Exercise 1

In this problem we consider bad sequences $x_0, \ldots, x_{\ell-1}$ over $(\mathbb{N}^k, \leq_x)$ that obey the following restriction: $x_i \leadsto x_{i+1}$ for all $i < \ell - 1$. The (nondeterministic) transition relation $\leadsto$ is defined by

$$x = (x[1], \ldots, x[k]) \leadsto y = (y[1], \ldots, y[k]) \overset{\text{def}}{=} \forall i \in \{1, \ldots, k\} : y[i] = 0 \lor y[i] = x[i] + 1$$

In other words, a $k$-tuple can be modified by incrementing some components and resetting the others.

For any $n \in \mathbb{N}$, we define

$$L_k(n) = \sup \{ \ell : \text{there exists a bad sequence } x_0 \leadsto x_1 \leadsto \cdots \leadsto x_{\ell-1} \text{ over } \mathbb{N}^k \text{ with } |x_0| \leq n \}.$$ 

We further let $L_k(\omega) = \sup \{ L_k(n) : n \in \mathbb{N} \}$. Recall that, in these definitions, the sup of an unbounded set of natural numbers will be written $\omega$ or $\infty$. Also recall that the norm $|x|$ of a $k$-tuple $x$ is $\max(x_1, \ldots, x_k)$, also called the “infinity norm”.

1. Explain quickly why $L_k(n)$ is finite when $n$ is.

$L_k(n)$ is finite as we saw when defining length functions in class (cf. section 2.1 of the lecture notes). The key argument is that there is only a finite number of possibilities for $x_0$ and every $x_i$ has finitely many possible successors (in fact, at most $2^k$) so the forest of bad sequences $x_0 \leadsto x_1 \leadsto \cdots \leadsto x_{\ell-1}$ has finitely many roots and is finitely branching. Since every branch is a bad sequence, the branches are finite. Thus the forest is finite by König’s Lemma and it has a branch of maximal length.

Accepted answer is: $L_k$ is bounded by the length function $L_{\text{Succ},\mathbb{N}^k}$ from the lecture notes since $x \leadsto y$ implies $|y| \leq \text{Succ}(|x|)$.

2. What is $L_1(n)$ for $n \in \mathbb{N}$?

In dimension 1, $L_1(n) = 2$ if $n > 0$, the longest bad sequence being $\langle n \rangle \leadsto \langle 0 \rangle$. One also has $L_1(0) = 1$.

3. Show that $L_2(\omega) = 5$.

A longest bad sequence is

$$\langle a, b \rangle \leadsto \langle a + 1, 0 \rangle \leadsto \langle 0, 1 \rangle \leadsto \langle 1, 0 \rangle \leadsto \langle 0, 0 \rangle,$$

assuming $a, b > 1$. 


4. Show that $L_k(\omega)$ is finite for any $k$.

We proceed by induction on $k$. The result is trivial for $k = 0$ (and has been proved for $k = 1, 2$ in the previous questions but we won’t assume that).

Let $k > 0$ and consider a bad sequence $x_0 \leadsto x_1 \cdots x_{\ell - 1}$ over $\mathbb{N}^k$. Wlog we can assume that the bad sequence ends with $x_{\ell} = 0 = (0, \ldots, 0)$. Therefore for any $j \in \{1, \ldots, k\}$ there is some $x_i$ in the sequence with $x_i[j] = 0$. Let us write $p_j$ for the smallest index such that $x_{p_j}[j] = 0$ (i.e., the first occurrence of 0 in the $j$-th coordinate) and $p$ for $\max(p_1, \ldots, p_k)$, the last time some coordinate is 0 for the first time. Wlog we may assume that $p = p_k$.

We now write $x_i$ under the form $x_i = (y_i, a_i)$ with $y_i = (x_i[1], \ldots, x_i[k - 1]) \in \mathbb{N}^{k - 1}$ and $a_i = x_i[k]$. For $i < p$, the $k$-coordinate is increasing, so $y_0 \leadsto y_1 \leadsto \cdots \leadsto y_{p - 1}$ is a bad sequence over $\mathbb{N}^{k - 1}$. Hence $p \leq L_{k - 1}(\omega)$. Furthermore, the suffix sequence $x_p \leadsto x_{p + 1} \leadsto \cdots \leadsto x_{\ell - 1}$ is bad too, and we further know that $|x_p| \leq p - 1$ since each coordinate has a 0 in $x_0, \ldots, x_{p - 1}$, and can at most increase by 1 in any step since that. Hence $\ell \leq L_{k - 1}(\omega) + L_k(L_{k - 1}(\omega) - 1)$. Since $L_{k - 1}(\omega)$ is finite by induction hypothesis, we deduce that $\ell$ is bounded by some fixed finite number, hence $L_k(\omega)$ is finite.

