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

Introduction/Definition

2-MCFG

Generalization of the CFG over tuples of strings

N-dimensional Dyck Language D^N

Defined over an ordered alphabet of N symbols:
$\{\alpha_1 < \cdots < \alpha_N\}$ 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
- abcabcabacbc

Non-dyck words

- aabb
- aabbbcc
- abcacb

D^3 - Some examples

Dyck words

- abc
- aabbcc
- abcabcabacbc

Non-dyck words

- aabb
- aabbbcc
- abcacb

$$a\,b\,a\,b\,a\,c\,b\,c\,a\,b\,c\,c$$

First-match policy

D^3 - Some examples

Dyck words

- abc
- aabbcc
- abcabcabacbc

Non-dyck words

- aabb
- aabbbcc
- abcacb

$$a\,b\,a\,b\,a\,c\,b\,c\,a\,b\,c\,c$$

First-match policy

D^3 - Some examples

Dyck words

- abc
- aabbcc
- abcabcabacbc

Non-dyck words

- aabb
- aabbbcc
- abcacb

$$a\,b\,a\,b\,a\,c\,b\,c\,a\,b\,c\,c$$

First-match policy

D^3 - Some examples

Dyck words

- abc
- aabbcc
- abcabcabacbc

Non-dyck words

- aabb
- aabbbcc
- abcacb

First-match policy

D^3 - Some examples

Dyck words

- abc
- aabbcc
- abcabcabacbc

Non-dyck words

- aabb
- aabbbcc
- abcacb

First-match policy

Motivation

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

$$S(xy) \leftarrow W(x, y). \tag{1}$$

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

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

$$\dots$$

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

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

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

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

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

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

G_0: Grammar of triple insertions

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

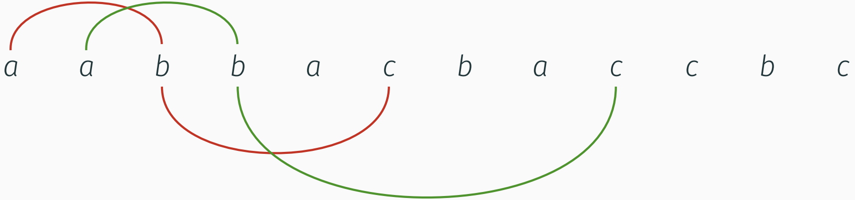
Straddling counter-example

G_0: Grammar of triple insertions

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

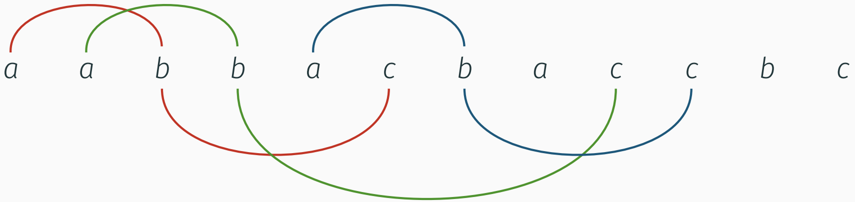
Straddling counter-example

G_0: Grammar of triple insertions



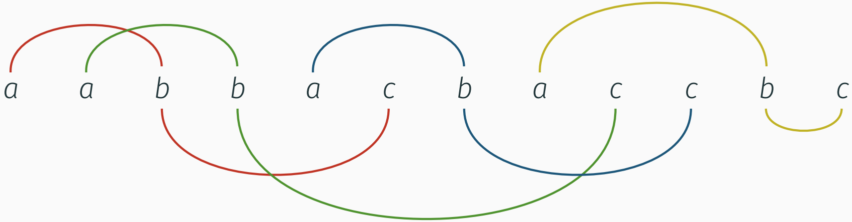
Straddling counter-example

G_0: Grammar of triple insertions



Straddling counter-example

G_0: Grammar of triple insertions



Straddling counter-example

Notation

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

Meta-grammars: Introduction

Notation

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

Meta-grammar G_1

$$\left.\begin{array}{l} \mathcal{O}_2[\![W \leftarrow \epsilon \mid \{a < b < c\}]\!]. \\ \mathcal{O}_2[\![W \leftarrow W_{xy} \mid \{x < y,\ a < b < c\}]\!]. \end{array}\right\} \quad \begin{array}{c} \text{triple} \\ \text{insertion} \end{array}$$

