Fundamenta Informaticae 140 (2015) 61–87
DOI 10.3233/FI-2015-1245
IOS Press

Catching a Structural Bug with a Flower

Florent Avellaneda*
LAAS - CNRS, 7 avenue du colonel Roche, F-31077 Toulouse, France
florent.avellaneda@lif.univ-mrs.fr; florent.avellaneda@laas.fr

Rémi Morin
Aix Marseille Université, CNRS, LIF UMR 7279, F-13288, Marseille, France
remi.morin@lif.univ-mrs.fr

Abstract. Checking the structural boundedness and the structural termination of vector addition systems with states boils down to detecting pathological cycles. As opposed to their non-structural variants which require exponential space, these properties need polynomial time only. The algorithm searches for a counter-example in the form of a multiset of arcs computed by means of linear programming. Yet the minimal length of a pathological cycle can be exponential in the size of the system which makes it difficult to visualize and to analyze the detected bug in details.

We propose to describe pathological cycles in the form of particular cycles called flowers. The latter are made of petals which are iterated circuits connected by simple paths that form a calyx. We present an algorithm that builds in polynomial time a flower from the multiset of arcs that represents a pathological cycle. Interestingly the number of petals within a flower is at most equal to the dimension of vectors which helps to describe in a concise way the underlying bug and to analyse it.

Keywords: Vector addition system with states, structural properties, counter-example, dynamic graph, zero-cycle.

1. Introduction

Consider a set of reactions that takes place among a collection of particles such that each reaction consumes a multiset of available particles and produces a linear combination of other particle types. This

*Address for correspondence: LAAS - CNRS, 7 avenue du colonel Roche, F-31077 Toulouse, France

kind of framework can be formalized by a vector addition system [13] or, equivalently, a (pure) Petri net. In this case, particles are called *tokens* and particle types are called *places*. Consider in addition a control state that determines which reactions can occur, and such that the occurrence of a reaction leads to a possibly distinct control state. Then the model becomes formally a vector addition system with states (a VASS), a notion introduced in [11]. In this paper we are interested in two *structural properties* for VASS, that is, properties that do not depend on a particular initial distribution of particles among places. In this way, we consider the initial marking as a parameter of the system.

The first problem we consider asks whether the number of particles in the system remains bounded for each initial configuration. In other words only finitely many distinct configurations can be reached. Since particles often represent the consumption of resources, such as messages in channels, this first problem asks whether there exists some amount of resources sufficient to cope with all configurations reachable from any fixed *finite* set of potential initial configurations. A second basic issue is to check that a given system terminates, i.e. whether there is no infinite execution, for each initial configuration. Thus we aim at checking that a system eventually deadlocks. Although one usually tries to avoid deadlocks in concurrent systems, termination remains in some cases a basic problem in formal verification: In particular non-termination can result from livelocks in concurrent programs when components fail to achieve their tasks.

Structural properties regard the initial configuration as a parameter of the system. However, they are also considered in practice for the analysis of systems with a fixed initial configuration. The reason is that these stronger properties turn out to be easier to check than their non-structural variants [17, 23]. More precisely checking boundedness for Petri nets provided with an initial configuration requires exponential space [8, 19] whereas structural boundedness is polynomial [8, 21]. Furthermore a similar gap exists between termination [4, Th. 3] and structural termination [24].

In order to describe a distributed system, it is often convenient to use a *vector* of control states whose components are the local states of each process. Then particles represent messages within channels. This model is called a *parallel-composition-VASS* [17, 18]. It is close to the notion of a communicating finite-state machine [3] but with non-FIFO message exchanges. Similarly to the simulation of a VASS by a Petri net, a parallel-composition-VASS can be simulated by a Petri net with additional places whose marking represent the current vector of control states. Interestingly structural properties are not preserved by this simulation. Besides checking the structural boundedness of a parallel-composition-VASS is NP-complete [17] whereas this problem can be solved in polynomial time in the particular case of Petri nets [8, 21].

We observe first in Section 2 that verifying the structural boundedness or the structural termination of a given VASS boils down to checking the costs of cycles within the system viewed as a weighted directed graph: A cycle is pathological for structural boundedness (resp. structural termination) if its arc weights sum to a positive (resp. non-negative) vector. Consequently these two problems are very close to the detection of a zero-cycle in dynamic graphs [12], which asks if there exists a cycle with a zero cost. In [15] Kosaraju and Sullivan showed how to decide the existence of such a cycle in polynomial time. Besides this problem was proved later to be equivalent to the general linear programming problem [6]. The idea is twofold. First cycles are identified with particular multisets of arcs called *circulations*. Second circulations with zero cost appear as solutions to some linear program. This technique adapts easily to the detection of pathological cycles for structural boundedness or structural termination. The resulting algorithm returns in polynomial time a circulation that represents a pathological cycle if such a cycle exists.

When the model of a system does not satisfy a given property, formal verification tools usually provide users with a counter-example execution in the form of a sequence of atomic steps that describes an unexpected behaviour. In this paper, we tackle the problem of providing a useful description of a pathological cycle for a structural property. The point is that the number of times an arc occurs in a pathological cycle can be exponential in the size of the given VASS, even though the time needed to compute the corresponding multiset of arcs is only polynomial. Consequently listing the sequence of arcs occurring along such a cycle is prohibitive in general. A first approach consists in providing a partial description of the detected pathological cycle (a sort of slice of the counter-example) as the set of all arcs occurring in this cycle —or simply the set of places interacting in the reactions performed by these arcs. However, this information may not be sufficient to understand fully the detected bug. In particular, it does not determine the minimal configuration required to execute a detected pathological cycle.

In the particular case of a VASS with a single state —that is to say: a pure Petri net— a multiset of arcs can be regarded as a multiset of cycles with a common starting state. Moreover, due to Carathéodory's theorem [22, Cor. 7.7i], we need at most p distinct arcs to describe a structural bug if the given VASS has p places. More precisely, given n integral p-dimensional vectors $x_1, ..., x_n \in \mathbb{Z}^p$ and a linear combination $\lambda_1 \cdot x_1 + ... + \lambda_n \cdot x_n = z$ with $\lambda_1, ..., \lambda_n \in \mathbb{N}$, if $n > p$ then we can compute in polynomial time other coefficients $\lambda'_1, ..., \lambda'_n \in \mathbb{N}$ such that $\lambda'_1 \cdot x_1 + ... + \lambda'_n \cdot x_n = m \cdot z$ for some $m \in \mathbb{N} \setminus \{0\}$ and moreover $\lambda'_i \neq 0$ for at most $n - 1$ values of $i \in [1..n]$ (Cor. 4.7). Thus each pathological cycle can be represented by p elementary cycles of length 1 and with a common starting state. In this work, we want to extend this property to any VASS: *We aim at decomposing a given pathological cycle in the form of a particular multiset of connected cycles. Moreover each component cycle should be easy to depict and the number of distinct cycles in this multiset should be at most equal to the number of places in the given VASS.*

We introduce in Section 3 the notion of a flower. Roughly speaking, a flower consists of iterated circuits connected by simple paths. Our main result asserts that we can compute in polynomial time such a structure that corresponds to a given multiset of arcs which represents a pathological cycle. Moreover the number of distinct circuits we need is at most equal to the number of places. Thus we propose to describe a structural bug to the user in the form of a small number of circuits and a connecting cycle together with the number of times each of these cycles occurs. Note that this information allows us to compute the minimal configuration required to execute this new pathological cycle. This information is useful to the user when structural properties are checked instead of their non-structural variants, if the abstraction process yields a false counter-example. Then the analysis of the detected pathological cycle can lead to a refined model with a larger set of initialized places.

The construction of a flower we present in this work relies on another class of particular cycles, called *wings*, introduced in [2]. Intuitively, a wing consists of a circuit provided with two simple paths back and forth from a fixed starting state to a state along the circuit. Additionally, the *valuation* of a wing determines the number of iterations of its cyclic component. An intermediate result is established in Section 4. We show how to compute in polynomial time a multiset of wings with a common starting state that corresponds to a pathological cycle given as a multiset of arcs. Moreover the common starting state of these wings can be chosen arbitrarily and, here again, the number of distinct wings we need is at most equal to the number of places. We show in Section 5 how to build a flower from such a multiset of wings.

Example 1.1. Along this paper, we shall use as a running example the 2-dimensional VASS depicted in Figure 1. This VASS has three states q_0, q_1, and q_2 and five weighted arcs a_1, a_2, a_3, l_1, and l_2. The cost of the cycle $\gamma = a_1.(l_1)^5.a_2.(l_2)^3.a_3$ is $(1, 4)^\top$. So this cycle is pathological for both structural termination and structural boundedness: This VASS is not structurally bounded and it does not terminate structurally. The cycle γ is actually a flower with l_1 and l_2 as iterated circuits while a_2 and $a_3.a_1$ are two connecting paths. Of course, it is easy to guess a pathological flower in this simple case. This running example is only used to illustrate the various concepts along the paper and the algorithm adopted to compute a flower. Observe here that the two cycles $\omega_1 = a_1.(l_1)^{10}.a_2.a_3$ and $\omega_2 = a_1.a_2.(l_2)^6.a_3$ are actually two wings starting from q_0 with valuations 10 and 6 respectively. The cost of the cycle $\omega_1.\omega_2$ is $(2,8)^\top$ so the multiset of wings $\omega_1 + \omega_2$ is pathological, too.

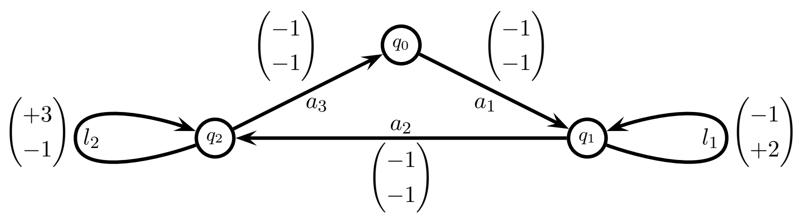

Figure 1. A vector addition system with states

2. Background

Let p be a fixed non-zero natural number. A vector addition system with states is a directed graph whose arcs are labeled by vectors from $\mathbb{Z}^p$.

Definition 2.1. [11] A *vector addition system with states* (for short, a VASS) is a pair $\mathcal{S} = (Q, A)$ where Q is a finite set of states, and $A \subseteq Q \times \mathbb{Z}^p \times Q$ is a finite set of arcs labeled by vectors from $\mathbb{Z}^p$.

Throughout the paper we let $\mathcal{S} = (Q, A)$ be a VASS. We let $|Q|$ and $|A|$ denote the cardinalities of Q and A respectively. The source and the target of a labeled arc $a \in A$ are denoted by $\mathrm{dom}(a)$ and $\mathrm{cod}(a)$ respectively. We let $\mathrm{cost}(a) \in \mathbb{Z}^p$ denote the column vector labeling each arc $a \in A$. The size of a VASS $\mathcal{S} = (Q, A)$ is $\mathrm{size}(\mathcal{S}) = |A| \times (2 \times \lceil \log_2(|Q| + 1) \rceil + p \times (1 + \lceil \log_2(1 + v_{\max}) \rceil))$ where $v_{\max}$ is the maximal absolute value of coefficients of vectors labeling arcs in $\mathcal{S}$.

2.1. Basics

Let $\mathcal{S} = (Q, A)$ be a VASS. A configuration is a pair $(q, r) \in Q \times \mathbb{N}^p$ consisting of a control state q and a multiset of available particles r. A labeled arc $a \in A$ is enabled at the configuration (q, r) and leads to the configuration (q', r') if $\mathrm{dom}(a) = q$, $\mathrm{cod}(a) = q'$, and $r + \mathrm{cost}(a) = r'$. An execution of $\mathcal{S}$ from an initial configuration $(q_{\text{in}}, r_{\text{in}})$ is a sequence of labeled arcs $a_1...a_n \in A^\star$ such that there are configurations (q_0, r_0), ..., (q_n, r_n) for which $(q_0, r_0) = (q_{\text{in}}, r_{\text{in}})$ and for each $i \in [1..n]$, the labeled arc a_i is enabled at (q_{i-1}, r_{i-1}) and leads to (q_i, r_i). Then the configuration (q_n, r_n) is reachable from $(q_{\text{in}}, r_{\text{in}})$.

A path is a sequence of arcs $\gamma = a_1...a_n \in A^\star$ such that we have $\mathrm{dom}(a_{i+1}) = \mathrm{cod}(a_i)$ for each $i \in [1..n-1]$. A path $\gamma = a_1...a_n \in A^\star$ is closed if $n \geqslant 1$ and $\mathrm{dom}(a_1) = \mathrm{cod}(a_n)$. A closed path is called a *cycle*. A path $\gamma = a_1...a_n \in A^\star$ is *simple* if $\mathrm{dom}(a_i) \neq \mathrm{dom}(a_j)$ for all distinct i, j. A *circuit* is a simple and closed path. The cost of a path $\gamma = a_1...a_n$ is the vector $\mathrm{cost}(\gamma) = \sum_{i=1}^{i=n} \mathrm{cost}(a_i)$. Further the cost of a multiset of arcs $x \in \mathbb{N}^A$ is $\mathrm{cost}(x) = \sum_{a \in A} x[a] \cdot \mathrm{cost}(a)$ and the cost of a finite multiset of paths $\mathcal{W}$ is $\mathrm{cost}(\mathcal{W}) = \sum_{\gamma \in A^\star} \mathcal{W}[\gamma] \cdot \mathrm{cost}(\gamma)$. Let v and v' be two integral vectors with n coordinates: $v = (v[1], ..., v[n])$ and $v' = (v'[1], ..., v'[n])$. We put as usual $v \geqslant v'$ if $v[i] \geqslant v'[i]$ for each i; $v > v'$ if $v[i] > v'[i]$ for each i; and $v \gneqq v'$ if $v \geqslant v'$ and $v \neq v'$.