Exercise 2

This exercise considers numerical functions computed by vector addition systems with states (VASS’s).

We recall that a ($d$-dimensional) VASS is some $S = (d, Q, R)$ where $d \in \mathbb{N}$ is a dimension, $Q$ is finite set of locations (also called control states, or just states), and $R \subseteq Q \times \mathbb{Z}^d \times Q$ is a finite set of transition rules. In graphical representation, $S$ is depicted as a directed graph with node set $Q$ and each rule $r \in R$ of the form $r = (q, v, p)$ is depicted as an edge labeled with $v$ and going from $q$ to $p$.

Fig. 1 has an example, with $d = 2$, $Q = \{p, q\}$, and $R = \{(p, (0, 0), q), (q, (-2, 1), q)\}$.

![Figure 1: A 2-dim VASS.](image)

A configuration of a VASS $S = (d, Q, R)$ is a pair $(p, x)$ where $p \in Q$ is a location and $x \in \mathbb{N}^d$ is a tuple of counter values. We write $\text{Conf}(S)$ for the set $Q \times \mathbb{N}^d$ of all configurations of $S$.

The operational semantics of $S$ is given by its steps: $S$ has a step $(p, x) \rightarrow_S (q, y)$ iff $R$ contains a rule $\delta$ of the form $r = (p, v, q)$ such that $y = x + v$. Usually we just write $(p, x) \rightarrow (q, y)$ when $S$ is clear from the context. We may write $(p, x) \xrightarrow{\delta} (q, v)$ to name the rule that justifies the step, and $(p, q) \xrightarrow{\delta} (q, v)$ to mean that there is a sequence of steps (possibly 0) going from $(p, x)$ to $(q, v)$. Note that $u, v, \ldots$ denote tuples over $\mathbb{Z}$ while $x, y, \ldots$ denote tuples over $\mathbb{N}$.

1. For the example from Fig. 1, describe the set $\text{Post}^*(p, x_1, x_2)$ of all configurations reachable from some initial configuration $(p, x_1, x_2)$ in terms of the parameters $x_1, x_2 \in \mathbb{N}$.

$$\text{Post}^*(p, x_1, x_2) = \{(p, x_1, x_2)\} \cup \{(q, y, y') \mid x_2 \leq y' \leq x_2 + \left\lfloor \frac{x_1}{2} \right\rfloor \land y' = x_1 - 2y' + 2x_2\}.$$ 

For a dimension $d = a + b + c$, we often see a tuple $x \in \mathbb{N}^d$ as being made of three parts, $x_1 \in \mathbb{N}^a$, $x_2 \in \mathbb{N}^b$ and $x_3 \in \mathbb{N}^c$, denoted $x = x_1, x_2, x_3$. We further write $0_a$ for the $a$-dim zero tuple $(0, 0, \ldots, 0) \in \mathbb{N}^a$. Throughout this exercise, the symbol “$\leq$” denotes the standard product ordering.
between tuples of integers (including 1-dim tuples, i.e., integers). It is only used between tuples of same dimension.

Let \( f : \mathbb{N}^a \to \mathbb{N}^b \) be a (total) function from \( a \)-tuples of natural numbers to \( b \)-tuples and \( S = (d, Q, R, q_0, q_\ell) \) be a \( d \)-dimensional VASS with distinguished locations \( q_0, q_\ell \in Q \) (\( q_0 \) is a designated starting location, \( q_\ell \) is a terminating location).

