(Un)decidability of the Emptiness Problem for Multi-dimensional Context-free Grammars

Daniel Průša

Czech Technical University in Prague

20th International Conference on Implementation and Application of Automata

August 18 – 21, 2015, Umeå, Sweden

Introduction

- Languages of multi-dimensional objects.
- Studied from 60's.
  - four-way finite automaton on a 2D tape [Blum and Hewitt 1967]
  - several models of multi-dimensional automata and grammars proposed since then
- The more complex topology changes a lot properties of recognized/generated languages.
- We demonstrate this for basic models of 2D and 3D context-free grammars, the main topic is the emptiness problem.

Introduction

- Languages of multi-dimensional objects.
- Studied from 60's.
  - four-way finite automaton on a 2D tape [Blum and Hewitt 1967]
  - several models of multi-dimensional automata and grammars proposed since then
- The more complex topology changes a lot properties of recognized/generated languages.
- We demonstrate this for basic models of 2D and 3D context-free grammars, the main topic is the emptiness problem.

Introduction

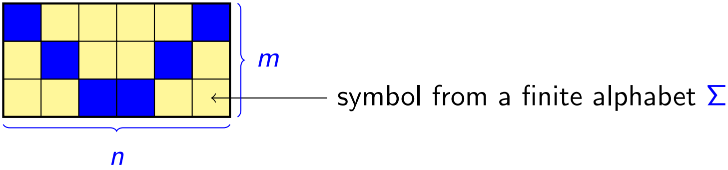
- Languages of multi-dimensional objects.
- Studied from 60's.
  - four-way finite automaton on a 2D tape [Blum and Hewitt 1967]
  - several models of multi-dimensional automata and grammars proposed since then
- The more complex topology changes a lot properties of recognized/generated languages.
- We demonstrate this for basic models of 2D and 3D context-free grammars, the main topic is the emptiness problem.

Introduction

- Languages of multi-dimensional objects.
- Studied from 60's.
  - four-way finite automaton on a 2D tape [Blum and Hewitt 1967]
  - several models of multi-dimensional automata and grammars proposed since then
- The more complex topology changes a lot properties of recognized/generated languages.
- We demonstrate this for basic models of 2D and 3D context-free grammars, the main topic is the emptiness problem.

Pictures, picture languages

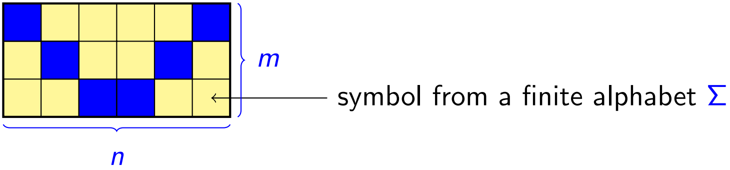
- P a picture of size $m \times n$, $P \in \Sigma^{m,n}$



- $\Sigma^{*,*}$ – the set all pictures over Σ.
- $L \subseteq \Sigma^{*,*}$ – a picture language.

Pictures, picture languages

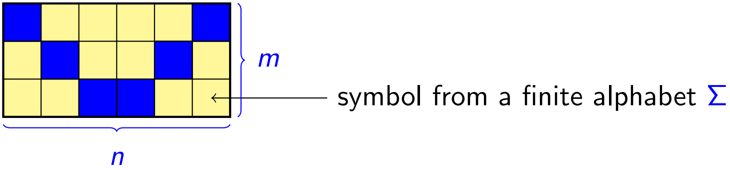
- P a picture of size $m \times n$, $P \in \Sigma^{m,n}$



- $\Sigma^{*,*}$ – the set all pictures over Σ.
- $L \subseteq \Sigma^{*,*}$ – a picture language.

Pictures, picture languages

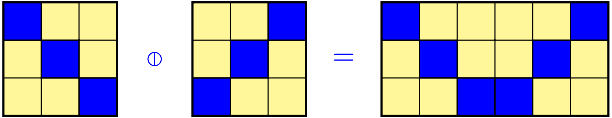
- P a picture of size $m \times n$, $P \in \Sigma^{m,n}$



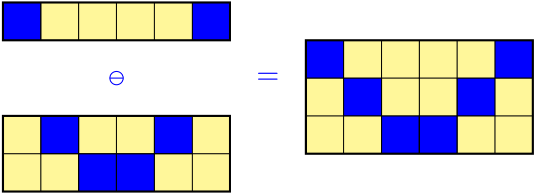
- $\Sigma^{*,*}$ – the set all pictures over Σ.
- $L \subseteq \Sigma^{*,*}$ – a picture language.