2.2. Structural properties and characterizations with cycles

A VASS provided with an initial configuration $(q_{\text{in}}, r_{\text{in}})$ *terminates*, if there is a natural number k such that the length of each execution of $\mathcal{S}$ from $(q_{\text{in}}, r_{\text{in}})$ is at most k. In other words, an initialized VASS terminates if and only if it has no infinite execution. In this paper, we are interested in *structural properties*, that is, properties that do not depend on an initial configuration.

Definition 2.2. A VASS $\mathcal{S}$ *terminates structurally* if it terminates for all initial configurations.

The structural termination problem for vector addition systems asks whether a given VASS has no infinite execution for all initial configurations. We observe first that this question boils down to searching for particular cycles in $\mathcal{S}$. This fact is easily established by Kőnig infinity lemma [14] and Dickson's lemma [13, Lemma 4.1].

Proposition 2.3. A VASS $\mathcal{S}$ terminates structurally if and only if there is no cycle γ with $\mathrm{cost}(\gamma) \geqslant \vec{0}$.

A VASS provided with an initial configuration $(q_{\text{in}}, r_{\text{in}})$ is *bounded* if it admits only finitely many reachable configurations. A VASS $\mathcal{S}$ is *structurally bounded* if it is bounded for all initial configurations. Similarly to Proposition 2.3, checking the structural boundedness of a VASS boils down to detecting a cycle with a non-negative non-zero cost: A VASS $\mathcal{S}$ is structurally bounded if and only if there exists no cycle γ with $\mathrm{cost}(\gamma) \gneqq \vec{0}$. In the sequel of this paper, we focus on structural termination, only. Consequently witnesses of non-satisfiability are formalized by the following notion of a pathological path. However all results presented in this paper adapt immediately to structural boundedness.

Definition 2.4. A cycle γ in a VASS $\mathcal{S}$ is *pathological* if $\mathrm{cost}(\gamma) \geqslant 0$.

Example 2.5. Consider the VASS with a single state and six self-loop arcs labeled respectively by the six following 6-dimensional vectors:

$$t_1 = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}; t_2 = \begin{pmatrix} -1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}; t_3 = \begin{pmatrix} 0 \\ -1 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix}; t_4 = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 1 \\ 0 \\ 0 \end{pmatrix}; t_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -2 \\ 1 \\ 0 \end{pmatrix}; t_6 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -2 \\ 1 \end{pmatrix}$$

It is easy to see that each pathological cycle needs all arcs because of their pairwise dependencies. Moreover a pathological cycle that contains one occurrence of t_6 needs 2 occurrences of t_5, 4 occurrences of t_4 and hence 4 occurrences of t_3, 2 occurrences of t_2 and one occurrence of t_1. Therefore the pathological cycle $\gamma = t_1.(t_2)^2.(t_3)^4.(t_4)^4.(t_5)^2.t_6$ has a minimal length. We can easily generalize this example to a VASS (with a single state) made of $2 \times m$ arcs, whose pathological cycles have a length greater than $2 \times (2^m - 1)$.

This example shows that the minimal length of a pathological cycle can be exponential in the size of the given VASS. For that reason, an abstract view of cycles is necessary to deal with them in polynomial time. Following [6, 15], we shall represent cycles of a VASS $\mathcal{S}$ as particular multisets of arcs called *circulations*.

2.3. Circulations vs. cycles

Let $x \in \mathbb{N}^A$ be a multiset of arcs. We denote by $\|x\| = |\{a \in A \mid x[a] \geqslant 1\}|$ the number of distinct arcs in x and by A_x the *support of* x, that is to say the set of arcs $a \in A$ such that $x[a] \geqslant 1$. Thus $\|x\| = |A_x|$. The *underlying graph* G_x of x is the (undirected) graph $G_x = (Q_x, E_x)$ where the set of vertices $Q_x = \{\mathrm{dom}(a) \mid a \in A_x\} \cup \{\mathrm{cod}(a) \mid a \in A_x\}$ collects the source and the target of all arcs in x and the set of edges $E_x = \{\{\mathrm{dom}(a), \mathrm{cod}(a)\} \mid a \in A_x \text{ and } \mathrm{dom}(a) \neq \mathrm{cod}(a)\}$ keeps track of all connections induced by arcs in x. The size of x is $\mathrm{size}(x) = \sum_{a \in A} \lceil \log_2(1 + x[a]) \rceil$ because the coefficients of x are encoded in binary.

A multiset of arcs $x \in \mathbb{N}^A$ is called *connected* if G_x is a connected graph. Let $x \in \mathbb{N}^A$ and $K_1, ..., K_n \subseteq Q_x$ be the connected components of G_x. For each $1 \leqslant i \leqslant n$ and each $a \in A$, we put $x_i[a] = x[a]$ if $\mathrm{dom}(a) \in K_i$ and $x_i[a] = 0$ otherwise. Then $x = x_1 + ... + x_n$ and the multisets $x_i \in \mathbb{N}^A$ are called the *connected components of* x.

Definition 2.6. A multiset of arcs $x \in \mathbb{N}^A$ is *Eulerian* if $\sum_{\mathrm{dom}(a)=q} x[a] = \sum_{\mathrm{cod}(a)=q} x[a]$ for each state $q \in Q$. A connected and Eulerian multiset of arcs is called a *circulation*.

Thus a multiset of arcs is Eulerian if for each state q the number of arcs incident from q equals the number of arcs incident to q. Note that if x and y are Eulerian, then $x + y$ is Eulerian. If moreover $x \leqslant y$ then $y - x$ is Eulerian, too. The *multiplicity* of a non-zero multiset $x \in \mathbb{N}^A \setminus \{\vec{0}\}$ within a multiset $y \in \mathbb{N}^A$ is the greatest natural number k such that $k \cdot x \leqslant y$.

Each cycle $\gamma = a_1 ... a_n$ of $\mathcal{S}$ is represented by the multiset of arcs $x_\gamma = \sum_{i=1}^{i=n} a_i$, i.e. $x_\gamma[a]$ is the number of occurrences of a in γ. Since γ is a cycle, the multiset of arcs x_γ is non-empty, Eulerian and connected. For instance, continuing Example 1.1, the multiset of arcs $a_1 + a_2 + a_3 + 5 \cdot l_1 + 3 \cdot l_2$ is the circulation corresponding to the cycle $\gamma = a_1.l_1^5.a_2.l_2^3.a_3$. Conversely, each non-empty circulation corresponds to a cycle of $\mathcal{S}$: This is an immediate variant of Euler's theorem [7, Th. 1.8.1].

Proposition 2.7. Let $x \in \mathbb{N}^A$ be a non-empty circulation. Then there exists a cycle γ such that $x_\gamma = x$.

Observe here that the directed graph made of the set of arcs A_x for a circulation x is strongly connected.

In [15], Kosaraju and Sullivan showed how to detect a cycle with a zero cost in polynomial time by a reduction to linear programming. We explain here why this technique can be adapted with no effort to the verification of structural termination. In fact, it is sufficient to replace a vector equality $x = \vec{0}$ by $x \geqslant \vec{0}$ in part of the linear programming problem considered. Given a VASS $\mathcal{S} = (Q, A)$, a first step consists in listing the subset $A' \subseteq A$ of all arcs that appear in a multiset $\mathcal{W}$ of cycles with $\mathrm{cost}(\mathcal{W}) \geqslant \vec{0}$. To do so, for each arc $b \in A$, one considers the following linear programming problem P_b with $|A|$

rational variables $x \in \mathbb{Q}^A$.

$$(P_b) \begin{cases} \sum_{\mathrm{cod}(a)=q} x[a] = \sum_{\mathrm{dom}(a)=q} x[a] \text{ for each } q \in Q \\ x[a] \geqslant 0 \text{ for each } a \in A \\ x[b] \geqslant 1 \\ \sum_{a \in A} x[a] \cdot \mathrm{cost}(a) \geqslant \vec{0} \end{cases}$$

It is clear that b belongs to A' if and only if P_b has a solution. In that case, we denote by x_b a solution to this problem with $x_b \in \mathbb{N}^A$. We need here to derive an integral solution $x' \in \mathbb{N}^A$ to P_b from a rational one $x \in \mathbb{Q}^A$. To do so, one can use Euclid's algorithm to compute the least common multiple m of the denominators of all coefficients of x and put $x' = m \cdot x$. Next, three cases appear:

1. If A' is empty then $\mathcal{S}$ terminates structurally, because no arc appears in a pathological cycle.

2. If A' is non-empty and strongly connected then $\mathcal{S}$ does not terminate structurally because the multiset of arcs $x = \sum_{b \in A'} x_b$ is connected, Eulerian and $\mathrm{cost}(x) \geqslant 0$. Observe here that the support A_x of x is maximal since $A_x = A'$. Thus the detected pathological cycle has a maximal support.

3. Let A_1,..., A_n be the strongly connected components of A', with $n \geqslant 2$. We consider the n VASSs $\mathcal{S}_1$,..., $\mathcal{S}_n$ obtained from $\mathcal{S}$ by a reduction to the subsets of arcs A_1,..., A_n respectively. Then $\mathcal{S}$ admits a pathological cycle if and only if one of the n subsystems $\mathcal{S}_1$,..., $\mathcal{S}_n$ admits a pathological cycle. Consequently it is sufficient to apply recursively the algorithm to each of these systems.

This algorithm yields a polynomial time procedure that checks whether the given VASS $\mathcal{S}$ terminates structurally and returns a pathological circulation x if it does not. Note here that the length $\sum_{a \in A} x[a]$ of a corresponding cycle can be exponential in the size of $\mathcal{S}$ because the coefficients of x are encoded in binary.

We have observed that the support of x is actually maximal. In practice, we can use this procedure iteratively to make sure that the support of x is minimal, that is, there exists no pathological circulation y with $A_y \subsetneq A_x$. However, searching for a pathological circulation with a minimal number of arcs in its support is known to be NP-hard [2].

3. Representation of circulations

In order to help the understanding of a structural bug detected by a verification tool in the form of a circulation, it is useful to represent this counter-example to the user as a pathological cycle. Then the length of such a pathological cycle equals the sum of the circulation coefficients. Consequently it can be exponential in the size of the VASS, as already observed in Example 2.5. Thus, listing the sequence of arcs occurring along a pathological cycle is prohibitive. For that reason, we need to design a *compact representation* of pathological cycles.

3.1. A format to describe pathological cycles

It is clear that any pathological cycle γ can be decomposed iteratively into a multiset $\mathcal{C}$ of circuits with $\mathrm{cost}(\mathcal{C}) = \mathrm{cost}(\gamma)$. Then we can select within $\mathcal{C}$ a set of p circuits (where p stands for the dimension of vectors) and compute a multiset $\mathcal{C}'$ built over these p circuits such that $\mathrm{cost}(\mathcal{C}') = m \cdot \mathrm{cost}(\gamma)$ for some $m \in \mathbb{N} \setminus \{0\}$. This fact relies on a convexity argument known as Caratheodory's theorem, as discussed in the introduction. However, the set of arcs occurring in the selected p circuits in this way need not to be connected, that is, $\mathcal{C}'$ does not represent a pathological cycle. A natural idea is to use an additional connecting cycle (called the *calyx*) on which the selected circuits (called *petals*) would hang. This leads us to the notion of flower.

Definition 3.1. A *flower* is a structure $\mathcal{F}$ that consists of

- a sequence of k connection states $q_0, \ldots, q_{k-1}$ with $k \geqslant 1$,
- a sequence of k circuits $\sigma_0, \ldots, \sigma_{k-1}$, where each circuit σ_i starts from q_i,
- a connecting cycle $\kappa_0 \ldots \kappa_{k-1}$, where κ_i is a simple path from q_i to $q_{i+1 \pmod k}$ that is empty if $q_i = q_{i+1 \pmod k}$,
- together with a sequence $n_0, \ldots, n_k$ of natural numbers, with $n_i \geqslant 1$.

Such a structure represents the cycle $\gamma_{\mathcal{F}} = \sigma_0^{n_0}.\kappa_0.\sigma_1^{n_1} \ldots \sigma_{k-1}^{n_{k-1}}\kappa_{k-1}.(\kappa_0 \ldots \kappa_{k-1})^{n_k-1}$ starting from q_0. Its cost is equal to $\mathrm{cost}(\mathcal{F}) = \mathrm{cost}(\gamma_{\mathcal{F}}) = \sum_{i=0}^{k-1} n_k \cdot \mathrm{cost}(\kappa_i) + n_i \cdot \mathrm{cost}(\sigma_i)$.