We say that “\( S \) computes \( f \)” \( \iff \) \( d \geq a + b \) (we write \( c \) for \( d - a - b \)) and the following two properties hold:

\[
\begin{aligned}
(\forall x, x' \in \mathbb{N}^a, y \in \mathbb{N}^b, z \in \mathbb{N}^c) & \quad (q_0, x, 0_b, 0_c) \xrightarrow{*} S (q_\ell, x', y, z) \implies y \leq f(x), \quad \text{(Safety)} \\
(\forall x \in \mathbb{N}^a, y \in \mathbb{N}^b) & \quad y \leq f(x) \implies (q_0, x, 0_b, 0_c) \xrightarrow{*} S (q_\ell, 0_a, y, 0_c) \quad \text{(Completeness)}
\end{aligned}
\]

Note that Safety states an universal property of configurations in \( Post^*(q_0, x, 0_b, 0_c) \) while Completeness states an existential property over the same set.

2. Give a VASS that computes the function \( f : \mathbb{N}^3 \to \mathbb{N} \) given by \( f(x_1, x_2, x_3) = 3x_1 + 2x_2 + x_3 \). Justify the correctness of your solution.

We put \( d = 4 \) (so \( c = 0 \)), \( Q = \{q\} \), let \( q_f = q_0 \defeq q \) and define the rule set

\[
R = \{ (q, (-1, 0, 0, 3), q), (q, (0, 1, 0, 2), q), (q, (0, 0, -1, 1), q), (q, (0, 0, 0, -1), q) \}.
\]

For correctness: Steps that use the first three rules keep \( 3x_1 + 2x_2 + x_3 - y \) invariant, so Safety for \( f \) is respected. The fourth rule decreases \( y \), also respecting Safety. We now check Completeness: by using the first three rules, one can set \( x_1, x_2, x_3 \) to 0, and \( y \) to \( f(x_1, x_2, x_3) \). Then one uses the fourth rule to reach any desired \( y \leq f(x) \).

A function \( f : \mathbb{N}^a \to \mathbb{N}^b \) is called monotonic if \( x \leq x' \) implies \( f(x) \leq f(x') \). It is called VASS-computable, if there exists a VASS \( S \) that computes it.

3. Prove that if \( f \) is VASS-computable then \( f \) is monotonic.

The operational semantics of VASSes imply the following property:

\[
(q, v) \to (q', v') \implies (q, v + \Delta) \to (q', v' + \Delta) \quad \text{for all} \quad \Delta \in \mathbb{N}^d, \quad \text{(Additivity)}
\]

and this immediately extends to sequences of steps.

Now assume that \( f \) is computed by some VASS \( S \) and consider any \( x \leq x' \). Write \( x' = x + \Delta \). With Completeness we know that \( (q_0, x, 0, 0) \xrightarrow{*} S (q_\ell, 0, y, 0) \) for \( y = f(x) \). Then \( (q_0, x', 0, 0) \xrightarrow{*} (q_\ell, \Delta, y, 0) \) using Additivity. Safety implies \( y \leq f(x') \), i.e., \( f(x) \leq f(x') \).

4. Prove that if \( f : \mathbb{N}^a \to \mathbb{N}^b \) and \( g : \mathbb{N}^b \to \mathbb{N}^c \) are VASS-computable then \( g \circ f \) is. (NB: this requires to prove safety and completeness for some object that has to be shown to exist).

Assume \( S_1 = (d_1, Q_1, R_1, q_{0,1}, q_{1,1}) \) computes \( f : \mathbb{N}^{a_1} \to \mathbb{N}^{b_1} \) and \( S_2 = (d_2, Q_2, R_2, q_{0,2}, q_{1,2}) \) computes \( g : \mathbb{N}^{a_2} \to \mathbb{N}^{b_2} \) with \( a_2 = b_1 \). Write \( c_1 = d_1 - a_1 - b_1 \) and \( c_2 = d_2 - a_2 - b_2 \) for the number of auxiliary counters in \( S_1 \) and \( S_2 \). The VASS for \( g \circ f \) is obtained from \( S_1 \) and \( S_2 \) by the following operations:

1. extend \( S_1 \) and \( S_2 \) to a new dimension \( d \defeq d_1 + d_2 - b_1 \), padding the rules with \( d - d_1 \) (i.e., \( b_2 + c_2 \)) extra zeroes for \( S_1 \), and with \( d - d_2 \) (i.e., \( a_1 + c_1 \)) extra zeroes for \( S_2 \);
2. reorder the counters inside (the rules of) $S_1$ and $S_2$ so that natural configurations now read $(q_1, x_1, 0_{b_2}, y_1, z_1, 0_{c_2})$ for $S_1$ and $(q_2, 0_{b_1}, y_2, x_2, 0_{c_1}, z_2)$ for $S_2$; 
3. build the disjoint union of the modified $S_1$ and $S_2$, add a rule $r_{1,2} = (q_f, 0_d, q_0, 0)$ that connect the two subsets of control locations, fix start state $q_{0,1}$ and final state $q_{f,2}$.