Concatenation

column concatenation ($P_1 \obar P_2$):



row concatenation ($P_3 \ominus P_4$):



2D Kolam grammar

2D Kolam type context-free grammar (2KG)

- [Siromoney et al. 1973, Schlesinger 1989, Matz 1997]
- early model of context-free picture grammar

$\mathcal{G} = (V_N, V_T, \mathcal{P}, S_0)$, where

V_N is a set of nonterminals
V_T is a set of terminals
$\mathcal{P}$ is a set of productions
$S_0 \in V_N$ is the initial nonterminal

Production types:

$$N \rightarrow a, \quad N \rightarrow AB, \quad N \rightarrow \begin{matrix} A \\ B \end{matrix}, \quad \text{(optionally) } S_0 \rightarrow \Lambda$$

- a – terminal; N, A, B – nonterminals; Λ – empty picture

Generated picture language

$L(\mathcal{G}, N)$ .. a picture language generated by a nonterminal N

1. $N \rightarrow a \Rightarrow a \in L(\mathcal{G}, N)$
2. $N \rightarrow A\,B$, $P_A \in L(\mathcal{G}, A)$, $P_B \in L(\mathcal{G}, B) \Rightarrow P_A \varobar P_B \in L(\mathcal{G}, N)$
3. $N \rightarrow \begin{matrix} A \\ B \end{matrix}$, $P_A \in L(\mathcal{G}, A)$, $P_B \in L(\mathcal{G}, B) \Rightarrow P_A \ominus P_B \in L(\mathcal{G}, N)$

$L(\mathcal{G}) = L(\mathcal{G}, S_0)$ – picture language generated by $\mathcal{G}$

Generated picture language

$L(\mathcal{G}, N)$ .. a picture language generated by a nonterminal N

1. $N \rightarrow a \Rightarrow a \in L(\mathcal{G}, N)$
2. $N \rightarrow A\,B$, $P_A \in L(\mathcal{G}, A)$, $P_B \in L(\mathcal{G}, B) \Rightarrow P_A \oplus P_B \in L(\mathcal{G}, N)$
3. $N \rightarrow \begin{matrix} A \\ B \end{matrix}$, $P_A \in L(\mathcal{G}, A)$, $P_B \in L(\mathcal{G}, B) \Rightarrow P_A \ominus P_B \in L(\mathcal{G}, N)$

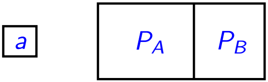
a

$L(\mathcal{G}) = L(\mathcal{G}, S_0)$ – picture language generated by $\mathcal{G}$

Generated picture language

$L(\mathcal{G}, N)$ .. a picture language generated by a nonterminal N

1. $N \to a \Rightarrow a \in L(\mathcal{G}, N)$
2. $N \to A\,B$, $P_A \in L(\mathcal{G}, A)$, $P_B \in L(\mathcal{G}, B) \Rightarrow P_A \oplus P_B \in L(\mathcal{G}, N)$
3. $N \to \begin{matrix} A \\ B \end{matrix}$, $P_A \in L(\mathcal{G}, A)$, $P_B \in L(\mathcal{G}, B) \Rightarrow P_A \ominus P_B \in L(\mathcal{G}, N)$

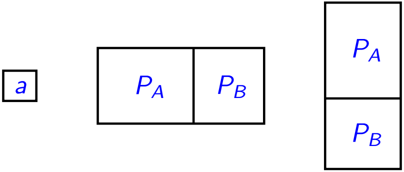


$L(\mathcal{G}) = L(\mathcal{G}, S_0)$ – picture language generated by $\mathcal{G}$