Meta-grammars: Introduction

Notation

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

Meta-grammar G_1

$$\left.\begin{array}{l}\mathcal{O}_2[\![W \leftarrow \epsilon \mid \{a < b < c\}]\!].\\ \mathcal{O}_2[\![W \leftarrow W_{xy} \mid \{x < y,\ a < b < c\}]\!].\end{array}\right\}$$

TRIPLE INSERTION

+

$$\mathcal{O}_2[\![W \leftarrow W_{xy}, W_{zw} \mid \{x < y,\ z < w\}]\!].\ \}$$

INTERLEAVING WORDS

G_2: Adding states

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

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

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

G_3: G_2 + Universal triple insertion

$$\mathsf{G}_3 = \mathsf{G}_2 + \forall\, \mathsf{K} \in \mathcal{S} \setminus \mathsf{W} : \\ \mathcal{O}_2[\![\mathsf{K} \leftarrow \mathsf{K}_{xy} \mid \{x < y,\ a < b < c\}]\!].$$

YOU SHALL NOT PARSE!

Refining states

Example

$$
\begin{array}{ccc}
 & A^+(x,y) & \\
\swarrow & & \searrow \\
A^+_{left} & & A^+_{right} \\
 & \vdots & \\
 & C^-(x,y) & \\
\swarrow & \downarrow & \searrow \\
C^-_{left} & C^-_{left,\ right} & C^-_{right}
\end{array}
$$

Refining states

Example

$$
\begin{tikzcd}
 & A^{+}(x, y) \arrow[dl] \arrow[dr] & \\
A^{+}_{left} & & A^{+}_{right}
\end{tikzcd}
$$

$$\vdots$$

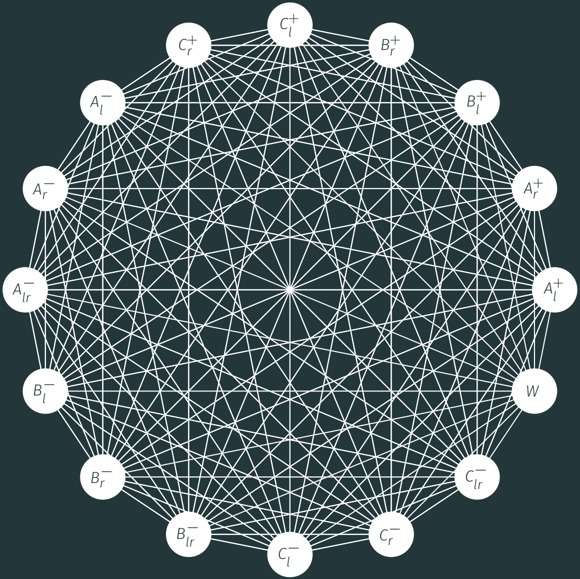
$$
\begin{tikzcd}
 & C^{-}(x, y) \arrow[dl] \arrow[d] \arrow[dr] & \\
C^{-}_{left} & C^{-}_{left,\ right} & C^{-}_{right}
\end{tikzcd}
$$

Why?

