Synchronized Dancing of Oblivious Chameleons

Shantanu Das[1], Paola Flocchini[2], Giuseppe Prencipe[3], and Nicola Santoro[4]

[1] LIF, Aix-Marseille University and CNRS, France
shantanu.das@lif.univ-mrs.fr
[2] EECS, University of Ottawa, Canada
flocchin@site.uottawa.ca
[3] Dipartimento di Informatica, Università di Pisa, Italy
prencipe@di.unipi.it
[4] School of Computer Science, Carleton University, Canada
santoro@scs.carleton.ca

Abstract. It has been recently discovered that oblivious iguanid lizards can form a periodic sequence of *tableaux vivants* with some restrictions on the tableaux. By viewing each tableau as a dance step, a formable sequence can be seen as a dance choreography, performable by the lizards. Interestingly, a complete characterization exists of the dances performable by all families of oblivious iguanid lizards except for the family of chameleons. This gap in knowledge opens the main research question addressed here: what choreographies can be danced by oblivious chameleons? We provide a full answer to this question, investigating formable tableaux, danceable choreographies as well as number of skin colours. We show that, unlike other lizards, in their feasible dances chameleons can touch and repeat steps. Also, they can do this even if they are asynchronous.

1 Introduction

Although the incidence of obliviousness among lizards of the suborder Iguania[1] is still matter of speculation, the body of studies and investigations on what these oblivious lizards can and cannot do continues to grow. Since oblivious lizards can decide, based on the observed environment, whether to stay still or to move to a specific location, it is possible (at least in principle) for a group of lizards of the same family to arrange themselves into a *tableau vivant* where all members of the group stay still. Since forming a specific tableau depends both on the behaviour of the lizards and on their initial location, the research quest of behaviouralist engineers has been to determine what behaviours (if any) would allow a group of oblivious lizards to create a given tableau, to characterize which tableaux can be formed from a given initial location, and to identify which tableaux cannot be formed regardless of the behaviour (e.g., see [1, 2, 5–9, 11–13]).

Several factors and conditions have been found to impact on the feasibility of a certain tableau to be formed (e.g., whether the lizards are affected by narcolepsy,

[1] Included in this suborder are iguanas, agamid lizards (such as the bearded dragon), anoles, and chameleons.

A. Ferro, F. Luccio, and P. Widmayer (Eds.): FUN 2014, LNCS 8496, pp. 113–124, 2014.
© Springer International Publishing Switzerland 2014

myopia, etc.). Foremost, whether or not a tableau can be formed depends on the relationship between the symmetry of the initial position of the lizards and the *symmetricity* of the tableau. Another crucial factor is the level of *synchronization* of the group of lizards: Full synchrony of the group allows all members to act simultaneously, while semi-synchrony allows only those awake to act simultaneously. Clearly any reduction in the level of synchrony, up to the complete absence of any synchrony (asynchrony), reduces the possibilities of behavioural engineering and thus the tableaux vivants that can be formed.

A recent result has opened new doors to the researchers of oblivious lizards of the suborder Iguania, especially to those artistically inclined. It has been shown [4] that, in spite of their obliviousness, a semi-synchronous group of those lizards can actually form not just a single tableau but a repeating *sequence* of distinct tableaux, pausing after each one! Since the tableaux must be formed in the order specified by the sequence, this means that some form of collective memory is possible in spite of the individual obliviousness. Clearly, not every singly formable tableau can be included in a formable sequence; in fact the study is on determining which repeating sequences of tableaux can be indeed formed. The artistic excitement generated by the announcement of this result is due to the fact that a sequence of tableaux vivants can be seen as the choreography of a *dance* where the tableaux are the dance steps. Hence the research question is: *What choreographies can be danced by oblivious lizards?*

This question has been recently answered [4]: any sequence of tableaux can be choreographed provided that: (R1) no tableaux is repeated in the sequence, (R2) in each tableaux the lizards never touch each other, and (R3) the symmetricity of every tableaux is the same as that of the starting configuration. This completely and fully characterizes all the choreographies achievable by all the families of the suborder Iguania with the *exception* of the family Chamaeleonidae.