Generated picture language

$L(\mathcal{G}, N)$ .. a picture language generated by a nonterminal N

1. $N \to a \Rightarrow a \in L(\mathcal{G}, N)$
2. $N \to A\,B$, $P_A \in L(\mathcal{G}, A)$, $P_B \in L(\mathcal{G}, B) \Rightarrow P_A \obar P_B \in L(\mathcal{G}, N)$
3. $N \to \begin{matrix} A \\ B \end{matrix}$, $P_A \in L(\mathcal{G}, A)$, $P_B \in L(\mathcal{G}, B) \Rightarrow P_A \ominus P_B \in L(\mathcal{G}, N)$

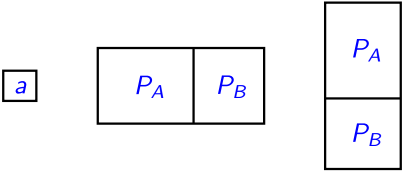


$L(\mathcal{G}) = L(\mathcal{G}, S_0)$ – picture language generated by $\mathcal{G}$

Generated picture language

$L(\mathcal{G}, N)$ .. a picture language generated by a nonterminal N

1. $N \rightarrow a \Rightarrow a \in L(\mathcal{G}, N)$
2. $N \rightarrow A\,B$, $P_A \in L(\mathcal{G}, A)$, $P_B \in L(\mathcal{G}, B) \Rightarrow P_A \obar P_B \in L(\mathcal{G}, N)$
3. $N \rightarrow \begin{matrix} A \\ B \end{matrix}$, $P_A \in L(\mathcal{G}, A)$, $P_B \in L(\mathcal{G}, B) \Rightarrow P_A \ominus P_B \in L(\mathcal{G}, N)$



$L(\mathcal{G}) = L(\mathcal{G}, S_0)$ – picture language generated by $\mathcal{G}$

Square pictures ($n \times n$)

$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$

$V_N = \{R, C, U, S\}$, $V_T = \{a\}$

- one-row pictures: $R \to a\,, \qquad R \to R\,R$
- one-column pictures: $C \to a\,, \qquad C \to \begin{matrix} C \\ C \end{matrix}$
- square 1×1: $S \to a$
- squares $n \times n, n \geq 2$: $U \to S\,C\,, \qquad S \to \begin{matrix} U \\ R \end{matrix}$

Square pictures ($n \times n$)

$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$

$V_N = \{R, C, U, S\}$, $V_T = \{a\}$

- one-row pictures: $R \to a$, $R \to R\,R$
- one-column pictures: $C \to a$, $C \to \begin{matrix} C \\ C \end{matrix}$
- square 1×1: $S \to a$
- squares $n \times n, n \geq 2$: $U \to S\,C$, $S \to \begin{matrix} U \\ R \end{matrix}$

Square pictures ($n \times n$)

$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$

$V_N = \{R, C, U, S\}$, $V_T = \{a\}$

- one-row pictures: $R \to a\,, \qquad R \to R\,R$
- one-column pictures: $C \to a\,, \qquad C \to \begin{matrix} C \\ C \end{matrix}$
- square 1×1: $S \to a$
- squares $n \times n$, $n \geq 2$: $U \to S\,C\,, \qquad S \to \begin{matrix} U \\ R \end{matrix}$

Square pictures ($n \times n$)

$$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$$

$$V_N = \{R, C, U, S\},\ V_T = \{a\}$$

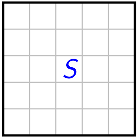
- one-row pictures: $R \to a\,,\quad R \to R\,R$
- one-column pictures: $C \to a\,,\quad C \to \begin{matrix} C \\ C \end{matrix}$
- square 1×1: $S \to a$
- squares $n \times n, n \geq 2$: $U \to S\,C\,,\quad S \to \begin{matrix} U \\ R \end{matrix}$

Square pictures ($n \times n$)

$$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$$

$$V_N = \{R, C, U, S\},\ V_T = \{a\}$$

- one-row pictures: $R \to a\,,\qquad R \to R\,R$
- one-column pictures: $C \to a\,,\qquad C \to \begin{matrix} C \\ C \end{matrix}$
- square 1×1: $S \to a$
- squares $n \times n, n \geq 2$: $U \to S\,C\,,\qquad S \to \begin{matrix} U \\ R \end{matrix}$