New orders in interactions

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

Refining states: Interactions



G_4: Automatic Rule Inference

State descriptors $\mathcal{D}$

$$\mathrm{W} \mapsto (\epsilon, \epsilon)$$
$$\mathrm{A}_l^+ \mapsto (a, \epsilon)$$
$$\mathrm{A}_r^+ \mapsto (\epsilon, a)$$
$$\mathrm{B}_l^+ \mapsto (b, \epsilon)$$
$$\mathrm{B}_r^+ \mapsto (\epsilon, b)$$
$$\mathrm{C}_l^+ \mapsto (c, \epsilon)$$
$$\mathrm{C}_r^+ \mapsto (\epsilon, c)$$
$$\mathrm{A}_l^- \mapsto (bc, \epsilon)$$
$$\mathrm{A}_r^- \mapsto (\epsilon, bc)$$
$$\mathrm{A}_{lr}^- \mapsto (b, c)$$
$$\mathrm{B}_l^- \mapsto (ac, \epsilon)$$
$$\mathrm{B}_r^- \mapsto (\epsilon, ac)$$
$$\mathrm{B}_{lr}^- \mapsto (a, c)$$
$$\mathrm{C}_l^- \mapsto (ab, \epsilon)$$
$$\mathrm{C}_r^- \mapsto (\epsilon, ab)$$
$$\mathrm{C}_{lr}^- \mapsto (a, b)$$

Automatic Rule Inference: Example

$$\text{Case of } \mathrm{A}^{-}_{lr}(x_{b},\ y_{c}) + \mathrm{B}^{-}_{lr}(z_{a},\ w_{c})$$

PERMUTATION

$$\vdots$$

$$(zxw,\ y)$$

$$\vdots$$

$$(xzw,\ y)$$

$$\vdots$$

Automatic Rule Inference: Example

$$\text{Case of } A^{-}_{lr}(x_b,\ y_c) + B^{-}_{lr}(z_a,\ w_c)$$

Permutation		Descriptor
$\vdots$		$\vdots$
$(zxw,\ y)$	$\longrightarrow$	$(abc,\ c)$
$\vdots$		$\vdots$
$(xzw,\ y)$	$\longrightarrow$	$(bac,\ c)$
$\vdots$		$\vdots$

Automatic Rule Inference: Example

Case of $A^{-}_{lr}(x_b,\ y_c) + B^{-}_{lr}(z_a,\ w_c)$

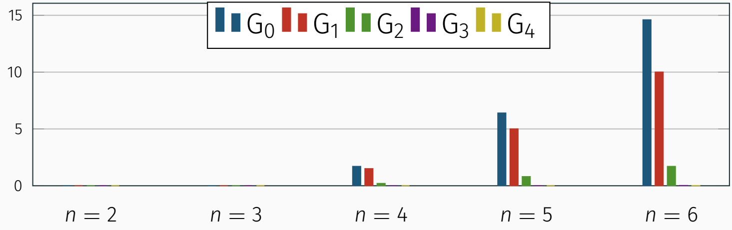
Permutation		Descriptor		Eliminated
$\vdots$		$\vdots$		$\vdots$
				$(c,\ \epsilon) \mapsfrom C^{+}_{l}$
$(zxw,\ y)$	$\longrightarrow$	$(abc,\ c)$	$\rightarrow$	
				$(\epsilon,\ c) \mapsfrom C^{+}_{r}$
$\vdots$		$\vdots$		$\vdots$
$(xzw,\ y)$	$\longrightarrow$	$(bac,\ c)$	$\longrightarrow$	$(bac,\ c) \notin \mathcal{D}$
$\vdots$		$\vdots$		$\vdots$

DSL in Python (`$ pip install dyck`)

