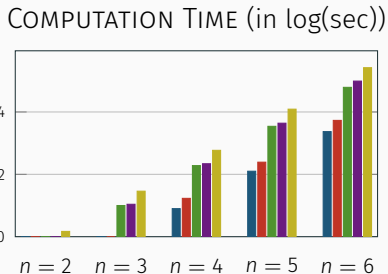
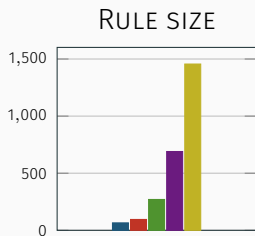
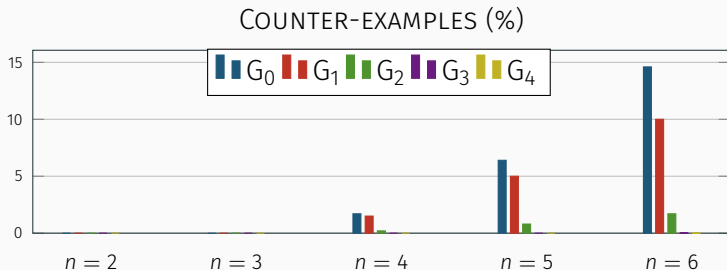
CASE OF $A_{lr}^-(x_b, y_c) + B_{lr}^-(z_a, w_c)$

<u>PERMUTATION</u>	<u>DESCRIPTOR</u>	<u>ELIMINATED</u>
\vdots	\vdots	\vdots
(zxw, y)	$\longrightarrow (abc, c)$	$\begin{array}{l} \nearrow (c, \epsilon) \leftarrow C_l^+ \\ \searrow (\epsilon, c) \leftarrow C_r^+ \end{array}$
\vdots	\vdots	\vdots
$(xz w, y)$	$\longrightarrow (bac, c)$	$\longrightarrow (bac, c) \notin \mathcal{D}$
\vdots	\vdots	\vdots

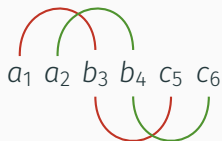
DSL IN PYTHON (\$ pip install dyck)

```
from dyck import *
G3 = Grammar(initial='W',
    # Base Cases
    O('W', {(a, b, c)}),
    O('A-', {(b, c)}), O('B-', {(a, c)}), O('C-', {(a, b)}),
    O('A+', {(a,)}), O('B+', {(b,)}), O('C+', {(c,)}),
    # Combinations
    O('C- <- A+, B+', {(x, y, z, w)}),
    O('B- <- A+, C+', {(x, y, z, w)}),
    O('A- <- B+, C+', {(x, y, z, w)}),
    O('C+ <- B-, A-', {(x, y, z, w)}),
    O('B+ <- C-, A-', {(x, y, z, w)}),
    O('A+ <- C-, B-', {(x, y, z, w)}),
    forall(all_states, lambda K: O('K <- K, W', {(x, y), (z, w)})),
    # Closures
    O('W <- A+, A-', {(x, y, z, w)}),
    O('W <- C-, C+', {(x, y, z, w)}),
    # Universal Triple Insertion
    forall(all_states, lambda K: O('K <- K', {(x, y), (a, b, c)})))
```

RESULTS

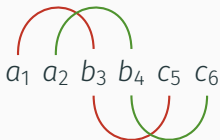


CORRESPONDENCES: PROMOTION ON YOUNG TABLEAUX



1	2
3	4
5	6

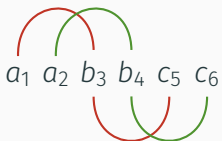
CORRESPONDENCES: PROMOTION ON YOUNG TABLEAUX



1	2
3	4
5	6

•	1
2	3
4	5

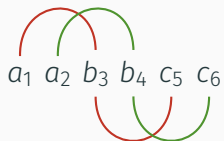
CORRESPONDENCES: PROMOTION ON YOUNG TABLEAUX



1	2
3	4
5	6

1	•
2	3
4	5

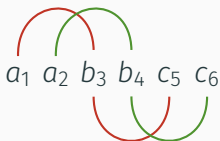
CORRESPONDENCES: PROMOTION ON YOUNG TABLEAUX



1	2
3	4
5	6

1	3
2	•
4	5

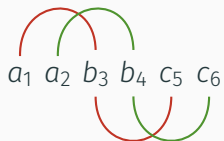
CORRESPONDENCES: PROMOTION ON YOUNG TABLEAUX