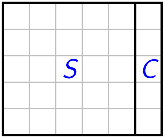


Square pictures ($n \times n$)

$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$

$V_N = \{R, C, U, S\}$, $V_T = \{a\}$

- one-row pictures: $R \to a$, $R \to R\,R$
- one-column pictures: $C \to a$, $C \to \begin{matrix} C \\ C \end{matrix}$
- square 1×1: $S \to a$
- squares $n \times n, n \geq 2$: $U \to S\,C$, $S \to \begin{matrix} U \\ R \end{matrix}$

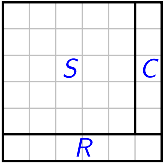


Square pictures ($n \times n$)

$$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$$

$$V_N = \{R, C, U, S\},\ V_T = \{a\}$$

- one-row pictures: $R \to a\,,\qquad R \to R\,R$
- one-column pictures: $C \to a\,,\qquad C \to \begin{matrix} C \\ C \end{matrix}$
- square 1×1: $S \to a$
- squares $n \times n, n \geq 2$: $U \to S\,C\,,\qquad S \to \begin{matrix} U \\ R \end{matrix}$



Exponentially sized pictures ($n \times 2^n$)

$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$

$V_N = \{A, R, D, S\}$, $V_T = \{a\}$

- one-row pictures: $R \to a\,,$ $R \to R\,R$
- picture 1×2: $A \to a\,,$ $S \to A\,A$
- pictures $n \times 2^n, n \geq 2$: $D \to S\,S\,,$ $S \to \begin{matrix} D \\ R \end{matrix}$

Exponentially sized pictures ($n \times 2^n$)

$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$

$V_N = \{A, R, D, S\}$, $V_T = \{a\}$

- one-row pictures: $R \to a\,,$ $R \to R\,R$
- picture 1×2: $A \to a\,,$ $S \to A\,A$
- pictures $n \times 2^n, n \geq 2$: $D \to S\,S\,,$ $S \to \begin{array}{c} D \\ R \end{array}$

Exponentially sized pictures ($n \times 2^n$)

$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$

$V_N = \{A, R, D, S\}$, $V_T = \{a\}$

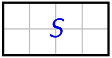
- one-row pictures: $R \to a$, $R \to R\,R$
- picture 1×2: $A \to a$, $S \to A\,A$
- pictures $n \times 2^n, n \geq 2$: $D \to S\,S$, $S \to \begin{matrix} D \\ R \end{matrix}$

Exponentially sized pictures ($n \times 2^n$)

$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$

$V_N = \{A, R, D, S\}$, $V_T = \{a\}$

- one-row pictures: $R \to a\,,\quad R \to R\,R$
- picture 1×2: $A \to a\,,\quad S \to A\,A$
- pictures $n \times 2^n, n \geq 2$: $D \to S\,S\,,\quad S \to \begin{matrix} D \\ R \end{matrix}$

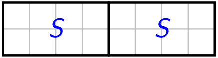


Exponentially sized pictures ($n \times 2^n$)

$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$

$V_N = \{A, R, D, S\}$, $V_T = \{a\}$

- one-row pictures: $R \to a\,,\quad R \to R\,R$
- picture 1×2: $A \to a\,,\quad S \to A\,A$
- pictures $n \times 2^n, n \geq 2$: $D \to S\,S\,,\quad S \to \begin{matrix} D \\ R \end{matrix}$

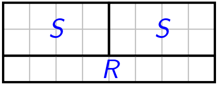


Exponentially sized pictures ($n \times 2^n$)

$\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$

$V_N = \{A, R, D, S\}$, $V_T = \{a\}$

- one-row pictures: $R \to a\,,\quad R \to R\,R$
- picture 1×2: $A \to a\,,\quad S \to A\,A$
- pictures $n \times 2^n, n \geq 2$: $D \to S\,S\,,\quad S \to \begin{matrix} D \\ R \end{matrix}$



Emptiness problem

- Emptiness problem is undecidable for
  - 2D Kolam grammar with extended productions
    (known result [DCFS 2014])
  - 3D Kolam grammar
    (new result)
- 2D Kolam grammar is powerful enough to
  - generate finite languages consisting of very large pictures, and
  - represent certain exponential Diophantine equations.