The component circuits σ_i of a flower are called the *petals* while the connecting cycle $\kappa_0 \ldots \kappa_{k-1}$ is called the *calyx*. Each petal σ_i is iterated n_i times while the calyx occurs n_k times in $\gamma_{\mathcal{F}}$. We say that the calyx of $\mathcal{F}$ is *iterated* if $n_k \geqslant 2$. A flower $\mathcal{F}$ is said to be pathological if $\mathrm{cost}(\mathcal{F}) \geqslant \vec{0}$. Continuing Example 1.1, the cycle $\gamma = l_1^5.a_2.l_2^3.a_3.a_1$ corresponds to a flower with two petals that are iterated 3 and 5 times respectively. Note that each flower contains at least one petal so that the represented cycle $\gamma_{\mathcal{F}}$ cannot be empty. We require that κ_i is empty if $q_i = q_{i+1 \pmod k}$ because the calyx is essentially meant to connect petals. In particular, a flower with a single petal has an empty calyx: It is simply an iterated circuit. Note here that a cycle can describe several distinct flowers because the order of petals hanging on the same state is meaningless and the calyx can be regarded as a petal itself, if it is an iterated circuit. These two situations will appear in examples below. Yet, the next example shows that we cannot require in general that the calyx be a circuit.

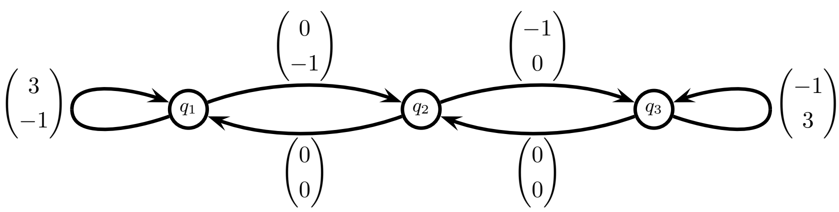

Figure 2. The calyx of a flower cannot be a circuit

Example 3.2. Consider the 2-dimensional VASS with 3 states from Fig. 2. Each non-empty pathological cycle in this VASS makes use of each arc. Consider a pathological flower for this VASS. The two self-loop arcs on q_1 and q_3 must occur as petals. Consequently the calyx goes through these two states and it cannot be a simple cycle.

3.2. Representation of circulations with flowers

In this paper we show that each non-empty pathological circulation can be represented by a flower with at most p petals.

Example 3.3. Let $p = 4$. Consider the VASS with a single state and four self-loop arcs labeled respectively by the four following 4-dimensional vectors:

$$t_1 = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}; t_2 = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}; t_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix}; t_4 = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

The cycle $\gamma = t_1.t_2.t_3.t_4$ satisfies $\mathrm{cost}(\gamma) = \vec{0}$ and corresponds to a flower with four petals: t_1, t_2, t_3, t_4 and an empty calyx. It is clear that any pathological cycle involves all arcs of this VASS because of their pairwise dependencies. Therefore any pathological flower needs precisely these four petals, too, because no self-loop arc can occur within a calyx and petals are required to be circuits.

This example easily adapts to any dimension p. Thus we cannot expect in general to have less than p petals in a representative flower of a pathological circulation. Theorem 3.4 below asserts that it is sufficient to consider flowers with at most p petals to represent any non-empty pathological circulation.

Theorem 3.4. Let H be a non-empty circulation of a VASS $\mathcal{S}$. We can compute in polynomial time a flower $\mathcal{F}$ with at most p petals such that $\mathrm{cost}(\mathcal{F}) = m \cdot \mathrm{cost}(H)$ for some $m \in \mathbb{N} \setminus \{0\}$.

We present in the following sections an algorithm that builds such a flower $\mathcal{F}$ from a given circulation H in time polynomial in the size of the inputs, that is, the size of the VASS $\mathcal{S}$ plus the size of the circulation H. With no surprise, the flower $\mathcal{F}$ is built only from the arcs appearing in the circulation H (because we can assume that all arcs of $\mathcal{S}$ occur in H).

Observe that the resulting flower $\mathcal{F}$ is pathological if and only if the given circulation H is pathological. Thus Theorem 3.4 yields a pathological flower built from any pathological circulation. Since the latter is meant to be produced by our variant of Kosaraju and Sullivan's algorithm, its size is polynomial in the size of the VASS $\mathcal{S}$ —although the length of any corresponding cycle may be exponential in the size of $\mathcal{S}$. Thus, in practice, we obtain a pathological flower in time polynomial in the size of $\mathcal{S}$.

Observe also that we have similarly $\mathrm{cost}(\mathcal{F}) = \vec{0}$ if and only if $\mathrm{cost}(H) = \vec{0}$, and $\mathrm{cost}(\mathcal{F}) \gneq \vec{0}$ if and only if $\mathrm{cost}(H) \gneq \vec{0}$. Consequently Theorem 3.4 applies to the representation of zero-cycles and to counter-examples of structural boundedness, too.

The factor m in the statement of Theorem 3.4 is not a drawback of the representation of circulations by flowers because the actual length of the resulting pathological cycle is not relevant. Moreover this factor is necessary to ensure that the flower has at most p petals, as the next example shows.

Example 3.5. Consider the VASS with a single state and four self-loop arcs labeled respectively by the four following 2-dimensional vectors:

$$t_1 = \begin{pmatrix} -1 \\ 5 \end{pmatrix}; t_2 = \begin{pmatrix} -1 \\ 10 \end{pmatrix}; t_3 = \begin{pmatrix} 1 \\ -6 \end{pmatrix}; t_4 = \begin{pmatrix} 1 \\ -8 \end{pmatrix}$$

Let $H = t_1 + t_2 + t_3 + t_4$. We have $\mathrm{cost}(H) = (0,1)^\top$ so the circulation H is pathological. This circulation can be regarded as a flower with four petals. It is easy to see that one needs only two petals to build an equivalent flower. For instance $H' = t_2 + t_4$ satisfies $\mathrm{cost}(H') = 2 \cdot \mathrm{cost}(H)$. Observe here that the support of H' does not include all arcs from H. Further, there exists no flower $\mathcal{F}''$ with at most two petals such that $\mathrm{cost}(\mathcal{F}'') = \mathrm{cost}(H)$.

The next example shows that Theorem 3.4 does not hold if we require that the iteration of the calyx is forbidden.

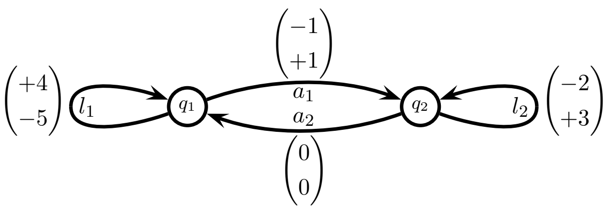

Figure 3. Iterating the calyx is necessary to a flower with at most p petals

Example 3.6. Consider the 2-dimensional VASS with two states q_1 and q_2 depicted in Figure 3 and the non-empty circulation $H_0 = l_1 + l_2 + 2 \cdot (a_1 + a_2)$. We have $\mathrm{cost}(H_0) = \vec{0}$. Let H be any non-empty circulation with $\mathrm{cost}(H) = \vec{0}$. We have $H = x \cdot l_1 + y \cdot l_2 + z \cdot (a_1 + a_2)$ because H is Eulerian. Clearly $x \neq 0$. Furthermore, $y \neq 0$ because the linear system of two equalities $4 \times x - z = 0$ and $-5 \times x + z = 0$ requires that $x = z = 0$. It follows that $z \geqslant 1$ because H is connected. Similarly, we have $z \neq 1$ because the linear system of two equalities $4 \times x - 2 \times y - 1 = 0$ and $-5 \times x + 3 \times y + 1 = 0$ has no integral solution. Consequently $z \geqslant 2$. It follows that any flower $\mathcal{F}$ with $\mathrm{cost}(\mathcal{F}) = \vec{0}$ with at most two petals must admit l_1 and l_2 as petals (since they cannot occur in the calyx) and $a_1.a_2$ as calyx; moreover this calyx must be iterated at least twice.

3.3. An intermediate format: Wings

In this paper, we shall use an alternative structure which represents pathological cycles in the form of a multiset of particular cycles called wings. Roughly speaking, a wing with valuation k is a cycle which consists of k iterations of a circuit plus a path back and forth from one state of the circuit to some fixed starting state. This shared starting state will ensure that a multiset of wings remains connected.

Definition 3.7. Let $q, q' \in Q$ be two states of $\mathcal{S}$. Let γ_0 be a circuit of $\mathcal{S}$ starting from q'. Let γ_1 be a simple path from q to q' and γ_2 be a simple path from q' to q. Let $k \in \mathbb{N} \setminus \{0\}$. Let $\omega = \gamma_1.\gamma_0^k.\gamma_2$ be the cycle which starts from q and which consists of γ_1, followed by k iterations of the cycle γ_0, followed by γ_2. Then ω is called a *wing* of $\mathcal{S}$ with *valuation* k. A wing is said to be *reduced* if q' differs from the domain of each arc of γ_1 and q differs from the domain of each arc of γ_2.

A wing is often represented by a multiset of arcs $W = D + k \cdot C$ where C is the set of arcs occurring in the cycle γ_0 while D is the multiset of arcs occurring in γ_1 and γ_2. Then the multiset W is connected and Eulerian. Note that the connecting cycle $\gamma_1.\gamma_2$ from q need not be simple (nor non-empty). However, each arc occurs *at most twice* in $\gamma_1.\gamma_2$. In this paper, we will only consider reduced wings because they are easier to use in order to build flowers, while the algorithm actually yields reduced wings. The point is that γ_1 and γ_2 are used to connect the iterated circuit γ_0 to the fixed state q, and they can be chosen as shortest paths from q to q' and from q' to q respectively.

Example 3.8. We continue Example 1.1 with $p = 2$. We have observed that the cost of the cycle γ is $\mathrm{cost}(\gamma) = (1, 4)^\top$. Consider the two wings $W_1 = a_1.l_1^{10}.a_2.a_3$ with valuation 10 and $W_2 = a_1.a_2.l_2^6.a_3$ with valuation 6. Noteworthy $2 \cdot \mathrm{cost}(\gamma) = \mathrm{cost}(W_1) + \mathrm{cost}(W_2)$. This equality illustrates precisely how wings can represent a cycle up to a scalar multiplication factor of its cost.

In Section 4 we establish the following intermediate result which asserts that there exists such a representation by wings for any pathological circulation.

Theorem 3.9. Let $\hat{H}$ be a non-empty circulation of a VASS $\mathcal{S}$ and $\hat{q} \in Q_{\hat{H}}$. We can compute in polynomial time a non-empty multiset $\mathcal{W}$ of wings built over at most p distinct reduced wings starting from $\hat{q}$ and such that $\mathrm{cost}(\mathcal{W}) = m \cdot \mathrm{cost}(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$.

The next section is devoted to the proof of Theorem 3.9. We shall present an algorithm that builds a multiset of wings from a circulation in time polynomial in the size of the inputs, that is, the size of the VASS $\mathcal{S}$ plus the size of the circulation $\hat{H}$. Similarly to Theorem 3.4, the given circulation is meant to be produced by our variant of Kosaraju and Sullivan's algorithm: Its size is polynomial in the size of the VASS $\mathcal{S}$. In this case, we compute a pathological multiset of wings in time polynomial in the size of $\mathcal{S}$. The construction of wings from a circulation is largely *non-deterministic*: It relies on a series of arbitrary choices at several stages, without any backtrack. These choices can be solved by, say, an arbitrarily fixed total order over the set of arcs in $\mathcal{S}$. This yields implicitly also a total order over the states of $\mathcal{S}$. Then the constructions of wings and flowers described in the two next sections become deterministic.

As opposed to flowers, wings have many advantages in terms of technical simplicity. Adding a new wing starting from $\hat{q}$ to a set of wings starting from $\hat{q}$ needs no effort, while adding a petal to a flower requires to find a new calyx, as soon as the new petal does not meet the original calyx. Furthermore, removing a wing from a multiset of wings (because it appears to be redundant by application of Carathéodory's theorem) needs also no effort, while removing a petal from a flower can affect the calyx and create a new petal, because the calyx must consist of simple paths, only. Thus, multisets of wings are easier to build and to reduce than flowers.

Still, the format of flowers was suggested in [2] as a simpler structure. The point is that flowers with p petals are conceptually simpler than p iterated wings because each wing $\gamma_1.\gamma_0^k.\gamma_2$ with valuation k that is iterated x times corresponds to a cyclic component γ_0 that is iterated $x \times k$ times and a connecting cycle $\gamma_1.\gamma_2$ that is iterated x times. Intuitively, the calyx in a flower replaces all connecting cycles within a multiset of wings. For that reason, we consider flowers to be a better format.

4. Construction of wings from a circulation

In this section, we fix a VASS $\mathcal{S} = (Q, A)$, a non-empty circulation $\hat{H} \in \mathbb{N}^A$ and a state $\hat{q} \in Q_{\hat{H}}$. We prove Theorem 3.9 in two steps. We show first how to compute in polynomial time a non-empty multiset $\mathcal{W}$ of reduced wings starting from $\hat{q}$ such that $\mathrm{cost}(\mathcal{W}) = m \cdot \mathrm{cost}(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$. Next we explain in Subsection 4.3 how to reduce the number of distinct wings in $\mathcal{W}$ to less than p.

The construction of $\mathcal{W}$ proceeds inductively over the number $\|\hat{H}\|$ of distinct arcs in $\hat{H}$. At each step, a wing $W = D + k \cdot C \leqslant \hat{H}$ with valuation k is added to $\mathcal{W}$ and removed from $\hat{H}$ until $\hat{H}$ is empty. This wing should satisfy the three following properties:

1. Some arc in the cyclic component C has multiplicity k within $\hat{H}$; in this way, at least one arc is removed from the support of $\hat{H}$ at each step: $\|\hat{H} - W\| < \|\hat{H}\|$.