In fact, in spite of the sharing of special features with members of different families of this suborder[2], *chameleons* have specialized cells, chromatophores, which contain pigments in their cytoplasm, allowing them to change their skin coloration and pattern (e.g., [10]). By taking advantage of this unique feature, chameleons are capable of forming colourful tableaux and dances, a feat that other iguanid lizards are incapable of. But, in addition to adding colours to a tableaux, are oblivious chameleons capable of forming different tableaux and, thus, performing more complex and sophisticated dances? In other words, by considering tableaux and dances irrespective of the chameleons' colours, the open research question is: *What choreographies can be danced by oblivious chameleons?*

In this paper we fully answer this question by completely characterizing the sequences of tableaux that can be formed by semi-synchronous oblivious chameleons irrespective of their colours. We prove that oblivious chameleons can form any sequence of tableaux even if (i) some tableaux are repeated within the sequence, (ii) the lizards may touch each other, and (iii) all tableaux do not necessarily have the same symmetricity, provided that the symmetricity of

[2] E.g., the teeth of both agamids and chameleons are borne on the outer rim of their mouths rather than on the inner side of their jaws, a feature unusual among lizards.

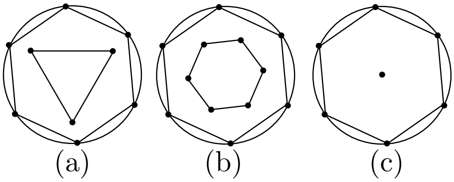

Fig. 1. (a) A pattern with symmetricity 3. (b) A pattern with symmetricity 6. (c) A pattern with symmetricity 1.

each tableaux is divisible by that of the starting configuration. In other words, the formable sequence of tableaux for chameleons do not suffer from restrictions (R1) and (R2) of the other iguanid lizards, and the restriction (R3) is substantially weaker. We first consider sequences when chameleons can touch but no tableau is repeated, then those when chameleons cannot touch but tableaux can be repeated, and finally the class of arbitrary sequences. For each class we determine the minimum number of colours needed to perform a feasible dance, and present a simple algorithm that allows such a number to suffice (in order of magnitude). In view of the recent result of [3] showing that asynchronous oblivious chameleons with a small constant number of colours can simulate any behaviour of any semi-synchronous oblivious iguanid lizards, it follows that all our results hold also in the fully asynchronous model. Due to space limitations, proofs are sketched or omitted.

2 Model and Definitions

Model: Let $V = \{v_1, \ldots, v_N\}$ be a set of points on a two dimensional plane, and let $size(V)$ be the cardinality of V. The smallest circle enclosing the points in V, denoted by $SEC(V)$, is the circle of minimum diameter such that every point of V is either on or in the interior of this circle. The point set V is said to be symmetric if V can be decomposed into a set of concentric circles centred in the centre c of $SEC(V)$, each containing a set of regular q-gons for some $q > 1$, divisor of N. The largest q for which this is true is called *symmetricity* of V and denoted by $q(V)$. The set of points in each regular $q(V)$-gon centred in c, is called a symmetricity *class*; the number of classes is denoted by $\alpha(V)$. If V is not symmetric then we define $q(V) = 1$. Note that, by definition, if the centre of $SEC(V)$ is an element of V, then $q(V) = 1$ (see Figure 1).

Given a set of distinct *colours* $C = \{c_1, \ldots, c_k\}$, we define a *colouring* as a function $\lambda : V \rightarrow C$. We say that λ is *proper* when $\lambda(x) = \lambda(y)$ iff x and y belong to the same symmetricity class in V. The *chromatic symmetricity* of V with respect to λ is the largest q for which V can be partitioned into a set of concentric regular q-gons where corners of a q-gon share the same color; $\beta(V, \lambda)$ denotes the number of chromatic symmetricity classes of V with respect to λ. We extend the chromatic symmetricity definition to multi-sets of points (e.g., multiple chameleons can be colocated). Let `GOLD` and `OFF` be two special colours.

Let $R = \{r_1, \ldots, r_n\}$ be a set of *chameleons* on the plane, each modelled as a computational unit provided with its own local memory and capable of performing local computations, and viewed as a point in $\mathbb{R}^2$. We assume that the chameleons start from distinct points in the plane and the colour of their skin is OFF, but during the course of the algorithm multiple chameleons may occupy the same point in $\mathbb{R}^2$. A chameleon coloured OFF will also be referred to as "uncoloured". Each chameleon has its own local coordinate system; the local coordinate systems of the chameleons might not be consistent with each other, but they all have the same *chirality* (e.g., a clockwise orientation of the plane). A chameleon is endowed with sensorial capabilities and it observes the world by activating its sensors, which return a snapshot of the positions of the other chameleons in its local coordinate system. The chameleons are *identical*; they execute the same protocol; they are *autonomous* (there is no central control); they are *silent* (they have no means of sonic communication to other chameleons). The skin of a chameleon can assume different colours (from the finite set C). The chameleons are *oblivious* (they do not have *persistent* memory of the past).