Emptiness problem

- Emptiness problem is undecidable for
  - 2D Kolam grammar with extended productions (known result [DCFS 2014])
  - 3D Kolam grammar (new result)
- 2D Kolam grammar is powerful enough to
  - generate finite languages consisting of very large pictures, and
  - represent certain exponential Diophantine equations.

2D context-free grammar (2CFG)

$$P_{i,j} \in L(\mathcal{G}, A_{i,j})$$

$$N \rightarrow \begin{matrix} A_{1,1} & \dots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{m,1} & \dots & A_{m,n} \end{matrix}$$

$$\begin{array}{|c|c|c|} \hline P_{1,1} & P_{1,2} & P_{1,3} \\ \hline P_{2,1} & P_{2,2} & P_{2,3} \\ \hline P_{3,1} & P_{3,2} & P_{3,3} \\ \hline \end{array} \in L(\mathcal{G}, N)$$

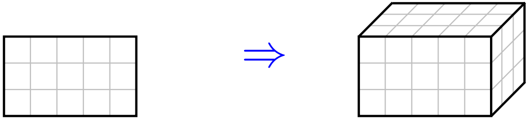
Theorem (Průša, DCFS 2014)

The emptiness problem is not decidable for 2CFG.

- Halting problem reduction, simulation of (1D) Turing machine.

3D languages

The basic entity is a *cuboid*:

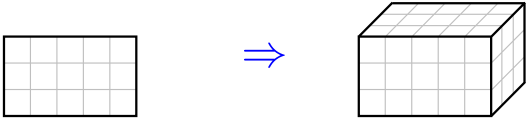


3D Kolam grammar (3KG):

- productions support three types of concatenation:

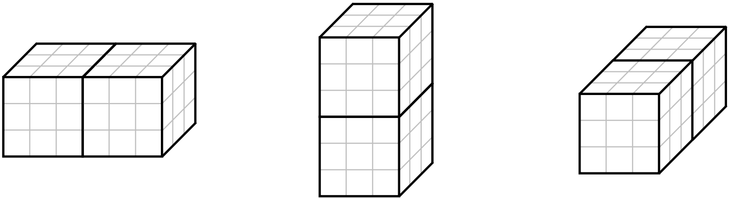
3D languages

The basic entity is a *cuboid*:



3D Kolam grammar (3KG):

- productions support three types of concatenation:



Emptiness problem in 3D

Theorem

The emptiness problem is not decidable for 3KG.

Post Correspondence Problem (PCP)
Input: $\alpha_1, \dots, \alpha_n$ and $\beta_1, \dots, \beta_n$, 2 lists of strings over $\{0, 1\}$
Is there a sequence of indices $(i_k)_{1 \le k \le K}$, $K \ge 1$, $1 \le i_k \le n$, such that $\alpha_{i_1} \dots \alpha_{i_K} = \beta_{i_1} \dots \beta_{i_K}$?

Example

α_1	α_2	α_3
0	01	110

β_1	β_2	β_3
100	00	11

$$\alpha_3\alpha_2\alpha_3\alpha_1 = 110011100 = \beta_3\beta_2\beta_3\beta_1$$

Encoding

$\mathcal{I} = (i_k)_{1 \leq k \leq K}, \quad 1 \leq i_k \leq n$

$\mathrm{code}(i_k)$ .. i_k written in binary as a string of length $\lceil \log_2(n+1) \rceil$

$\mathrm{code}(\mathcal{I}) = 1\,\mathrm{code}(i_1)\,\mathrm{code}(i_2)\ldots\mathrm{code}(i_K)$

$\mathrm{str}_\alpha(\mathcal{I}) = 1\alpha_{i_1}\alpha_{i_2}\ldots\alpha_{i_K}, \qquad \mathrm{str}_\beta(\mathcal{I}) = 1\beta_{i_1}\beta_{i_2}\ldots\beta_{i_K}$