2. The Eulerian multiset of remaining arcs $\hat{H} - W$ is connected; this ensures that we can proceed recursively.

3. The fixed state $\hat{q}$ belongs to the new circulation $\hat{H} - W$, so that all wings share this common starting state —except of course if $\hat{H} - W$ is already empty.

The first idea for the search of such a wing W within $\hat{H}$ is that it is sufficient to find a circuit C satisfying these conditions. This leads us to the following central notion of an *adequate* circuit.

Definition 4.1. Let $H \in \mathbb{N}^A$ be a non-empty circulation and $q_0 \in Q_H$. A circuit C with multiplicity $k \geqslant 1$ in H is *adequate* for H and q_0 if it satisfies the two next conditions:

- the multiset of arcs $H - k \cdot C$ is connected;

- if $H - k \cdot C$ is not empty then $Q_{H-k\cdot C}$ contains q_0.

Example 4.2. Continuing Example 1.1, we consider the circulation $H = a_1 + a_2 + a_3 + 5 \cdot l_1 + 3 \cdot l_2$ for the VASS depicted in Figure 1. Then the two circuits l_1 and l_2 are adequate for H and q_0 whereas the circuit $a_1.a_2.a_3$ is not for two reasons: First, the multiset of arcs $H - a_1 - a_2 - a_3$ is not connected; second, it does not contains q_0.

Note that $\|H - k \cdot C\| < \|H\|$ for any circuit C with multiplicity k in H. The construction of $\mathcal{W}$ relies on two independent algorithms presented in the two next subsections. The first algorithm shows how to find an adequate circuit for any non-empty circulation $H \in \mathbb{N}^A$ and any state $q_0 \in Q_H$. The second one is much easier. It explains how to build the expected multiset $\mathcal{W}$ of wings with the help of adequate circuits as inputs.

4.1. Finding an adequate circuit in a circulation for a fixed state

The search for a circuit C adequate for H and q_0 proceeds non-deterministically and inductively over the number $\|\hat{H}\|$ of arcs in A_H. Each step distinguishes two main cases. The simpler case assumes that all circuits within H contain q_0. Then each circuit is adequate for H and q_0. The reason is that any connected component of the Eulerian multiset $H - k \cdot C$ contains a circuit, and hence contains q_0.

The more interesting case considers that there exists a circuit $C \leqslant H$ that does not contain q_0. Let k be the multiplicity of C within H. Then $q_0 \in Q_{H-k\cdot C}$ because q_0 does not occur in C. Hence $H - k \cdot C$ is not empty. Then the circuit C is adequate if $H - k \cdot C$ is connected. In this case, the search is terminated. Otherwise we consider a connected component H' of $H - k \cdot C$ that does not contain q_0, as illustrated in Fig. 4.1. We will show how to find in H' a circuit C', with multiplicity k' in H', such that

1. at least one arc $a \in A_{C'} \setminus A_C$ satisfies $H'[a] = k'$. Then $H'[a] = H[a]$ and k' is also the multiplicity of a in H; hence $\|H - k' \cdot C'\| < \|H\|$.

2. each connected component of $H' - k' \cdot C'$ contains a state from C. Then $H - k' \cdot C'$ is connected; moreover $q_0 \in Q_{H-k' \cdot C'}$ because q_0 does not occur in H'.

It follows that C' is adequate for H and q_0.

The search for an appropriate circuit C' within H' can be regarded as a generalisation of the search for an adequate circuit C within H where the connectivity of $H - k \cdot C$ is replaced by the connectivity of $H' - k' \cdot C'$ if one incorporates the circuit C. Actually, for simplicity's sake, we will consider at this point a simple path σ made of all but one arcs from C. Intuitively, σ will play the role of C. However we shall also consider a special case where σ is the empty path in order to deal with adequate circuits as a special case.

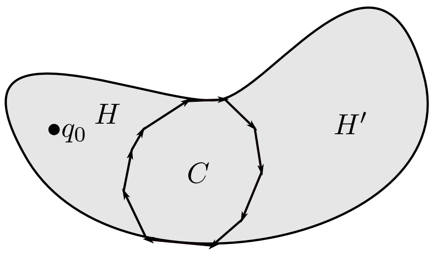

Figure 4. Searching for an adequate circuit

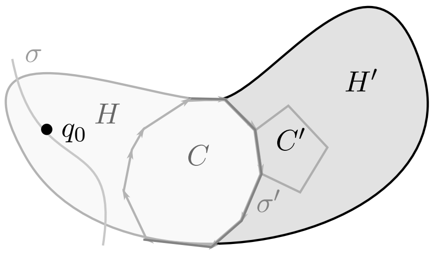

Figure 5. Induction step

Definition 4.3. Let $H \in \mathbb{N}^A$ be a non-empty circulation, $q_0 \in Q_H$, and $\sigma \in A^\star$ be a simple path. A circuit C with multiplicity $k \geqslant 1$ in H is *appropriate* for H and (q_0, σ) if it satisfies the two next conditions:

1. there exists an arc $a \in A_C \setminus A_\sigma$ such that $H[a] = k$;

2. each connected component of $H - k \cdot C$ contains a state from $Q_\sigma \cup \{q_0\}$.

Observe that a circuit C is appropriate for H and (q_0, ϵ) where ϵ denotes the empty path (Def. 4.3) if, and only if, it is adequate for H and q_0 (Def. 4.1). For that reason, the search for an adequate circuit will simply ask for an appropriate circuit w.r.t. the empty path ϵ in Algorithm 2 below. In this way, the role of state q_0 for adequate circuits is extended to a path σ.

We present now in Algorithm 1 a way to compute circuits that are appropriate for H and (q_0, σ), provided that σ is not a circuit, $q_0 \in Q_H$, and $q_0 \in Q_\sigma$ if σ is not empty.

Proposition 4.4. Let $H \in \mathbb{N}^A$ be a circulation. Let $q_0 \in Q_H$ and $\sigma \in A^\star$ be a simple path such that $q_0 \in Q_\sigma$ if σ is not empty. Provided that σ is not a circuit, Algorithm 1 returns a circuit that is appropriate for H and (q_0, σ).

Assume that $H \in \mathbb{N}^A$ is a non-empty circulation and $\sigma = a_1 ... a_n$ is a simple path consisting of arcs from A such that σ is not a circuit. Let $q_0 \in Q_H$ be a state of H such that $q_0 \in Q_\sigma$ if σ is non-empty. Searching for an appropriate circuit C for H and (q_0, σ) is slightly more involved than searching for an adequate one. However, Algorithm 1 proceeds similarly to the above discussion and distinguishes two main cases.

Algorithm 1 AppropriateCircuit(H, q_0, σ)

Require: $H \in \mathbb{N}^A$ is a non-empty circulation.
Require: σ is a (possibly empty) simple path consisting of arcs from A and such that σ is not a circuit.
Require: $q_0 \in Q_H$ and $q_0 \in Q_\sigma$ if the path σ is non-empty.

```
if all circuits C ⩽ H satisfy Q_C ∩ (Q_σ ∪ {q_0}) ≠ ∅ then
  Choose b ∈ A_H \ A_σ arbitrarily
  β ← b                                          # Initially β is a path of length 1
  while β contains no circuit do
    if there exists an arc b′ ∈ A_H \ A_σ with dom(b′) = cod(b) then
      Choose some b′ ∈ A_H \ A_σ with dom(b′) = cod(b)
    else
      Find the arc b′ ∈ A_H ∩ A_σ such that dom(b′) = cod(b)
    end if
    Add the arc b′ to the end of the path β
    b ← b′                                       # β remains the last arc of β
  end while
  return a circuit C within β
else
  Choose a circuit C ⩽ H such that Q_C ∩ (Q_σ ∪ {q_0}) = ∅
  Let k be the multiplicity of C in H
  if each connected component of H − k · C contains a state from Q_σ ∪ {q_0} then
    return C                                     # In particular if H = k · C.
  else
    Choose a connected component H′ of H − k · C with Q_H′ ∩ (Q_σ ∪ {q_0}) = ∅.
    Choose a state q′_0 from Q_H′ ∩ Q_C and an arc a ∈ A_C with H[a] = k.
    Let σ′ be the path made of all arcs from A_C \ {a}
    return AppropriateCircuit(H′, q′_0, σ′)      # Then ‖H′‖ < ‖H‖
  end if
end if
```

We need first to determine whether all circuits in H contain a state from $Q_\sigma \cup \{q_0\}$. To do so, one considers the subset $A' \subseteq A$ consisting of all arcs from A_H whose source and target do not belong to $Q_\sigma \cup \{q_0\}$. Let A'_1,..., A'_n be the strongly connected components of A'. Then there exists a circuit C in H with $Q_C \cap (Q_\sigma \cup \{q_0\}) = \emptyset$ if, and only if, A' contains a self-loop arc or one of the strongly connected components A'_i has two states. Depending on whether this condition is satisfied, we investigate one of the following two cases:

1. We assume first that all circuits in H contain a state from $Q_\sigma \cup \{q_0\}$. Algorithm 1 builds a circuit $C = a_0 a_1 ... a_{n-1}$ in H using preferably arcs that do not appear in σ. Since σ is not a circuit and H is a non-empty circulation, we can choose first an arbitrary arc $b \in A_H \setminus A_\sigma$ and consider the path $\beta = b$. This path is extended iteratively by adding arcs from A_H to the end of β until β contains a circuit C. At each iteration, there are potential candidates to complete β because H is Eulerian. However, we require that arcs from $A_H \setminus A_\sigma$ are preferred to the others in this extension process.

Clearly this loop terminates after at most $|Q_H|$ iterations. At this point, we claim that the circuit C within β is appropriate for H and (q_0, σ).

Proof. Let $k \geqslant 1$ be the multiplicity of C in H. Since H is Eulerian, $H - k \cdot C$ is Eulerian. Let H' be a connected component of $H - k \cdot C$. Since $H - k \cdot C$ is Eulerian, H' is Eulerian. Therefore there is a circuit in H' and hence H' contains a state from $Q_\sigma \cup \{q_0\}$. Thus, all connected components of $H - k \cdot C$ contain a state from $Q_\sigma \cup \{q_0\}$.

Since the simple path σ is not closed, by hypothesis, the circuit $C = a_0 ... a_{n-1}$ within β cannot be made of arcs from σ only. In other words, C contains at least one arc that does not belong to A_σ. Assume that there is an arc $a_i \in A_\sigma \cap A_C$. Due to the priority of arcs adopted, there exists no arc $b \in A_H \setminus A_\sigma$ such that $\mathrm{dom}(b) = \mathrm{dom}(a_i)$. Since σ is a simple path, the arc a_i is the single arc from A_H such that $\mathrm{dom}(a_i) = \mathrm{cod}(a_{i-1 \pmod n})$. Consequently, we have $H[a_{i-1 \pmod n}] \leqslant H[a_i]$ because H is Eulerian. Since C contains at least one arc that does not belong to A_σ, there exists an arc $a \in A_C \setminus A_\sigma$ such that $H[a] \leqslant H[a_i]$. It follows that there exists $a \in A_C \setminus A_\sigma$ such that $H[a]$ is equal to the multiplicity C in H. ■

2. We assume now that there exists a circuit C in H with $Q_C \cap (Q_\sigma \cup \{q_0\}) = \emptyset$. Let $k \geqslant 1$ be the multiplicity of C in H. If each connected component of $H - k \cdot C$ contains at least one state from $Q_\sigma \cup \{q_0\}$ then C is appropriate for H and (q_0, σ). Therefore we assume now that $H - k \cdot C$ is non-empty and admits some connected component H' of $H - k \cdot C$ that contains no state from $Q_\sigma \cup \{q_0\}$. The situation is illustrated in Fig. 4.1. Let $a \in A_C$ be such that $H[a] = k$. Then $H'[a] = 0$ and hence $\|H'\| < \|H\|$. Moreover $Q_{H'} \cap Q_C \neq \emptyset$, otherwise there would be no path from $Q_{H'}$ to Q_C in the circulation H. We fix some state $q_0' \in Q_{H'} \cap Q_C$. We let also σ' denote the simple path made of all arcs from $A_C \setminus \{a\}$. Then σ' contains all arcs from $A_C \cap A_{H'}$. Moreover σ' is not a circuit and $q_0' \in Q_{\sigma'}$ as soon as σ' is not empty. At this point, we claim that any circuit C' appropriate for H' and (q_0', σ') is also appropriate for H and (q_0, σ).

Proof. Let $k' \geqslant 1$ be the multiplicity of C' in H'. Then,

- There exists an arc $a' \in A_{C'} \setminus A_{\sigma'}$ such that $H'[a'] = k'$.
- Each connected component of $H' - k' \cdot C'$ contains a state from $Q_{\sigma'} \cup \{q_0'\}$.

Since σ' contains all arcs from C that occur in H', we have $a' \notin A_C$. Therefore $H[a'] = (H - k \cdot C)[a'] = H'[a']$. It follows that k' is also the multiplicity of C' in H. Since H' contains no state from $Q_\sigma \cup \{q_0\}$, C' contains no state from $Q_\sigma \cup \{q_0\}$ either. Further, we have $a' \in A_{C'} \setminus A_\sigma$. Since $q_0 \in H$ and $q_0 \notin H'$, q_0 appears in $H - k' \cdot C'$. To conclude the proof, we show simply that the Eulerian multiset of arcs $H - k' \cdot C'$ is connected.