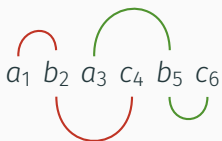
1	2
3	4
5	6

1	3
2	5
4	•

CORRESPONDENCES: PROMOTION ON YOUNG TABLEAUX



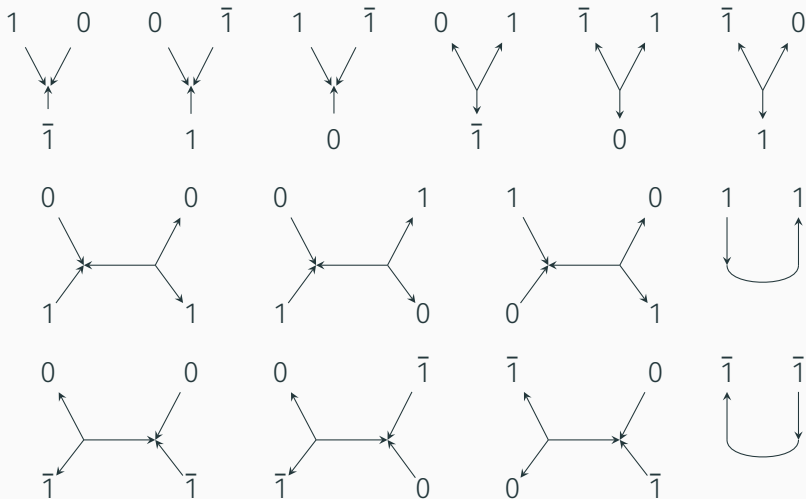
1	2
3	4
5	6



1	3
2	5
4	6

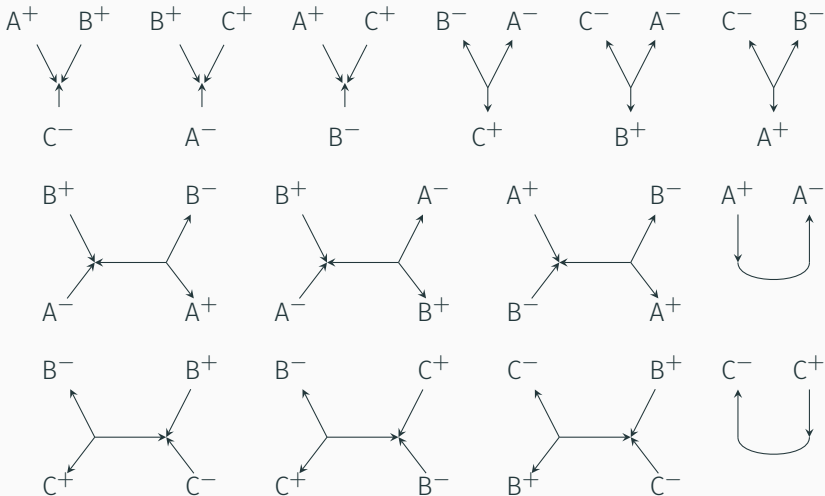
CORRESPONDENCES: SPIDER WEBS

GROWTH RULES



CORRESPONDENCES: SPIDER WEBS

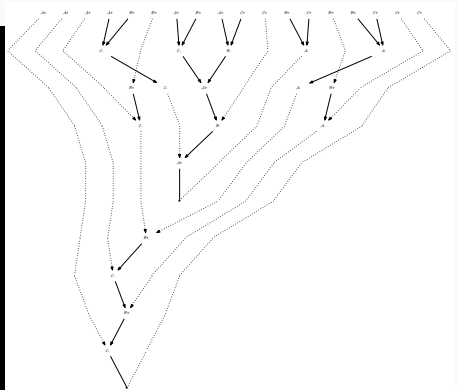
GROWTH RULES



PYTHON VISUALIZATION PACKAGE (\$ pip install dyckviz)

```
λ dyckviz --word 'aaaabbabaccbcbccccc'
```

1.	a	a	b	b	a	c	c	b	a	b	a	c	b	c				
2.	a	a	b	b	a	b	a	c	c	b	c	a	a	b	b	c	c	
3.	a	b	a	c	b	a	c	a	a	b	a	c	b	b	c	c		
4.	a	b	a	a	b	b	a	c	c	b	a	c	a	b	b	c	c	
5.	a	a	a	b	b	a	b	a	c	c	b	c	b	a	c	b	c	
6.	a	b	a	a	c	a	a	b	a	c	c	b	b	c	b	c		
7.	a	a	b	a	a	b	b	a	c	c	b	b	c	a	c	b	c	
8.	a	b	a	a	a	c	b	a	b	a	c	c	b	b	b	c	c	
9.	a	b	a	a	a	b	a	c	c	b	c	a	a	b	b	c	c	
10.	a	a	b	a	c	b	b	c	a	a	b	b	a	c	a	c	b	c
11.	a	a	b	b	a	c	a	a	b	a	c	b	b	c	b	c	c	
12.	a	a	a	b	a	c	c	b	a	c	b	a	b	b	c	c		
13.	a	b	a	a	c	b	b	a	c	b	a	b	a	c	b	c	c	
14.	a	a	b	b	a	a	c	b	a	b	a	c	c	b	c	b	c	
15.	a	a	a	b	a	a	b	a	c	c	b	b	c	b	c	c	c	
16.	a	a	a	b	b	b	a	c	b	a	b	a	c	b	b	c	c	
17.	a	a	a	a	b	b	a	b	a	c	c	b	c	b	b	c	c	
18.	a	a	a	b	a	c	b	b	c	a	a	b	b	b	c	c	c	



CONCLUSION

- Conjecture still open :(
- Lots of fun along the way
- We are confident we have a complete 3-MCFG, though
 - Currently mechanizing the proof using coq

QUESTIONS?