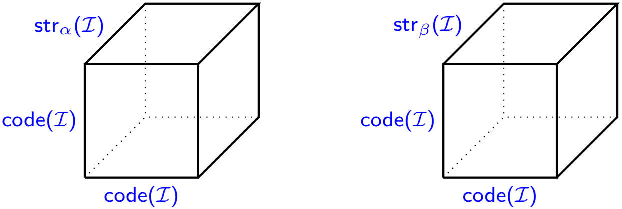
Encoding

$\mathcal{I} = (i_k)_{1 \leq k \leq K}, \quad 1 \leq i_k \leq n$

$\mathrm{code}(i_k)$ .. i_k written in binary as a string of length $\lceil \log_2(n+1) \rceil$

$\mathrm{code}(\mathcal{I}) = 1\,\mathrm{code}(i_1)\,\mathrm{code}(i_2) \ldots \mathrm{code}(i_K)$

$\mathrm{str}_\alpha(\mathcal{I}) = 1\alpha_{i_1}\alpha_{i_2}\ldots\alpha_{i_K}, \qquad \mathrm{str}_\beta(\mathcal{I}) = 1\beta_{i_1}\beta_{i_2}\ldots\beta_{i_K}$



Generating representatives of $(\mathcal{I}, \mathrm{str}_\alpha(\mathcal{I}))$ inductively

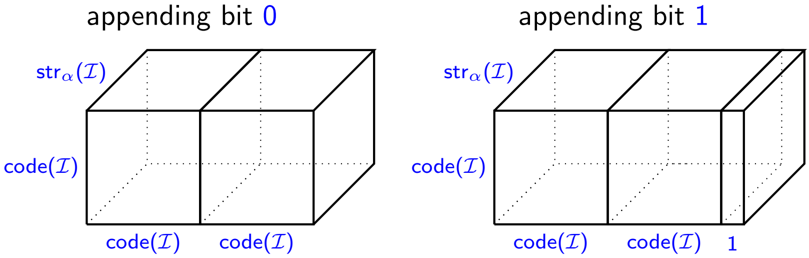
$$\mathcal{I}' = (i_1, \dots, i_K, i_{K+1})$$
$$\mathrm{code}(\mathcal{I}') = 1\,\mathrm{code}(i_1)\,\mathrm{code}(i_2)\dots\mathrm{code}(i_K)\mathrm{code}(i_{K+1})$$
$$\mathrm{str}_\alpha(\mathcal{I}') = 1\alpha_{i_1}\alpha_{i_2}\dots\alpha_{i_K}\alpha_{i_{K+1}}$$

Generating representatives of $(\mathcal{I}, \mathrm{str}_\alpha(\mathcal{I}))$ inductively

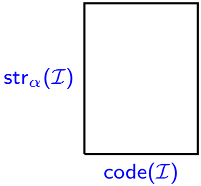
$\mathcal{I}' = (i_1, \ldots, i_K, i_{K+1})$

$\mathrm{code}(\mathcal{I}') = 1\,\mathrm{code}(i_1)\,\mathrm{code}(i_2)\ldots\mathrm{code}(i_K)\mathrm{code}(i_{K+1})$

$\mathrm{str}_\alpha(\mathcal{I}') = 1\alpha_{i_1}\alpha_{i_2}\ldots\alpha_{i_K}\alpha_{i_{K+1}}$

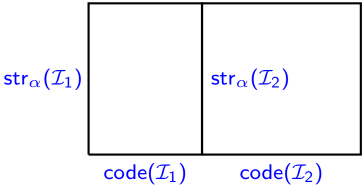


The proof does not work in 2D



Doubling picture width cannot be controlled by a production $D \rightarrow C\ C$.

Consider there are two index sequences $\mathcal{I}_1$, $\mathcal{I}_2$ such that $\mathrm{str}_\alpha(\mathcal{I}_1) = \mathrm{str}_\alpha(\mathcal{I}_2)$.



2D Kolam grammar

Membership problem is decidable.

Parsing algorithm [Schlesinger 2002; Reghizzi and Pradella 2007]

- dynamic programming, generalization of the Cocke-Younger-Kasami algorithm
- time complexity $\mathcal{O}(m^2 n^2 (m + n))$ for a picture $m \times n$

Problem

Is there a recursive upper bound on the maximal size of the smallest picture generated by a 2KG with n nonterminals?

2D Kolam grammar

Membership problem is decidable.