Since $H - k \cdot C \geqslant k' \cdot C'$, we have $H - k' \cdot C' \geqslant k \cdot C \geqslant C$. Thus all states of Q_C are strongly connected to each other in $H - k' \cdot C'$. Let $q'' \in Q_{H - k' \cdot C'}$. It remains to show that there exists a path from q'' to a state from C made of arcs from $H - k' \cdot C'$. The claim is trivial if $q'' \in Q_C$. If $q'' \notin Q_C$ then q'' belongs to one of the connected components of $H - k \cdot C$. We distinguish two cases:

- $q'' \in Q_{H'}$. Since $q'' \in Q_{H - k' \cdot C'}$, there exists some arc $a'' \in H - k' \cdot C'$ such that $q'' = \mathrm{dom}(a'')$ or $q'' = \mathrm{cod}(a'')$. Since $q'' \notin Q_C$, we have $a'' \notin C$ and hence $H[a''] = H'[a'']$. Then $H'[a''] - k' \cdot C'[a''] = H[a''] - k' \cdot C'[a''] \geqslant 1$. It follows that $q'' \in Q_{H' - k' \cdot C'}$. Since each connected component of $Q_{H'-k'\cdot C'}$ contains a state from $Q_{\sigma'} \cup \{q'_0\}$ and $Q_{\sigma'} \cup \{q'_0\} \subseteq Q_C$, there exists a path from q'' to C in $H' - k' \cdot C'$ and hence in $H - k' \cdot C'$.

- $q'' \in Q_{H''}$ where H'' is a connected component of $H - k \cdot C$ different from H'. Then $Q_{H''} \cap Q_C \neq \emptyset$ otherwise there would be no path from the set of states $Q_{H''}$ to the set of states Q_C in H. Therefore there exists a path from q'' to C in H'' and hence in $H - k' \cdot C'$.

Thus $H - k' \cdot C'$ is connected and the circuit C' is appropriate for H and (q_0, σ). ∎

4.2. Building a multiset of wings from a pathological circulation

The construction of a representative multiset $\mathcal{W}$ of wings from the multiset $\hat{H}$ of arcs is described in Algorithm 2. Initially $\mathcal{W}$ is empty and we put $H = \hat{H}$. Hence $\mathrm{cost}(\mathcal{W}) + \mathrm{cost}(H) = m \cdot \mathrm{cost}(\hat{H})$ with $m = 1$. This equality will act as a loop invariant of the main loop. First, a circuit C adequate for $\hat{H}$ and $\hat{q}$ is found with the help of Algorithm 1. Recall here that a circuit C is appropriate for H and $(\hat{q}, \epsilon)$ (where ϵ denotes the empty path) if, and only if, it is adequate for H and $\hat{q}$. Let k be the multiplicity of C in H. Then the Eulerian multiset $H - k \cdot C$ is connected and $\hat{q} \in Q_{H-k\cdot C}$ provided that $H - k \cdot C$ is not empty. Moreover $\|H - k \cdot C\| < \|H\|$.

We build from C a wing W starting from $\hat{q}$ with C as its cyclic component. If $\hat{q}$ appears in C then $W = k \cdot C$ is a wing starting from $\hat{q}$. Assume that $\hat{q} \notin Q_C$. Then $\hat{q} \in Q_{H-k\cdot C}$. Since H is connected, there is a state $q \in Q_C \cap Q_{H-k\cdot C}$. Since $H - k \cdot C$ is a circulation, there are a simple path γ_1 from $\hat{q}$ to q and a simple path γ_2 from q to $\hat{q}$ made of arcs from $A_{H-k\cdot C}$. We let D denote the multiset of arcs that corresponds to the cycle $\gamma_1.\gamma_2$. Then the multiset $W = D + k \cdot C$ represents a wing which starts from $\hat{q}$. Moreover $D[a] \leqslant 2$ for each $a \in A$ because γ_1 and γ_2 are simple paths, hence $W \leqslant 3 \cdot H$, because $k \cdot C \leqslant H$. Furthermore, each arc $a \in A_C$ with multiplicity k in H does not occur in $\gamma_1.\gamma_2$, since it does not occur in $H - k \cdot C$. We can require that γ_1 and γ_2 are shortest paths from $\hat{q}$ to q and from q to $\hat{q}$ respectively. Thus W is a *reduced* wing. We distinguish then three cases:

1. If $W = H$ then the wing W is added to $\mathcal{W}$ and removed from H leading to the empty multiset $H' = \vec{0}$.

2. If $W \leqslant H$, $H - W$ is connected and $\hat{q} \in Q_{H-W}$ then the wing W is added to $\mathcal{W}$ and removed from H leading to the new circulation $H' = H - W$ such that $\hat{q} \in Q_{H'}$. Since k is the multiplicity of C in H, we get $\|H'\| < \|H\|$.

3. Otherwise the multiset of wings $\mathcal{W}$ is multiplied by 3. Then $\mathrm{cost}(\mathcal{W}) + \mathrm{cost}(3 \cdot H) = m \cdot \mathrm{cost}(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$. We consider the new wing $W' = D + 3k \cdot C$. We have $W' \leqslant 3 \cdot H$. We claim that $3 \cdot H - W'$ is a circulation that contains $\hat{q}$ if it is not empty; moreover $\|3 \cdot H - W'\| < \|H\|$.

   Proof. Let a be an arc from C such that $H[a] = k$. Then $3 \cdot H[a] - D[a] = 3k$ because a does not occur in $\gamma_1.\gamma_2$. On the other hand, for each arc a' from C with $H[a'] \geqslant k + 1$, we have $3 \cdot H[a'] - D[a'] \geqslant 3k + 1$ because $D[a'] \leqslant 2$. It follows that $3k$ is the multiplicity of C in $3 \cdot H - D$. The wing W' is added to $\mathcal{W}$ and removed from $3 \cdot H$ leading to the new Eulerian multiset of arcs $H' = 3 \cdot H - W'$. For each $a \in A$, we have $3(H - k \cdot C)[a] \geqslant H'[a] \geqslant 3(H - k \cdot C)[a] - 2$, because $D[a] \leqslant 2$. Hence $A_{H'} = A_{H-k\cdot C}$. Consequently, H' is connected, $\|H'\| < \|H\|$, and $\hat{q} \in Q_{H'}$ if $H' \neq \vec{0}$. ∎

Algorithm 2 Computing a multiset of wings

Require: A non-empty circulation $\hat{H}$ and a state $\hat{q} \in Q_{\hat{H}}$
  $\mathcal{W} \leftarrow \vec{0}$ # Initially $\mathcal{W}$ is the empty multiset of wings
  $H \leftarrow \hat{H}$ # Initially $\mathrm{cost}(\mathcal{W}) + \mathrm{cost}(H) = m \cdot \mathrm{cost}(\hat{H})$ with $m = 1$
  while $H \neq \vec{0}$ **do**
    $C \leftarrow$ AppropriateCircuit$(H, \hat{q}, \epsilon)$ # C is adequate for H and $\hat{q}$.
    Let k be the multiplicity of C in H # $k \cdot C \leqslant H$ and $H - k \cdot C$ is connected
    if $\hat{q} \in Q_C$ **then**
      $D \leftarrow \vec{0}$ # $D \in \mathbb{N}^A$ is the empty multiset of arcs
      $W \leftarrow k \cdot C$ # The multiset W represents a wing such that $W \leqslant H$
    else
      Let q be some state in $Q_C \cap Q_{H-k \cdot C}$.
      Let γ_1 be a shortest path from $\hat{q}$ to q made of arcs from $A_{H-k \cdot C}$. # γ_1 is a simple path
      Let γ_2 be a shortest path from q to $\hat{q}$ made of arcs from $A_{H-k \cdot C}$. # γ_2 is a simple path
      Let D be the multiset of arcs that corresponds to the cycle $\gamma_1 . \gamma_2$. # Then $D \leqslant 2 \cdot H$
      $W \leftarrow D + k \cdot C$ # The multiset W represents a reduced wing such that $W \leqslant 3 \cdot H$
    end if
    if $(H = W)$ or $(W \leqslant H$ and $H - W$ is connected and $\hat{q} \in Q_{H-W})$ **then**
      Add the wing W to $\mathcal{W}$.
      $H \leftarrow H - W$ # $\mathrm{cost}(\mathcal{W}) + \mathrm{cost}(H) = m \cdot \mathrm{cost}(\hat{H})$ for some $m \geqslant 1$
    else
      $W' \leftarrow D + 3k \cdot C$ # W' is a reduced wing; moreover we have $A_{H-k \cdot C} = A_{3 \cdot H - W'}$
      $\mathcal{W} \leftarrow 3 \cdot \mathcal{W}$ # $\mathrm{cost}(\mathcal{W}) + \mathrm{cost}(3 \cdot H) = m \cdot \mathrm{cost}(\hat{H})$ for some $m \geqslant 1$
      Add the wing W' to $\mathcal{W}$.
      $H \leftarrow 3 \cdot H - W'$ # $\mathrm{cost}(\mathcal{W}) + \mathrm{cost}(H) = m \cdot \mathrm{cost}(\hat{H})$ for some $m \geqslant 1$
    end if
  end while
  return $\mathcal{W}$

Thus, in all cases we get that H' is Eulerian and connected. Moreover $\hat{q} \in Q_{H'}$ provided that H' is not empty and hence the next iteration of the algorithm can proceed analogously. Furthermore we have $\|H'\| < \|H\|$ henceforth Alg. 2 terminates after at most $|A|$ iterations.

Example 4.5. We continue Examples 1.1 and 4.2 to illustrate an execution of Alg. 2 with the VASS depicted in Figure 1, the circulation $\hat{H} = a_1 + a_2 + a_3 + 5 \cdot l_1 + 3 \cdot l_2$, and the base state $\hat{q} = q_0$. First, the adequate circuit l_1 with multiplicity 5 can be chosen which leads to the wing $W_1 = a_1 + a_2 + a_3 + 5 \cdot l_1$. Since $\hat{H} - W_1$ does not contain $\hat{q}$, we put $W'_1 = a_1 + a_2 + a_3 + 15 \cdot l_1$ and get $\mathcal{W} = \{W'_1\}$ and $H = 3 \cdot \hat{H} - W'_1 = 2 \cdot a_1 + 2 \cdot a_2 + 2 \cdot a_3 + 9 \cdot l_2$ at the end of the first iteration.

In the second iteration, l_2 is the unique adequate circuit for H and $\hat{q}$. Therefore we put $W_2 = a_1 + a_2 + a_3 + 9 \cdot l_2$ and get $\mathcal{W} = \{W'_1, W_2\}$ and $H' = H - W_2 = a_1 + a_2 + a_3$ because this Eulerian multiset of arcs is connected and contains $\hat{q}$. The third and last iteration selects the adequate circuit $W_3 = a_1 + a_2 + a_3$ which yields the multiset of wings $\mathcal{W} = \{W'_1, W_2, W_3\}$ depicted in Fig. 6. Observe here that $\mathrm{cost}(\mathcal{W}) = (3, 12)^\top = 3 \cdot \mathrm{cost}(\hat{H})$.

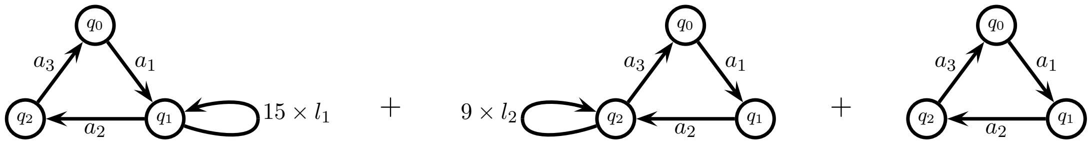

Figure 6. Multiset of wings computed in Example 4.5

It is clear that the property that $\mathrm{cost}(\mathcal{W}) + \mathrm{cost}(H) = m \cdot \mathrm{cost}(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$ is a loop invariant of Algorithm 2. Consequently,

Theorem 4.6. Let $\hat{H}$ be a non-empty circulation of a VASS $\mathcal{S}$ and $\hat{q} \in Q_{\hat{H}}$. Algorithm 2 returns in polynomial time a non-empty multiset $\mathcal{W}$ of reduced wings starting from $\hat{q}$ such that $\mathrm{cost}(\mathcal{W}) = m \cdot \mathrm{cost}(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$.

Clearly the multiset $\mathcal{W}$ built by Algorithm 2 is made of at most $|A|$ wings. Moreover the valuation of each wing in $\mathcal{W}$ is at most $3^{|A|} \times \max_{a \in A} \hat{H}[a]$.

4.3. An upper bound for the number of distinct wings

Since Algorithm 2 terminates in less than $|A|$ iterations, it provides us with a multiset $\mathcal{W}$ of wings starting from the arbitrarily fixed state $\hat{q}$ with at most $|A|$ distinct wings. To conclude the proof of Theorem 3.9, we show that we can make sure that the representative multiset $\mathcal{W}$ contains at most p distinct wings. This results essentially from Carathéodory's theorem [22, Cor. 7.7i] which states that for each set $X \subseteq \mathbb{Q}^p$ of p-dimensional rational vectors, any rational vector $v \in \mathbb{Q}^p$ that lies in Cone$(X) = \{\lambda_1 \cdot x_1 + ... + \lambda_n \cdot x_n \mid n \geqslant 1; x_1, ..., x_n \in X; \lambda_1, ..., \lambda_n \in \mathbb{Q}^+\}$ lies in Cone(X') for some $X' \subseteq X$ with $|X'| \leqslant p$, i.e. $v = \lambda_1 \cdot x_1 + ... + \lambda_n \cdot x_n$ with $p \geqslant n \geqslant 1$, $x_1, ..., x_n \in X$ and $\lambda_1, ..., \lambda_n \in \mathbb{Q}^+$. We implement and adapt this property to integral vectors and natural numbers as follows:

Corollary 4.7. Let $x_1, ..., x_n \in \mathbb{Z}^p$ be n integral vectors and $\lambda_1 \cdot x_1 + ... + \lambda_n \cdot x_n = z$ be a linear combination with $\lambda_1, ..., \lambda_n \in \mathbb{N}$. If $n > p$, we can compute in polynomial time $\lambda'_1, ..., \lambda'_n \in \mathbb{N}$ and $m \in \mathbb{N} \setminus \{0\}$ such that $\lambda'_1 \cdot x_1 + ... + \lambda'_n \cdot x_n = m \cdot z$ and $\lambda'_i \neq 0$ for at most $n - 1$ values of $i \in [1..n]$.

