Exercises to the lecture
Semantics
Sheet 8

Prof. Dr. Roland Meyer
Jan Grünke
Delivery until 03.07.2025 at 16:45

Exercise 8.1 (r-Bad Sequences)
We consider in this exercise a generalisation of good sequences: a sequence $x_0, x_1, ...$ over a qo $(X, \leq)$ is r-*good* if we can extract an increasing subsequence of length $r+1$, i.e. if there exist $r+1$ indices $i_0 < ... < i_r$ s.t. $x_{i_0} \leq ... \leq x_{i_r}$. A sequence is r-*bad* if it is not r-good. Thus good and bad stand for 1-good and 1-bad respectively. By characterizations of wqos, r-bad sequences over a wqo are always finite. Let $(X, \leq, |\cdot|)$ be a nwqo. Then, similar to bad (g, n)-controlled sequences we define the maximal length of an r-bad (g, n)-controlled sequence over X to be $L_{r,g,X}(n)$.
Our purpose is to show that questions about the length of r-bad sequences reduce to questions about bad sequences: $L_{r,g,X}(n) = L_{g,X\times\Gamma_r}(n)$.

1. Show that such a maximal (g, n)-controlled r-bad sequence is $(r-1)$-good.
2. Show $L_{r,g,X}(n) \leq L_{g,X\times\Gamma_r}(n)$ by transforming a r-bad sequence over the nwqo X into a bad sequence over the nwqo $X \times \Gamma_r$.

   Hint: Use the following definition of goodness in your transformation. Given a sequence $x_0, x_1, ..., x_l$ over X, an index i is p-*good* if it starts an increasing subsequence of length $p+1$, i.e. if there exist indices $i = i_0 < ... < i_p$ s.t. $x_{i_0} \leq ... \leq x_{i_p}$. The *goodness* of an index i is the largest p s.t. i is p-good.
3. Show the converse, i.e. that $L_{r,g,X}(n) \geq L_{g,X\times\Gamma_r}(n)$ by transforming a r-good sequence over X into a good sequence over $X \times \Gamma_r$.

Exercise 8.2 (Complexity of Karp and Miller Trees)
Recall that a (d-dimensional) vector addition system (VAS) is a finite subset $T \subseteq \mathbb{Z}^d$ together with an initial marking $s_0 \in \mathbb{N}^d$. A transition $t \in T$ is *enabled* in a marking $s \in \mathbb{N}^d$ if $s' := s + t \in \mathbb{N}^d$, we write $s \xrightarrow{t} s'$. In this excercise we want to determine the time complexity of the algorithm that constructs the Karp Miller tree (KM tree) for a given VAS. Recall that the KM tree is a tree that has nodes labeled by generlized markings $(\mathbb{N} \cup \{\omega\})^d$ and edges labeled by transitions. The algorithm constructs the tree as follows:

1. The root is lableded by s_0.
2. Repeat: For each leaf s_l with path $s_0, s_1, ..., s_l$ and for each transition $s_l \xrightarrow{t} s_c$ so that $s_c \not\leq s_i$ for all $i = 0, ..., l$ we add a new t-labeled edge to a child with label

$$s'_c := \begin{cases} s_c + \omega \cdot (s_c - s_i) & \text{for smallest } i \text{ with } s_c > s_i, \\ s_c & \text{if no such } i \text{ exists.} \end{cases}$$

Determine the running time of the KM tree construction for a fixed dimension d.
Hint: Use the result from the previous excercise.