Parsing algorithm [Schlesinger 2002; Reghizzi and Pradella 2007]

- dynamic programming, generalization of the Cocke-Younger-Kasami algorithm
- time complexity $\mathcal{O}(m^2 n^2 (m + n))$ for a picture $m \times n$

Problem

Is there a recursive upper bound on the maximal size of the smallest picture generated by a 2KG *with* n *nonterminals?*

1D context-free grammars

Theorem (pumping lemma)

Let $\mathcal{G} = (V_N, V_T, \mathcal{P}, S)$ be a context-free grammar in the Chomsky normal form. Let $p = 2^{|V_N|-1}$ and $q = 2^{|V_N|}$. If $z \in L(\mathcal{G})$ and $|z| > p$, then z can be written as $z = uvwxy$, where $|vwx| \leq q$ and $|vx| > 0$, such that for each $i \in \mathbb{N}$, $uv^iwx^iy \in L(\mathcal{G})$.

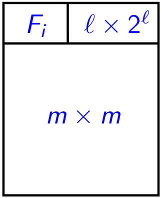
$2^{|V_N|-1}$ is an upper bound on length of the maximal shortest word generated by a (1D) context-free grammar in the Chomsky normal form $\Rightarrow$ the emptiness problem is decidable.

2D Kolam grammar

There is a 2KG with $2n + 6$ nonterminals generating exactly one picture whose width and height is at least $2 \uparrow\uparrow (n-1)$.

$F_1 \to a$

$F_{i+1} \to$



Knuth's up-arrow notation:

$$b \uparrow\uparrow k = \underbrace{b^{b^{\cdot^{\cdot^{\cdot^{b}}}}}}_{k}$$

$i = 1, \ldots, n-1$

Diophantine equations

Definition (Diophantine equation)

A Diophantine equation is an equation of the form $P(x_1, \ldots, x_k) = 0$ where $P(x_1, \ldots, x_k)$ is a polynomial with integer coefficients.

Example

$2x^3y^2z + 3y^7z^3 - 5z = 0$

Problem (Hilbert's 10th problem)

Is there an algorithm deciding whether a given Diophantine equation has a solution among the positive integers?

Solvability of a Diophantine equation is not decidable. [Matiyasevich, Robinson, Davis, Putnam 1970]

Diophantine equations

Definition (Diophantine equation)

A Diophantine equation is an equation of the form $P(x_1, \ldots, x_k) = 0$ where $P(x_1, \ldots, x_k)$ is a polynomial with integer coefficients.

Example

$2x^3y^2z + 3y^7z^3 - 5z = 0$

Problem (Hilbert's 10th problem)

Is there an algorithm deciding whether a given Diophantine equation has a solution among the positive integers?

Solvability of a Diophantine equation is not decidable.
[Matiyasevich, Robinson, Davis, Putnam 1970]

Diophantine equations

Definition (Diophantine equation)

A Diophantine equation is an equation of the form $P(x_1, \ldots, x_k) = 0$ where $P(x_1, \ldots, x_k)$ is a polynomial with integer coefficients.

Example

$2x^3y^2z + 3y^7z^3 - 5z = 0$

Problem (Hilbert's 10th problem)

Is there an algorithm deciding whether a given Diophantine equation has a solution among the positive integers?

Solvability of a Diophantine equation is not decidable.
[Matiyasevich, Robinson, Davis, Putnam 1970]

Exponential Diophantine equations

Example (Exponential Diophantine equation)

$2x^2 3^y z - 2^z = 0$

A subclass with all summands containing one variable:

$$f(x_1, \dots, x_m, y_1, \dots, y_n) = \sum_{i=1}^{m} a_i x_i^{d_i} + \sum_{j=1}^{n} b_j 2^{y_j} = 0$$

where $a_i, b_j \in \mathbb{Z}$ and $d_i \in \mathbb{N}$ for all $i = 1, \dots, m$, $j = 1, \dots, n$.

Still quite powerful:

Example

$x^3 + y^3 = 4981z^3$

Components of the smallest integral solution have over 10^7 digits.