Call $S$ the resulting VASS. We claim it computes $g \circ f$. For Safety we have to consider an arbitrary run from $(q_{0,1}, x_1, 0_{b_2}, 0_{b_1} + c_1 + c_2)$ to $q_{f,2}$. The topology of the new $S$ implies that the run must use $r_{1,2}$ exactly once. It has the form:

$$
(q_{0,1}, x_1, 0_{b_2}, 0_{b_1}, o_i, 0_{c_2}) \xrightarrow{R_1} (q_{f,1}, x'_1, y_2, y_1, z_1, z_2) \xrightarrow{r_{1,2}} (q_{0,2}, x'_1, y_2, y_1, z_1, z_2) \xrightarrow{R_2} (q_{f,2}, x''_1, y'_2, y'_1, z'_1, z'_2).
$$

Since $R_1$ and $R_2$ do not modify the new counters that have been added, we know that $y_2 = 0_{b_2}$ and $z_2 = 0_{c_2}$ from $R_1$, and $x''_1 = x'_1$ and $z'_1 = z_1$ from $R_2$. Furthermore, and since the first half can be seen as a $S_1$ run, we deduce from $S_1$'s safety that $y_1 \leq f(x_1)$. Since the second half can be seen as a $S_2$ run, we deduce from $S_2$'s safety that $y'_2 \leq g(y_1)$. **Note that $S_2$'s safety can only be invoked since $y_2$ and $z_2$ are zeroes!** Finally, since $g$ is monotonic (question 3) we deduce $y'_1 \leq (g \circ f)(x_1)$, which is Safety for $S$.

Completeness is easier and omitted.

5. Show that all constants functions $f : \mathbb{N}^a \rightarrow \mathbb{N}^b$ are VASS-computable. Show that addition $\langle (x_1, x_2) \rangle \in \mathbb{N}^2 \mapsto x_1 + x_2 \in \mathbb{N}$, projections $\langle (x_1, \ldots, x_a) \rangle \in \mathbb{N}^a \mapsto x_i \in \mathbb{N}$, and copies $(x_1, \ldots, x_a) \in \mathbb{N}^a \mapsto (x_1, x_1, \ldots, x_a, x_a) \in \mathbb{N}^{2a}$ are V-computable.

Easy and omitted.

6. Show that multiplication $\langle (x_1, x_2) \rangle \in \mathbb{N}^2 \mapsto x_1 \times x_2 \in \mathbb{N}$ is VASS-computable. Show that all multivariate polynomials with coefficients in $\mathbb{N}$ (e.g., $(x_1, x_2, x_3) \mapsto 2x_1^2x_3^5 + 3x_2x_3 + 5$) are VASS-computable.

For multiplication we use $d = 4$ (i.e., one extra counter) and two states $q, p$. Rules are

$$
r_1 : q \xrightarrow{0, -1, 0, 0} q, \quad r_2 : q \xrightarrow{-1, 0, 0, 0} p, \quad r_3 : p \xrightarrow{0, -1, 1, 1} p, \quad r_4 : p \xrightarrow{0} q, \quad r_5 : q \xrightarrow{0, 1, 0, -1} q,
$$

and we let $q_0 = q_f = q$. A generic run can be:

$$
q \quad p \quad p \quad p \quad q \quad q \quad q \\
0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \\
x_1 \quad x_1 \quad x_1 \quad x_1 \quad x_1 \quad x_1 \quad x_1 \\
1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \\
x_2 \quad x_2 \quad x_2 \quad x_2 \quad x_2 \quad x_2 \quad x_2 \\
0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \\
x_1x_2 \quad x_1x_2 \quad x_1x_2 \quad 2x_2 \quad 2x_2 \quad 2x_2 \quad x_1x_2 \\
0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0
$$

With multiplication, multivariate polynomials can be obtained by composing (question 4) copies, constants, multiplications and additions from the previous question.

7. Give a VASS that computes $x \mapsto \lfloor \frac{x}{2} \rfloor$ and one that computes $x \mapsto 2^x$. 