Each chameleon can freely move in the plane. At any point in time, a chameleon is either *active* or *inactive.* When *active*, a chameleon executes a *Look*-*Compute*-*Move* (LCM) cycle. In *Look*, a chameleon observes the world obtaining the snapshot of the positions of all chameleons with respect to its own coordinate system (since chameleons are viewed as points, it gets the set of their coordinates). In *Compute*, the chameleon executes its algorithm, using the snapshot as input. The result of the computation is a destination point. In *Move*, the chameleon moves to the destination (always reaching it); if the destination is the current location, the chameleon stays still. When *inactive*, a chameleon is idle. All chameleons are initially inactive. The amount of time to complete a cycle is assumed to be finite, and the *Look* is assumed to be instantaneous.

As mentioned before, each chameleon can colour its skin; the colour is visible to all the chameleons when they perform their *Look* and can be updated by the chameleon during the *Compute* operation. The colour is persistent; i.e., while the chameleons are oblivious forgetting all other information from previous cycles, their colours are not automatically turned off at the end of a cycle.

With respect to the activation schedule of the chameleons and their LCM cycle, we distinguish the fully-synchronous (FSYNC), the semi-synchronous (SSYNC), and the asynchronous (ASYNC) *models.* In ASYNC, the chameleons are activated independently, and the duration of each *Compute*, *Move* and inactivity is finite but unpredictable. As a result, the chameleons do not have a common notion of time, chameleons can be seen while moving, and computations can be made based on obsolete observations. On the opposite side of the spectrum, in FSYNC, the activations of all chameleons can be logically divided into global rounds; in each round, the chameleons are all activated, obtain the same snapshot, compute and perform their move. Note that this is computationally equivalent to a fully synchronized system in which all chameleons are activated simultaneously and all operations are instantaneous. The SSYNC model is like the fully-synchronous model where however not all chameleons are necessarily activated in each round.

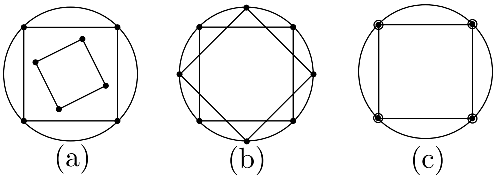

Fig. 2. (a) A pattern consisting of two classes. (b) and (c) show the two possible kinds of contractions: in (b), the two classes are contracted into just one, where all chameleons occupy distinct positions; in (c) the two classes are contracted into just one, where points of multiplicity two are created (the circled dots).

In any case, the activation scheduler is assumed to be fair. We assume the SSYNC model in this paper and show how to extend the results to the ASYNC model.

Notations: We will describe the global positions of the chameleons using a fixed coordinate system Z, unknown to the chameleons: $r_i(t)$ denotes position of r_i at time t, and $d(r_i(t), r_j(t))$ the Euclidean distance between $r_i(t)$ and $r_j(t)$. The *configuration* of the n chameleons on the plane at time t is denoted by the multi-set $\Gamma(t) = \{(r_i(t), \lambda(r_i(t))), 1 \leq i \leq n\}$ where $\lambda(r_i(t))$ is the colour of chameleon r_i at time t. Given a configuration $\Gamma(t)$, we denote by $L(\Gamma(t))$ the set of distinct points occupied by the chameleons in the configuration $\Gamma(t)$, and by $size(\Gamma(t))$ the cardinality of $L(\Gamma(t))$. We define $q(\Gamma(t)) = q(L(\Gamma(t)))$ and $\alpha(\Gamma(t)) = \alpha(L(\Gamma(t)))$. When no ambiguity arises, we will omit t. Note that $\alpha(V) = n/q(V)$, if there are n points in V.

A *tableau* or *pattern* P is a set of distinct points. A pattern P_i is said to be *isomorphic* to a pattern P_j, denoted $P_i \equiv P_j$, if P_j can be obtained by a combination of translation, rotation and uniform scaling of pattern P_i. Two patterns that are not isomorphic to each other are said to be *distinct*. We will denote the size of a pattern P_i by $size(P_i)$. We say that the chameleons have *formed* the pattern P at time t if $L(\Gamma(t)) \equiv P$.