Proof. We can assume that the n vectors x_i are distinct and that all natural numbers λ_1,..., λ_n are positive. Since $n > p$, the n vectors x_i are linearly dependent: There are rational numbers μ_1,..., μ_n not all zero such that $(*) \sum_{i=1}^{n} \mu_i \cdot x_i = \vec{0}$. These rational numbers can be computed in polynomial time by solving the following n linear programs

$$(P_j) \begin{cases} \sum_{i=1}^{n} \mu_i \cdot x_i = \vec{0} \\ \mu_j \geqslant 1 \end{cases}$$

for $j \in [1..n]$. We can assume that $\mu_i \in \mathbb{Z}$ for each $i \in [1..n]$ —because we can derive a non-zero integral solution to $(*)$ from a rational one with the help of Euclid's algorithm, again. Further, we have $\mu_j \geqslant 1$ for some $j \in [1..n]$.

Recall that $\lambda_i \geqslant 1$ for each $i \in [1..n]$. Let $k \in [1..n]$ be such that

$$\frac{\mu_k}{\lambda_k} = \max_{i \in [1..n]} \frac{\mu_i}{\lambda_i}$$

Then $\mu_k > 0$ because $\mu_j \geqslant 1$ for some $j \in [1..n]$. Moreover $\lambda_i.\mu_k - \lambda_k.\mu_i \geqslant 0$ for each $i \in [1..n]$. Furthermore we have $\sum_{i=1}^{n}(\lambda_i.\mu_k - \lambda_k.\mu_i) \cdot x_i = \mu_k \cdot z$. To conclude, we put $\lambda'_i = \lambda_i.\mu_k - \lambda_k.\mu_i$ and observe that $\lambda'_k = 0$. ∎

We can apply iteratively Cor. 4.7 to the cost of the multiset of wings $\mathcal{W} = \lambda_0 \cdot W_0 + ... + \lambda_n \cdot W_n$ produced by Alg. 2 in order to compute a multiset $\mathcal{W}'$ built over at most p reduced wings starting from $\hat{q}$ and such that $\mathrm{cost}(\mathcal{W}') = m' \cdot \mathrm{cost}(\mathcal{W})$ for some $m' \in \mathbb{N} \setminus \{0\}$. Since $\mathrm{cost}(\mathcal{W}) = m \cdot \mathrm{cost}(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$, we get $\mathrm{cost}(\mathcal{W}') = m'' \cdot \mathrm{cost}(\hat{H})$ for some $m'' \in \mathbb{N} \setminus \{0\}$, too. Since our algorithm is polynomial, the size of the valuation of these wings and the size of the number of occurrences of these wings are polynomial in the size of the inputs $\mathcal{S}$ and $\hat{H}$.

Example 4.8. We continue Example 4.5 to illustrate how Alg. 2 and Corollary 4.7 lead to Theorem 3.9. We have obtained that the multiset of wings $\mathcal{W} = 1 \cdot W'_1 + 1 \cdot W_2 + 1 \cdot W_3$ satisfies $\mathrm{cost}(\mathcal{W}) = (3, 12)^\top$ because $\mathrm{cost}(W'_1) = (-18, 27)^\top$, $\mathrm{cost}(W_2) = (24, -12)^\top$, and $\mathrm{cost}(W_3) = (-3, -3)^\top$. Using a linear programming solver yields: $4 \cdot \mathrm{cost}(W'_1) + 5 \cdot \mathrm{cost}(W_2) + 16 \cdot \mathrm{cost}(W_3) = \vec{0}$. Consequently W_3 can be removed from $\mathcal{W}$ and we get another pathological multiset of wings $\mathcal{W}' = 12 \cdot W'_1 + 11 \cdot W_2$ with $\mathrm{cost}(\mathcal{W}') = 16 \cdot \mathrm{cost}(\mathcal{W})$.

This example concludes the illustration of the construction of wings from circulations. Note here that the simpler multiset of wings $\mathcal{W}'' = 1 \cdot W'_1 + 2 \cdot W_2$ is also pathological. However, since multisets of wings are only an intermediate format, it is no use in practice to reduce their iteration factors at this point.

5. Construction of a flower from a multiset of wings

In this section, we fix a non-empty circulation $\hat{H}$ and present a proof of Theorem 3.4. We observe first that it is possible to build a flower directly from $\hat{H}$ as follows: First, one builds a cycle that corresponds to $\hat{H}$; next, one extracts petals from it by detecting iterated states along the cycle; finally, one can reduce the number of petals to less than p, as we will see below. However this approach requires exponential space since the length of the cycle is exponential in the size of the circulation (and may be exponential in the size of the VASS $\mathcal{S}$ as shown by Example 2.5). For that reason, we adopt another strategy that relies on the representation by wings.

First, we apply Theorem 3.9 to get a non-empty multiset $\mathcal{W}$ of reduced wings starting from a fixed state $\hat{q}$ such that $\mathrm{cost}(\mathcal{W}) = m \cdot \mathrm{cost}(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$ and $\mathcal{W}$ is built over at most p distinct wings. Then we build from $\mathcal{W}$ a flower $\mathcal{F}$ with at most p petals and such that $\mathrm{cost}(\mathcal{F}) = m' \cdot \mathrm{cost}(\mathcal{W})$ for some $m' \in \mathbb{N} \setminus \{0\}$. To do so, we proceed in three steps. We observe first that, intuitively, an iterated wing can be regarded as a flower. Formally we show that for each wing $\omega = \gamma_1.\gamma_0^k.\gamma_2$ with valuation k, and for all $x \geqslant |Q| + 1$, we can build a flower $\mathcal{F}_{\omega,x}$ starting from the starting state of γ_0 such that $\mathrm{cost}(\omega^x) = \mathrm{cost}(\gamma_1) + \mathrm{cost}(\mathcal{F}_{\omega,x}) + \mathrm{cost}(\gamma_2)$. Moreover, $\mathcal{F}_{\omega,x}$ does not iterate its calyx. Next we show how to connect the flowers associated to each iterated wing of $\mathcal{W}$ in order to get a single flower $\mathcal{F}$. At this point, we assume that each wing is iterated at least $|Q| + 1$ times. For that reason, $\mathcal{F}$ has the same cost as $\mathcal{W}$, up to a multiplication factor. This flower has at most $3 \times |Q| \times p$ petals. Finally, we show how to use again Cor. 4.7 to reduce the number of petals in the representative flower to less than p.

As opposed to the previous section, we do not give the construction as formal algorithms. We rather provide sufficient details to explain and to justify the global procedure. We will also illustrate the construction of flowers on our simple running example of Fig. 1.

Example 5.1. We consider again the VASS from Fig. 1 and the two reduced wings $\omega_1 = a_1.l_1^{15}.(a_2.a_3)$ and $\omega_2 = (a_1.a_2).l_2^9.a_3$ with valuation 15 and 9 respectively. The cycle $\omega_1.\omega_2^2$ is pathological since $\mathrm{cost}(\omega_1.\omega_2^2) = (30,3)^\top$. For each natural number $x \geqslant 2$, we have $\mathrm{cost}(\omega_2^x) = \mathrm{cost}(a_1.a_2) + \mathrm{cost}(\mathcal{F}_{2,x}) + \mathrm{cost}(a_3)$ where $\mathcal{F}_{2,x}$ denotes the flower with an empty calyx and two petals: l_2 that is iterated $9 \times x$ times, and $a_3.a_1.a_2$ that is iterated $x-1$ times.

The transformation of an iterated wing $(\gamma_1.\gamma_0^k.\gamma_2)^x$ into a flower is not always as easy as in the above example, because the connecting cycle $\gamma_1.\gamma_2$ needs not to be simple in general. Consequently, we have to extract petals from it.

5.1. From an iterated reduced wing to a flower

Consider a reduced wing $\omega = \gamma_1.\gamma_0^k.\gamma_2$. The special case where ω consists only in its cyclic component is trivial. Since ω is reduced, we can assume that both γ_1 and γ_2 are not empty and their starting state differ. The following basic remark is illustrated by Fig. 7.

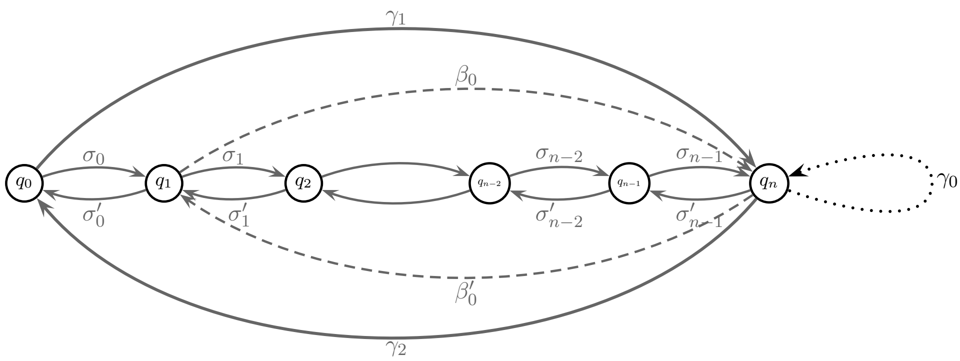

Figure 7. Extracting intrinsic petals from the connecting cycle of a wing

Proposition 5.2. Let q and $q' \in Q$ be two distinct states and γ_1 and γ_2 be simple paths from q to q' and from q' to q respectively such that q' differs from the domain of each arc of γ_1 and q differs from the domain of each arc of γ_2. We can compute in polynomial time

- a natural number $n \geqslant 1$ smaller than the length of γ_1,
- $n+1$ states $q_0, \ldots, q_n \in Q$ with $q_0 = q$ and $q_n = q'$,
- n simple paths $\sigma_0, \ldots, \sigma_{n-1}$ where σ_i is a path from q_i to q_{i+1}, and
- n simple paths $\sigma'_0, \ldots, \sigma'_{n-1}$ where σ'_i is a path from q_{i+1} to q_i,

such that $\gamma_1 = \sigma_0 \ldots \sigma_{n-1}$, $\gamma_2 = \sigma'_{n-1} \ldots \sigma'_0$, and for each $i \in [0..n-1]$, the cycle $\sigma_i.\sigma'_i$ is a circuit.

Proof. We proceed by induction over the length of γ_1. The claim is clear if $|\gamma_1| = 1$ because the cycle $\gamma_1.\gamma_2$ is simple. Induction step. We can assume that the cycle $\gamma_1.\gamma_2$ is not simple: We can find along the path $\gamma_1.\gamma_2$ some state $\widetilde{q}$ that occurs twice and such that no other state occurs twice between these two occurrences of $\widetilde{q}$. Then $\widetilde{q} \neq q$ because γ_1 is a simple path and q differs from the domain of each arc of γ_2. Similarly $\widetilde{q} \neq q'$. Since γ_1 and γ_2 are both simple paths, we have

- $\gamma_1 = \gamma_1'.\gamma_1''$ where γ_1' leads from q to $\widetilde{q}$ and γ_1'' leads from $\widetilde{q}$ to q';
- $\gamma_2 = \gamma_2''.\gamma_2'$ where γ_2'' leads from q' to $\widetilde{q}$ and γ_1' leads from $\widetilde{q}$ to q.

Moreover γ_1'' and γ_2'' are not empty since $\widetilde{q} \neq q'$. Since no state occurs twice between the two occurrences of $\widetilde{q}$, the cycle $\gamma_1''.\gamma_2''$ starting from $\widetilde{q}$ is a circuit. It is now sufficient to apply the induction hypothesis to γ_1' and γ_2' because $|\gamma_1'| \leqslant |\gamma_1| - 1$. ■

This decomposition allows us to represent an iterated wing as a flower.

Proposition 5.3. Let $\omega = \gamma_1.\gamma_0^k.\gamma_2$ be a reduced wing starting from q with valuation $k \geqslant 1$ and q' be the starting state of its cyclic component γ_0. Let $x > |Q|$. We can build in polynomial time a flower $\mathcal{F}_{\omega,x}$ starting from q' with at most $2 \times |Q| - 1$ petals such that

- the calyx is not iterated,
- the first petal at q' is iterated at least twice,
- $x \cdot \mathrm{cost}(\omega) = \mathrm{cost}(\gamma_1) + \mathrm{cost}(\mathcal{F}_{\omega,x}) + \mathrm{cost}(\gamma_2)$.

