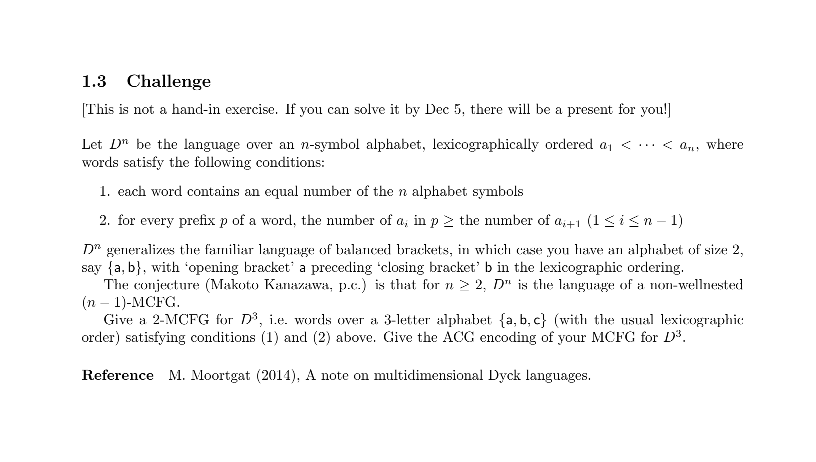
D^3 as a 2-MCFL

Orestis Melkonian, Konstantinos Kogkalidis

January 25, 2018

Universiteit Utrecht



1.3 Challenge

[This is not a hand-in exercise. If you can solve it by Dec 5, there will be a present for you!]

Let D^n be the language over an n-symbol alphabet, lexicographically ordered $a_1 < \cdots < a_n$, where words satisfy the following conditions:

1. each word contains an equal number of the n alphabet symbols
2. for every prefix p of a word, the number of a_i in $p \geq$ the number of a_{i+1} $(1 \leq i \leq n-1)$

D^n generalizes the familiar language of balanced brackets, in which case you have an alphabet of size 2, say $\{\mathsf{a}, \mathsf{b}\}$, with 'opening bracket' a preceding 'closing bracket' b in the lexicographic ordering.

The conjecture (Makoto Kanazawa, p.c.) is that for $n \geq 2$, D^n is the language of a non-wellnested $(n-1)$-MCFG.

Give a 2-MCFG for D^3, i.e. words over a 3-letter alphabet $\{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ (with the usual lexicographic order) satisfying conditions (1) and (2) above. Give the ACG encoding of your MCFG for D^3.

Reference M. Moortgat (2014), A note on multidimensional Dyck languages.

Some examples

Dyck words

- abc
- aabbcc
- abcabcabacbc

Some examples

Dyck words

- abc
- aabbcc
- abcabcabacbc

Non-dyck words

- aabb

Some examples

Dyck words

- abc
- aabbcc
- abcabcabacbc

Non-dyck words

- aabb
- aabbbcc

Some examples

Dyck words

- abc
- aabbcc
- abcabcabacbc

Non-dyck words

- aabb
- aabbbcc
- abcacb

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

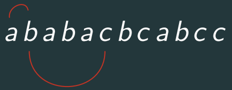
Some examples

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

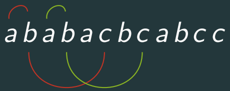
Some examples

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

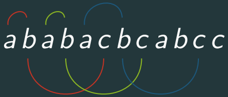
Some examples

Dyck words

- abc
- aabbcc
- abcabcabacbc

Non-dyck words

- aabb
- aabbbcc
- abcacb



First-match policy

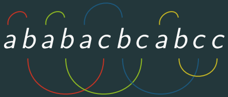
Some examples

Dyck words

- abc
- aabbcc
- abcabcabacbc

Non-dyck words

- aabb
- aabbbcc
- abcacb



First-match policy

Grammar of triple insertions

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

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

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

$$\ldots\ldots$$

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

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

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

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

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

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

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

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

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

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

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

Overview

- Background
- $G0$: Triple insertion
- Meta-grammar notation
- G_0': Triple insertion (in O_2 notation)
- G_1: G0' + interleavings
- G_2: incomplete words
- G_3: G2 + 3-ins
- DEMO: **dyck**
- Refined states
- Constraints (notation)
- ARIS
- Results
- Road to completeness
- Correspondences
- DEMO: **dyckviz**

Theoretical background

Modelling techniques

Implemented grammars

Road to completeness

Tools

Demo