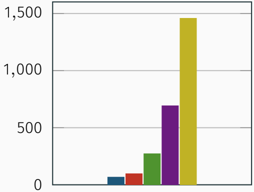
```python
from dyck import *
G3 = Grammar(initial='W',
  # Base Cases
  O('W', {(a, b, c)}),
  O('A-', {(b, c)}), O('B-', {(a, c)}), O('C-', {(a, b)}),
  O('A+', {(a,)}),   O('B+', {(b,)}),   O('C+', {(c,)}),
  # Combinations
  O('C- <- A+, B+', {(x, y, z, w)}),
  O('B- <- A+, C+', {(x, y, z, w)}),
  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

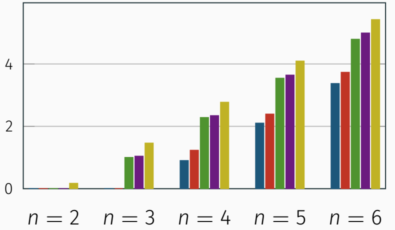
Counter-examples (%)



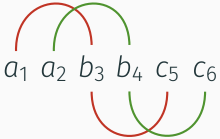
Rule size



Computation Time (in log(sec))



Correspondences: Promotion on Young Tableaux



1	2
3	4
5	6

·	1
2	3
4	5

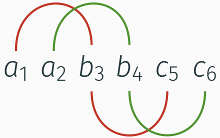
Correspondences: Promotion on Young Tableaux

$$a_1 \quad a_2 \quad b_3 \quad b_4 \quad c_5 \quad c_6$$

1	2
3	4
5	6

1	.
2	3
4	5

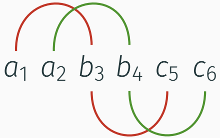
Correspondences: Promotion on Young Tableaux



1	2
3	4
5	6

1	3
2	.
4	5

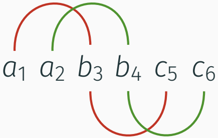
Correspondences: Promotion on Young Tableaux



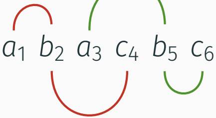
1	2
3	4
5	6

1	3
2	5
4	.

Correspondences: Promotion on Young Tableaux



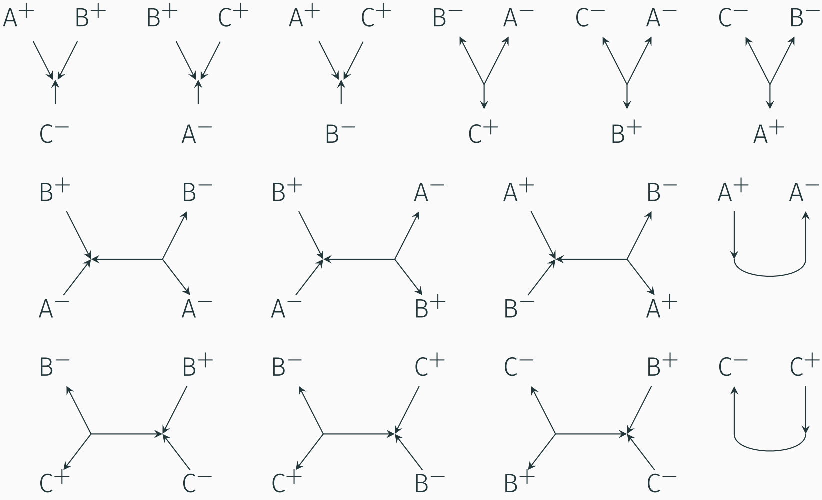
1	2
3	4
5	6



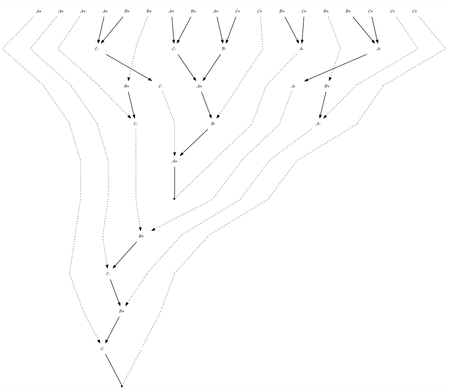
1	3
2	5
4	6

Correspondences: Spider Webs

Growth rules



Python Visualization Package (`$ pip install dyckviz`)



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?