Proof. The particular case where $q = q'$ is trivial because ω is simply an iterated circuit starting from q'. Consequently both connecting paths γ_1 and γ_2 are empty: The flower $\mathcal{F}_{\omega,x}$ has an empty calyx and it admits γ_0 as unique petal. We assume now that $q \neq q'$. We apply first Prop. 5.2 and get a natural number n with $n \in [1..|Q|]$, $n+1$ states $q_0, \ldots, q_n \in Q$ with $q_0 = q$ and $q_n = q'$, n simple paths $\sigma_0, \ldots, \sigma_{n-1}$ where σ_i is a path from q_i to q_{i+1}, and n simple paths $\sigma_0', \ldots, \sigma_{n-1}'$ where σ_i' is a path from q_{i+1} to q_i such that $\gamma_1 = \sigma_0 \ldots \sigma_{n-1}$, $\gamma_2 = \sigma_{n-1}' \ldots \sigma_0'$, and for each i, the cycle $\sigma_i.\sigma_i'$ is a circuit. We have $x > |Q| \geqslant n$. The situation is depicted in Fig. 7. We consider

- the circuit $\alpha_i = \sigma_i'.\sigma_i$ that starts from q_{i+1} for each $i \in [0, n-1]$,
- the simple path $\beta_i = \sigma_{i+1} \ldots \sigma_{n-1}$ from q_{i+1} to q_n for each $i \in [0, n-2]$, and
- the simple path $\beta_i' = \sigma_{n-1}' \ldots \sigma_{i+1}'$ from q_n to q_{i+1} for each $i \in [0, n-2]$.

Clearly the path

$$\gamma = \gamma_0^{k(x-n+1)}.(\gamma_0^k.\beta_0'.\alpha_0^{x-1}.\beta_0).(\gamma_0^k.\beta_1'.\alpha_1^{x-2}.\beta_1) \ldots (\gamma_0^k.\beta_{n-2}'.\alpha_{n-2}^{x-n+1}.\beta_{n-2}).\alpha_{n-1}^{(x-n)}$$

is a cycle that starts from q'. Moreover this cycle corresponds to a flower $\mathcal{F}_{\omega,x}$ with $2n-1$ petals and a calyx equal to $\beta_0'.\beta_0.\beta_1'.\beta_1 \ldots \beta_{n-2}'.\beta_{n-2}$ that is not iterated. This flower starts from q' with a first petal γ_0 that is iterated $k(x-n+2)$ times. Observe that σ_{n-1} occurs in the $x-n$ occurrences of the petal α_{n-1} and in each of the $n-1$ paths β_i. Further, for each $0 \leqslant j \leqslant n-2$, the path σ_j occurs $x-1-j$ times within the petal α_j and j times within the calyx. The occurrences of the opposite paths σ_j' are analogous. Thus we have

$$\mathrm{cost}(\mathcal{F}_{\omega,x}) = x \cdot \mathrm{cost}(\gamma_0^k) + (x-1) \cdot \sum_{i=0}^{n-1} (\mathrm{cost}(\sigma_i) + \mathrm{cost}(\sigma_i')) = x \cdot \mathrm{cost}(\gamma_0^k) + (x-1) \cdot \mathrm{cost}(\gamma_1.\gamma_2).$$

■

5.2. From a multiset of wings to a flower

We explain now how to connect the flowers associated to iterated wings with a common starting state in order to build a single representative flower for $\mathcal{W}$. Recall that these flowers come equipped with the two connecting paths back and forth from the fixed starting state $\hat{q}$. Moreover they do not iterate their calyx. For that reason, we can join easily these flowers into a single one that does not iterate its calyx either. However, some new petals may appear in this process, as shown by the next example.

Example 5.4. We continue Example 5.1 and consider the multiset of wings $\omega_1 + 2 \cdot \omega_2$. Making use of the representation of the iterated wing ω_2^2 by the flower $\mathcal{F}_{2,2}$ we shall obtain a flower description of $\omega_1 + 2 \cdot \omega_2$ as a new flower $\mathcal{F}_{\omega_1+2\cdot\omega_2} = a_1.l_1^{15}.(a_2.a_3.a_1).a_2.l_2^{18}.(a_3.a_1.a_2).a_3$ with four petals: l_1, l_2, $a_2.a_3.a_1$, and $a_3.a_1.a_2$.

Proposition 5.5. Let $\mathcal{W}$ be a multiset of wings starting from $\hat{q}$ built over at most p distinct reduced wings. We can build in polynomial time a flower $\mathcal{F}$ with at most $3 \times |Q| \times p$ petals, no calyx iteration, and such that $\mathrm{cost}(\mathcal{F}) = m \cdot \mathrm{cost}(\mathcal{W})$ for some $m \in \mathbb{N} \setminus \{0\}$.

Proof. Let $\omega_0, \ldots, \omega_{l-1}$ be the l reduced wings starting from $\hat{q}$ that occur at least once in $\mathcal{W}$. We have $\mathrm{cost}(\mathcal{W}) = \sum_{i=0}^{l-1} \mathcal{W}[\omega_i] \cdot \mathrm{cost}(\omega_i)$. We can assume w.l.o.g. that $\mathcal{W}[\omega_i] \geqslant |Q| + 1$ for each wing ω_i: If this property does not hold, we can replace $\mathcal{W}$ by $(|Q| + 1) \cdot \mathcal{W}$ —due to the multiplication factor m in the statement of Prop. 5.5. Each wing ω_i is made of a cyclic component $\gamma_{i,0}$ starting from q_i and two paths $\gamma_{i,1}$ and $\gamma_{i,2}$ from $\hat{q}$ to q_i and from q_i to $\hat{q}$ respectively. Then $\omega_i = \gamma_{i,1}.\gamma_{i,0}^{k_i}.\gamma_{i,2}$ where $k_i \geqslant 1$ is the valuation of ω_i.

By Prop. 5.3, the cycle $\gamma_{i,0}^{k_i \mathcal{W}[\omega_i]}.(\gamma_{i,2}.\gamma_{i,1})^{\mathcal{W}[\omega_i]-1}$ is equivalent to a flower $\mathcal{F}_i$ with at most $2 \times |Q| - 1$ petals, no calyx iteration, and such that the first petal at q_i is iterated at least twice. In order to connect these l flowers, we consider the l paths $\kappa_i = \gamma_{i,2}.\gamma_{i+1 \pmod{l},1}$ from q_i to $q_{i+1 \pmod{l}}$ for each $i \in [0..l-1]$. Although $\gamma_{i,2}$ and $\gamma_{i+1 \pmod{l},1}$ are simple, κ_i need not to be simple. However, if κ_i is not simple, it can be decomposed into a simple path followed by a circuit that acts as a petal, followed by a simple path, similarly to the proof of Prop. 5.2: It is sufficient to find along κ_i some state $\widetilde{q}$ that occurs twice and such that no other state occurs twice between these two occurrences of $\widetilde{q}$. Let $\gamma_{\mathcal{F}_i}$ denote the cycle corresponding to the flower $\mathcal{F}_i$. Then the cycle $\gamma = \gamma_{\mathcal{F}_0}.\kappa_0.\gamma_{\mathcal{F}_1}.\kappa_1 \ldots \gamma_{\mathcal{F}_{l-1}}.\kappa_{l-1}$ can be regarded as a flower $\mathcal{F}_\gamma$ when all but one occurrences of first petal of $\mathcal{F}_i$ are connected to the beginning of κ_i and the last one is connected to the end of $\kappa_{i-1 \pmod{l}}$. This flower has at most $(2 \times |Q| + 1) \times l$ petals and no calyx iteration. Observe that $\sum_{i=0}^{l-1} \mathrm{cost}(\kappa_i) = \sum_{i=0}^{l-1} \mathrm{cost}(\gamma_{i,2}.\gamma_{i,1})$. Hence $\mathrm{cost}(\mathcal{F}_\gamma) = \sum_{i=0}^{l-1} \mathrm{cost}(\mathcal{F}_i) + \mathrm{cost}(\gamma_{i,2}.\gamma_{i,1}) = \sum_{i=0}^{l-1} \mathcal{W}[\omega_i] \cdot \mathrm{cost}(\omega_i) = \mathrm{cost}(\mathcal{W})$. ■

5.3. Reducing the number of petals in a flower

To conclude the proof of Theorem 3.4, we need to reduce the number of petals in $\mathcal{F}$ from $3 \times |Q| \times p$ to p. This is done by Cor. 5.8 below. Due to Carathéodory's theorem, if there are more than $p + 1$ petals then one of them is redundant and can be removed, provided that we adapt the iteration numbers of the calyx and of the remaining petals. However this removal of petals can lead to connection paths between the remaining petals that are no longer simple paths. That is why some new petals can appear also along this last step. This remark is formalized by the following observation and illustrated by Fig. 8.

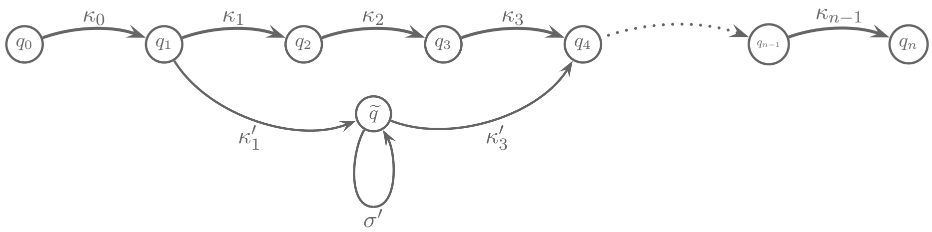

Figure 8. Reducing the size of the calyx

Proposition 5.6. Let $q_0, \ldots, q_n$ be $n+1$ states and $\kappa_0, \ldots, \kappa_{n-1}$ be n simple paths such that κ_i leads from q_i to q_{i+1}. Let $\gamma = \kappa_0 \ldots \kappa_{n-1}$ be the resulting path from q_0 to q_n. If γ is not simple then we can compute in polynomial time

- $k+1$ states $q'_0, \ldots, q'_k$ with $k \leqslant n$, $q'_0 = q_0$, and $q'_k = q_n$,
- k simple paths $\kappa'_0, \ldots, \kappa'_{k-1}$ such that κ'_i leads from q'_i to q'_{i+1}, and
- $k-1$ circuits $\sigma'_1, \ldots, \sigma'_{k-1}$ which start from q_i

such that $\gamma = \kappa'_0.\sigma'_1.\kappa'_1\sigma'_2 \ldots \sigma'_n.\kappa'_n$.

Proof. We proceed by induction over n. The base case for $n = 1$ is trivial because κ_0 is simple. Induction step: We assume that $n \geqslant 2$ and γ is not simple. We can find along γ some state $\widetilde{q}$ that occurs twice and such that no other state occurs twice between these two occurrences of $\widetilde{q}$. We assume that these two occurrences of $\widetilde{q}$ take place in κ_j and $\kappa_{j'}$ with $j < j'$. The path κ_j can be split into two simple paths $\kappa_{j,1}$ and $\kappa_{j,2}$ such that $\kappa_{j,1}$ leads from q_j to $\widetilde{q}$ and $\kappa_{j,2}$ leads from $\widetilde{q}$ to q_{j+1}. Similarly, the path $\kappa_{j'}$ can be split into two simple paths $\kappa_{j',1}$ and $\kappa_{j',2}$ such that $\kappa_{j',1}$ leads from $q_{j'}$ to $\widetilde{q}$ and $\kappa_{j',2}$ leads from $\widetilde{q}$ to $q_{j'+1}$. Then the cycle $\sigma' = \kappa_{j,2}.\kappa_{j+1} \ldots \kappa_{j'-1}.\kappa_{j',1}$ starting from $\widetilde{q}$ is non empty and simple. Furthermore the path $\gamma_1 = \kappa_0 \ldots \kappa_{j-1}.\kappa_{j,1}$ from q_0 to $\widetilde{q}$ is made of at most $j+1$ simple paths, where $j+1 \leqslant j' \leqslant n-1$. Similarly, the path $\gamma_2 = \kappa_{j',2}.\kappa_{j'+1} \ldots \kappa_{n-1}$ from $\widetilde{q}$ to q_n is made of at most $n-j'$ simple paths. Thus we can apply the induction hypothesis to both γ_1 and γ_2 and get the expected decomposition of γ with at most $j+1+n-j' \leqslant n$ component simple paths κ'_i. ■

As already stressed by Example 3.6, the iteration of the calyx is necessary to get a flower with at most p petals. Thus, with no surprise, removing petals requires finally to allow for the iteration of a calyx.

Proposition 5.7. Let $\mathcal{F}$ be a flower with $k \geqslant p+1$ petals. We can compute in polynomial time a flower $\mathcal{F}'$ such that $\mathrm{cost}(\mathcal{F}') = m \cdot \mathrm{cost}(\mathcal{F})$ for some $m \in \mathbb{N} \setminus \{0\}$ and

- either $\mathcal{F}'$ has at most $k-1$ petals,
- or $\mathcal{F}'$ has k petals and the length of its calyx is strictly smaller than the length of the calyx of $\mathcal{F}$.

Proof. Let $\mathcal{F}$ be a flower consisting of

- a sequence of $k \geqslant 2$ petals $\sigma_0, \ldots, \sigma_{k-1}$, where each circuit σ_i starts from q_i,
- a calyx $\kappa_0 \ldots \kappa_{k-1}$, where κ_i is a simple path from q_i to $q_{i+1 \pmod k}$,
- and a sequence $n_0, \ldots, n_k$ of natural numbers, with $n_i \geqslant 1$.