Exponential Diophantine equations

Example (Exponential Diophantine equation)

$2x^2 3^y z - 2^z = 0$

A subclass with all summands containing one variable:

$$f(x_1, \ldots, x_m, y_1, \ldots, y_n) = \sum_{i=1}^{m} a_i x_i^{d_i} + \sum_{j=1}^{n} b_j 2^{y_j} = 0$$

where $a_i, b_j \in \mathbb{Z}$ and $d_i \in \mathbb{N}$ for all $i = 1, \ldots, m$, $j = 1, \ldots, n$.

Still quite powerful:

Example

$x^3 + y^3 = 4981z^3$

Components of the smallest integral solution have over 10^7 digits.

Exponential Diophantine equations

Example (Exponential Diophantine equation)

$2x^2 3^y z - 2^z = 0$

A subclass with all summands containing one variable:

$$f(x_1, \ldots, x_m, y_1, \ldots, y_n) = \sum_{i=1}^{m} a_i x_i^{d_i} + \sum_{j=1}^{n} b_j 2^{y_j} = 0$$

where $a_i, b_j \in \mathbb{Z}$ and $d_i \in \mathbb{N}$ for all $i = 1, \ldots, m$, $j = 1, \ldots, n$.

Still quite powerful:

Example

$x^3 + y^3 = 4981z^3$

Components of the smallest integral solution have over 10^7 digits.

Exponential Diophantine equations

For each f, there is a 2KG $\mathcal{G}_f$ such that

- $L(\mathcal{G}_f) \neq \emptyset$ iff $f(x_1, \ldots, x_m, y_1, \ldots, y_n) = 0$ has a solution
- sizes of generated pictures are proportional to solutions
- the number of nonterminals of $\mathcal{G}_f$ is linear in a binary representation of the equation

Representable functions

Definition

A function $f : \mathbb{N}^+ \to \mathbb{N}^+$ is *representable* by 2KG if $L(f) = \{ a^{n,f(n)} \mid n \in \mathbb{N}^+ \}$ is generated by a 2KG.

Examples of representable functions:

n, 2^n, n^d $(d \in \mathbb{N})$

Lemma

If f, g are representable by 2KG *and $c \in \mathbb{N}^+$, then cf and $f + g$ are also representable by* 2KG.

Proof: The class of picture languages generated by a 2KG is closed under concatenation.

Representable functions

Definition

A function $f : \mathbb{N}^+ \rightarrow \mathbb{N}^+$ is *representable* by 2KG if $L(f) = \{ a^{n,f(n)} \mid n \in \mathbb{N}^+ \}$ is generated by a 2KG.

Examples of representable functions:

n, 2^n, n^d ($d \in \mathbb{N}$)

Lemma

If f, g are representable by 2KG and $c \in \mathbb{N}^+$, then cf and $f + g$ are also representable by 2KG.

Proof: The class of picture languages generated by a 2KG is closed under concatenation.

Representable functions

Definition

A function $f : \mathbb{N}^+ \to \mathbb{N}^+$ is *representable* by 2KG if $L(f) = \{ a^{n,f(n)} \mid n \in \mathbb{N}^+ \}$ is generated by a 2KG.

Examples of representable functions:

n, 2^n, n^d $(d \in \mathbb{N})$

Lemma

If f, g are representable by 2KG and $c \in \mathbb{N}^+$, then cf and $f + g$ are also representable by 2KG.

Proof: The class of picture languages generated by a 2KG is closed under concatenation.

Representable functions

Definition

A function $f : \mathbb{N}^+ \to \mathbb{N}^+$ is *representable* by 2KG if $L(f) = \{ a^{n,f(n)} \mid n \in \mathbb{N}^+ \}$ is generated by a 2KG.

Examples of representable functions:

n, 2^n, n^d ($d \in \mathbb{N}$)

Lemma

If f, g are representable by 2KG *and $c \in \mathbb{N}^+$, then cf and $f + g$ are also representable by* 2KG.

Proof: The class of picture languages generated by a 2KG is closed under concatenation.

Conclusion

- 2D and 3D context-free grammars over a unary alphabet are quite powerful.
- Decidability/undecidability of the emptiness problem is an open problem for 2KG.
- Proving decidability would reveal a significant difference between 2D and 3D Kolam grammars.

Thank you!