

# Languages, automata and computation II

## Tutorial 3 – Applications of well-quasi orders

Winter semester 2024/2025

### Regular languages

**Exercise 1.** Consider the set of finite words well-quasi ordered by the subword relation  $(\Sigma^*, \sqsubseteq)$ . Show that *every* downward closed language over  $\Sigma$  is regular.

### Lossy rewrite systems

**Exercise 2.** A *rewrite system* over a finite alphabet  $\Sigma$  is a finite set of pairs  $u \rightarrow v$  with  $u, v \in \Sigma^*$ . Consider the least reflexive and transitive congruence  $\rightarrow^*$  on  $\Sigma^*$  containing  $\rightarrow$ . A rewrite system is *lossy* if it contains transitions  $a \rightarrow \varepsilon$  for every  $a \in \Sigma$ . Show that the relation  $\rightarrow^*$  is decidable when  $\rightarrow$  is lossy.

### Vector addition systems

**Exercise 3.** Let  $\mathcal{V}$  be a  $d$ -dimensional VASS and consider a target configuration  $t \in P \times \mathbb{N}^d$ , where  $P$  is the set of states. Show that one can compute the set of *all configurations*  $s$  which can cover  $t$ .

**Exercise 4.** Let  $\mathcal{V}$  be a  $d$ -dimensional VAS and consider a source configuration  $s \in \mathbb{N}^d$ . Show that one can decide whether there are only *finitely many* configurations reachable from  $s$ .

**Exercise 5.** Let  $\mathcal{V}$  be a  $d$ -dimensional VAS and consider a source configuration  $s \in \mathbb{N}^d$ . Show that for any coordinate  $k \in \{1, \dots, d\}$  it is decidable whether there exists a number  $n \in \mathbb{N}$  such that every configuration  $t$  reachable from  $s$  has the  $k$ th coordinate bounded by  $n$ .

**Exercise 6.** Show that a VASS of dimension  $d$  can be simulated by a VAS (without states) of dimension  $d + 3$ .

### Vector addition systems over $\mathbb{Z}$

This part is about VASSes but unrelated with well quasi orders. We show that reachability in VASSes is considerably simpler if we relax the requirement that counters cannot become negative.

**Exercise 7.** Let a  $\mathbb{Z}$ -VASS of dimension  $d \in \mathbb{N}$  be a pair  $(Q, T)$  where  $Q$  is a finite set of states and  $T \subseteq Q \times \mathbb{Z}^d \times Q$  be a finite set of transitions. The semantics is as in VASS, except that now configurations are in  $Q \times \mathbb{Z}^d$  (instead of the more restrictive  $Q \times \mathbb{N}^d$ ). Show that reachability is decidable for  $\mathbb{Z}$ -VASSes.

## Strassen's matrix multiplication algorithm

This section is unrelated with well-quasi orders. We begin with a simpler problem.

**Problem 1.** Consider three complex numbers  $a = a_1 + a_2i, b = b_1 + b_2i, c = c_1 + c_2i \in \mathbb{C}$ . The naive multiplication algorithm would compute the product  $c = a \cdot b$  as

$$\begin{aligned} c_1 &= a_1 \cdot b_1 - a_2 \cdot b_2, \\ c_2 &= a_1 \cdot b_2 + a_2 \cdot b_1, \end{aligned}$$

which uses four multiplications in  $\mathbb{R}$ . Find a more efficient algorithm that uses only three multiplications and any number of additions.

Consider  $2 \times 2$  matrices of rational numbers

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}, C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} \in \mathbb{Q}^{2 \times 2}.$$

The naive multiplication algorithm would compute the product  $C = A \cdot B$  as

$$\begin{aligned} c_{11} &= a_{11} \cdot b_{11} + a_{12} \cdot b_{21}, \\ c_{12} &= a_{11} \cdot b_{12} + a_{12} \cdot b_{22}, \\ c_{21} &= a_{21} \cdot b_{11} + a_{22} \cdot b_{21}, \\ c_{22} &= a_{21} \cdot b_{12} + a_{22} \cdot b_{22}, \end{aligned}$$

which uses 8 multiplications (we do not care about additions). When applied recursively, this yields the following formula for the number of ring multiplications used in order to compute the product of two  $n \times n$  matrices  $A, B$

$$M(n) \leq O(n^2) + 8 \cdot M(n/2).$$

From this we obtain a complexity upper bound  $M(n) \leq O(n^3)$  for naive matrix multiplication. Strassen's algorithm uses only 7 multiplications by computing the following products:

$$\begin{aligned} m_1 &= (a_{11} + a_{22}) \cdot (b_{11} + b_{22}), \\ m_2 &= (a_{21} + a_{22}) \cdot b_{11}, \\ m_3 &= a_{11} \cdot (b_{12} - b_{22}), \\ m_4 &= a_{22} \cdot (b_{21} - b_{11}), \\ m_5 &= (a_{11} + a_{12}) \cdot b_{22}, \\ m_6 &= (a_{21} - a_{11}) \cdot (b_{11} + b_{12}), \\ m_7 &= (a_{12} - a_{22}) \cdot (b_{21} + b_{22}). \end{aligned}$$

The number of ring multiplications for the improved algorithm satisfies

$$M(n) \leq O(n^2) + 7 \cdot M(n/2).$$

and thus  $M(n) \leq O(n^{\log_2 7})$ , where  $\log_2 7 \approx 2.81$ .