We have $\mathrm{cost}(\mathcal{F}) = n_k \cdot \mathrm{cost}(\kappa_0 \ldots \kappa_{k-1}) + \sum_{i=0}^{k-1} n_i \cdot \mathrm{cost}(\sigma_i)$. According to Cor. 4.7, we can compute in polynomial time a sequence $n'_0, \ldots, n'_{k-1}$ of natural numbers such that $\sum_{i=0}^{k-1} n'_i \cdot \mathrm{cost}(\sigma_i) = m \cdot (\mathrm{cost}(\mathcal{F}) - n_k \cdot \mathrm{cost}(\kappa_0 \ldots \kappa_{k-1}))$ for some $m \in \mathbb{N} \setminus \{0\}$ and $n'_i \neq 0$ for at most $k-1$ values of $i \in [0..k-1]$. We consider the path

$$\gamma' = \sigma_0^{n'_0}.\kappa_0.\sigma_1^{n'_1} \ldots \sigma_{k-1}^{n'_{k-1}} \kappa_{k-1}.(\kappa_0 \ldots \kappa_{k-1})^{m.n_k - 1}$$

We have $\mathrm{cost}(\gamma') = m \cdot \mathrm{cost}(\mathcal{F})$. If the calyx $\kappa_0 \ldots \kappa_{k-1}$ of $\mathcal{F}$ is empty, γ' corresponds to a flower with at most $k-1$ petals. Therefore we can assume now that the calyx of $\mathcal{F}$ is not empty.

We assume first that for each $i \in [0, k-1]$ and each $r \in [2..k]$ such that $n'_{i+i' \pmod k} = 0$ for all $i' \in [1, r-1]$, the path $\kappa'_i = \kappa_i.\kappa_{i+1 \pmod k} \ldots \kappa_{i+r-1 \pmod k}$ is simple. If $n'_i \neq 0$ for at least two i's then γ' corresponds to a flower with at most $k-1$ petals. On the other hand, if $n'_i \neq 0$ for at most one i then the calyx $\kappa_0 ... \kappa_{k-1}$ is a circuit and γ' corresponds to a flower with at most 2 petals and an empty calyx.

We can assume now that there are $i \in [0, k-1]$ and $r \in [2..k]$ such that $n'_{i+i' \pmod k} = 0$ for all $i' \in [1, r-1]$ and the path $\kappa'_i = \kappa_i.\kappa_{i+1 \pmod k} \ldots \kappa_{i+r-1 \pmod k}$ is *not* simple. At this point we can use Prop. 5.6 to regard γ' as a flower with at most k petals and whose calyx is smaller than the calyx of $\mathcal{F}$. ■

Corollary 5.8. Let $\mathcal{F}$ be a flower. We can compute in polynomial time a flower $\mathcal{F}'$ with at most p petals and such that $\mathrm{cost}(\mathcal{F}') = m \cdot \mathrm{cost}(\mathcal{F})$ for some $m \in \mathbb{N} \setminus \{0\}$.

Proof. Observe that Prop. 5.7 yields a flower with at most $k-1$ petals if the calyx of the given flower is empty. Consider a flower with $k \geqslant p+1$ petals. First we apply Prop. 5.7 iteratively until we get a flower with at most $k-1$ petals. Next we iterate this first step until we get a flower with at most p petals. ■

Example 5.9. We continue Example 5.4 to illustrate the last step of the proof of Theorem 3.4. The flower $\mathcal{F}_{\omega_1+2\cdot\omega_2} = a_1.l_1^{15}.(a_2.a_3.a_1).a_2.l_2^{18}.(a_3.a_1.a_2).a_3$ has four petals: l_1, l_2, $a_2.a_3.a_1$, and $a_3.a_1.a_2$. Since $a_2.a_3.a_1$ and $a_3.a_1.a_2$ have the same cost, one of them can be removed and we get the new flower $\mathcal{F}' = a_1.l_1^{15}.(a_2.a_3.a_1)^2.a_2.l_2^{18}.a_3$ with only three petals. Further, using a linear programming solver yields: $12 \cdot \mathrm{cost}(l_1) + 9 \cdot \mathrm{cost}(l_2) + 5 \cdot \mathrm{cost}(a_2.a_3.a_1) = \vec{0}$. Similarly to the proof of Cor. 4.7, we observe that $5/2 > 12/15 > 9/18$, so we can remove the petal $a_2.a_3.a_1$ from $\mathcal{F}'$ and get a last flower $\mathcal{F}'' = a_1.l_1^{51}.a_2.l_2^{72}.a_3.(a_1.a_2.a_3)^4$ with only two petals l_1 and l_2, and a calyx that is iterated 5 times. Note that $\mathrm{cost}(\mathcal{F}'') = 5 \cdot \mathrm{cost}(\mathcal{F}_{\omega_1+2\cdot\omega_2})$.

This example concludes the illustration of the construction of a flower from a multiset of wings. Note here that the simpler flower $\mathcal{F}''' = a_1 l_1^5.a_2.l_2^3.a_3$ is also pathological and shares with $\mathcal{F}''$ its structure: They have the same petals and the same calyx. Therefore, in practice, it might be useful to reduce the iteration factors of the resulting flower at the end of this process, in order to provide the user with a simpler counter-example.

6. Conclusion and future work

In this paper we tackle the problem of illustrating a structural bug detected as a pathological circulation in a concise way. We propose to represent counter-examples for structural termination in the form of a flower, that is, iterated circuits connected by simple paths. Our main result shows how to compute such a structure in polynomial time (Th. 3.4) from any given pathological circulation. Further we need only p iterated circuits in such a flower. Interestingly this result applies immediately to structural boundedness. We can draw a parallel between this setting and the emptiness problem for Büchi automata. Accepting infinite words are detected by means of lasso-like words $u.v^{\omega}$ where u leads from the initial state to an accepting state and v is a circuit over this accepting state. Besides the title of this paper is inspired by Trivedi's blog [25] that suggests another title for the seminal paper [26]. Indeed we claim that flowers should play the rôle of lassos when it comes to structural properties of a VASS.

We give a slightly involved construction of flowers that makes use of an intermediate representative format as a multiset of wings (Th. 3.9), a result first presented in [2]. It would be nice to build flowers from circulations directly, possibly with adequate circuits as inputs. However, so far, we failed to design such an alternative construction or any simpler valid approach. The point is that adding a petal to a flower may require to adapt and, more important, to iterate the calyx. This constraint corresponds to the multiplication by 3 of the given circulation (and the partially constructed multiset of wings) in Algorithm 2 in order to keep the remaining circulation connected when a new wing is built. The iteration of the calyx makes the insertion of further petals more difficult: Intuitively, the goal would be similar to building a representative multiset of wings such that each wing is iterated the same number of times. Besides, we have observed that flowers that do not iterate their calyx are easier to handle. Furthermore the construction of a flower from a multiset of wings consists essentially in concentrating the iteration factors in the petals and simplifying the description of the connecting paths into a single calyx. Anyway, a simpler construction of a flower from a circulation would be valuable.

Several technical steps detailed in this paper rely on the multiplication factor allowed from the cost of the given circulation to the cost of the representative flower. This feature is necessary to keep the structure connected along its construction and to reduce the number of iterated components at the end. It is also useful to regard iterated wings as flowers. We have observed that it is a good idea to try to reduce the values of iterations within a flower built in this way, in order to get a new simpler flower with the same structure but with no particular link with the cost of the original pathological circulation. Finding alternative iteration values with a bounded size or even minimizing their sum by integral linear programming will probably prove to be useful in practice to get a simpler flower. Finding shortest counter-examples is often desirable in automated verification, because they are easier to analyse, see e.g. [5, 16]. Thus, considering pathological cycles built over a minimal number of arcs, or with a minimal number of interacting places, is certainly valuable. Unfortunately, searching for such circulations is NP-hard [2]. Still, it would be interesting to design a method to compute such a circulation effectively using the powerful solvers available nowadays.

Message sequence graphs are a well-known formalism to describe communication protocols by means of partial orders of events called message sequence charts [9, 10]. As discussed in [1], this model can be regarded as a special case of VASSs when the latter are provided with a partial-order semantics. In this way, new features can be stirred into message sequence graphs such as message loss, message duplication, counters or timers. For that reason we found it useful to develop a prototype that implements the model-checking and the reachability techniques from [1]. In the future our verification tool should benefit from the description of structural bugs by flowers presented in this paper. The model of VASS is also similar to communicating finite-state machines (CFM). However the latter adopt a FIFO restriction on the ordering of messages along executions [3]. Many properties decidable for VASS are undecidable for CFM due to this restriction, in particular structural termination [20]. Thus cycles and circulations are not representative of structural bugs in this context and our results have no chance to apply to this model.

Acknowledgements

We would like to thank the anonymous reviewer who detected a mistake in an earlier version of [2] and whose observations helped us to correct Algorithm 2. Thanks also to the reviewers of this extended paper for their suggestions that guided us to improve the presentation.

References

[1] F. Avellaneda and R. Morin. Checking partial-order properties of vector addition systems with states. In Josep Carmona, Mihai T. Lazarescu, and Marta Pietkiewicz-Koutny, editors, *13th International Conference on Application of Concurrency to System Design, ACSD 2013, Barcelona, Spain, 8-10 July, 2013*, pages 100–109. IEEE, 2013.

[2] F. Avellaneda and R. Morin. Exhibition of a structural bug with wings. In Gianfranco Ciardo and Ekkart Kindler, editors, *Application and Theory of Petri Nets and Concurrency - 35th International Conference, PETRI NETS 2014, Tunis, Tunisia, June 23-27, 2014. Proceedings*, volume 8489 of *Lecture Notes in Computer Science*, pages 253–272. Springer, 2014.

[3] D. Brand and P. Zafiropulo. On communicating finite-state machines. *J. ACM*, 30(2):323–342, 1983.

[4] H. Carstensen. Decidability questions for fairness in Petri nets. In Franz-Josef Brandenburg, Guy Vidal-Naquet, and Martin Wirsing, editors, *STACS*, volume 247 of *Lecture Notes in Computer Science*, pages 396–407. Springer, 1987.

[5] E. M. Clarke, O. Grumberg, K. L. McMillan, and X. Zhao. Efficient generation of counterexamples and witnesses in symbolic model checking. In *Proceedings of the 32Nd Annual ACM/IEEE Design Automation Conference*, DAC '95, pages 427–432, New York, NY, USA, 1995. ACM.

[6] E. Cohen and N. Megiddo. Strongly polynomial-time and NC algorithms for detecting cycles in dynamic graphs (preliminary version). In David S. Johnson, editor, *STOC*, pages 523–534. ACM, 1989.

[7] R. Diestel. *Graph Theory*. Springer-Verlag, Heidelberg, 2010.

[8] J. Esparza and M. Nielsen. Decidability issues for Petri nets - a survey. *Bulletin of the EATCS*, 52:244–262, 1994.

[9] B. Genest, D. Kuske, and A. Muscholl. A Kleene theorem and model checking algorithms for existentially bounded communicating automata. *Inf. Comput.*, 204(6):920–956, 2006.

[10] J. G. Henriksen, M. Mukund, K. Narayan Kumar, M. A. Sohoni, and P. S. Thiagarajan. A theory of regular MSC languages. *Information and Computation*, 202(1):1–38, 2005.

[11] J.E. Hopcroft and J-J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8:135–159, 1979.

[12] K. Iwano and K. Steiglitz. Testing for cycles in infinite graphs with periodic structure (extended abstract). In Alfred V. Aho, editor, *STOC*, pages 46–55. ACM, 1987.

[13] R.M. Karp and R.E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969.

[14] D. Kőnig. *Theorie der endlichen und unendlichen Graphen*. Leipzig: Akademische Verlagsgesellschaft, 1936.

[15] S. R. Kosaraju and G. F. Sullivan. Detecting cycles in dynamic graphs in polynomial time (preliminary version). In Janos Simon, editor, *STOC*, pages 398–406. ACM, 1988.

[16] O. Kupferman and S. Sheinvald-Faragy. Finding shortest witnesses to the nonemptiness of automata on infinite words. In Christel Baier and Holger Hermanns, editors, *CONCUR*, volume 4137 of *Lecture Notes in Computer Science*, pages 492–508. Springer, 2006.

[17] S. Leue, R. Mayr, and W. Wei. A scalable incomplete test for the boundedness of UML RT models. In Kurt Jensen and Andreas Podelski, editors, *TACAS*, volume 2988 of *Lecture Notes in Computer Science*, pages 327–341. Springer, 2004.

[18] S. Leue and W. Wei. Integer linear programming-based property checking for asynchronous reactive systems. *IEEE Trans. Software Eng.*, 39(2):216–236, 2013.

[19] R.J. Lipton. The reachability problem requires exponential space. Technical Report 63, Yale University, 1976.

[20] R. Mayr. Undecidable problems in unreliable computations. *Theor. Comput. Sci.*, 297(1-3):337–354, 2003.

[21] G. Memmi and G. Roucairol. Linear algebra in net theory. In Wilfried Brauer, editor, *Advanced Course: Net Theory and Applications*, volume 84 of *Lecture Notes in Computer Science*, pages 213–223. Springer, 1980.

[22] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[23] J. Sifakis. Structural properties of Petri nets. In Józef Winkowski, editor, *MFCS*, volume 64 of *Lecture Notes in Computer Science*, pages 474–483. Springer, 1978.

[24] D.D. Sleator. Data structures and terminating Petri nets. In Imre Simon, editor, *LATIN*, volume 583 of *Lecture Notes in Computer Science*, pages 488–497. Springer, 1992.

[25] A. Trivedi. Interesting title for Vardi's 1986 paper. `http://blogs.warwick.ac.uk/atrivedi/entry/interesting_title_for`, 2005.

[26] M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *LICS*, pages 332–344. IEEE Computer Society, 1986.