Let $\mathcal{S} =< S_0, \ldots, S_{m-1} >$ be an ordered *sequence* of patterns with $S_i \not\equiv S_{i+1}$, called *choreography*. We define $\alpha(\mathcal{S}) = \max_i\{\alpha(S_i)\}$. Given $P \in \mathcal{S}$, we denote by $\mu(P)$ the number of occurrences of P in $\mathcal{S}$, and $\mu(\mathcal{S}) = \max_i\{\mu(S_i)\}$. We say that $\mathcal{S}$ has *repetitions* if $\mu(\mathcal{S}) > 1$, and that $\mathcal{S}$ has *contractions* if there is a pattern P in $\mathcal{S}$ such that $\alpha(P) < \alpha(\mathcal{S})$ (see Fig. 2). A set of chameleons executing an algorithm $\mathcal{A}$ starting from a configuration $\Gamma(t_0)$ is said to *form* $\mathcal{S}$ if, during any possible execution of $\mathcal{A}$ from $\Gamma(t_0)$, there exist times $t_1, \ldots t_m$, where, $\forall 0 < j < m$, $t_0 < t_j < t_{j+1}$ and $L(\Gamma(t_j)) \equiv S_j$. A set of chameleons executing $\mathcal{A}$, starting from a configuration $\Gamma(t_0)$ *performs* the dance described by the choreography $\mathcal{S}$, if they repeatedly form $\mathcal{S}$, i.e. if they form $\mathcal{S}^\infty = \langle S_0, \ldots, S_{m-1}\rangle^\infty$.

3 Fundamental Limitations and Techniques

3.1 Limits

To establish the artistic limits of oblivious chameleons, we first show the following:

Lemma 1. *If the initial configuration $\mathit{\Gamma}_0$ has symmetricity $q = q(\mathit{\Gamma}_0)$ then, for any algorithm, an adversary can ensure that any subsequent configuration $\mathit{\Gamma}'$ has symmetricity $q(\mathit{\Gamma}') = a \cdot q$, for some integer $a \geq 1$.*

Proof. The adversary can decide the coordinate system of each chameleon and also the activation schedule. First, observe that, if there is a chameleon at the center of $SEC(\mathit{\Gamma}_0)$, then $q = 1$ by definition, and thus the lemma holds trivially. Assume now that that there are no chameleons in the center of $SEC(\mathit{\Gamma}_0)$. Then the adversary can define the coordinate system of each chameleon r_i as follows: the origin is at the location of r_i, and the point of coordinates $(1, 0)$ is at the center of $SEC(\mathit{\Gamma}_0)$. If the adversary activates all chameleons together in each round, the chameleons in the same class would always occupy the corners of a regular q-gon and the symmetricity would be a multiple of q.

Theorem 1. *A set of n oblivious chameleons starting from initial configuration $\mathit{\Gamma}(t_0)$, regardless of the number of available colours,* cannot *perform the dance $\mathcal{S}^\infty = \langle S_0, \ldots, S_{m-1}\rangle^\infty$ if any of the following holds, where $q_0 = q(\mathit{\Gamma}(t_0))$ and $n_0 = size(\mathit{\Gamma}(t_0))$: (1) $q(S_i)$ is a not multiple of q_0, for some $S_i \in \mathcal{S}$, (2) $size(S_i) > n_0$, for some $S_i \in \mathcal{S}$.*

Proof. Part (1) follows from Lemma 1 and from the definition of formed pattern. For part (2) note that, if two chameleons are co-located in $\mathit{\Gamma}(t_0)$ and they have the same coordinate system, then they will choose the same point as the next destination (and the same colour, if they change colour). Thus, if chameleons that are co-located and have the same colour are always activated together, the number of distinct points in a configuration can never increase, and (2) holds.

A pattern is *feasible* from initial configuration $\mathit{\Gamma}(t_0)$ if none of the two forbidden conditions stated in the previous theorem hold; furthermore, a choreography is *feasible* if it is composed only of feasible patterns. In the following, we will only consider feasible choreographies and patterns, and when no ambiguity arises, we shall omit the term *feasible.*

3.2 Techniques

It is straightforward that the chameleons can agree on a total ordering of the classes in any $S_i \in \mathcal{S}$. Also, since they agree on chirality, it follows that in a given $\mathit{\Gamma}(t)$ the chameleons can agree on a total ordering of the classes in $L(\mathit{\Gamma}(t))$.

Lemma 2. *In any configuration $\mathit{\Gamma}$, the chameleons can elect a leader class among the $\alpha(\mathit{\Gamma})$ classes.*

In addition to this observation, we will make use of four techniques.

Identification. The first technique that we will use in the following is based on an idea introduced in [4], and is used to identify the pattern of the choreography that the chameleons are currently forming.

Given a choreography $\mathcal{S}$, each element $S_i \in \mathcal{S}$, $1 \leq i \leq m$, is mapped to a real number $F(S_i)$ using an appropriate injective function[3] $F : \mathcal{S} \mapsto \mathbb{R}$. This mapping is employed to allow the oblivious chameleons to distinguish which pattern of the sequence they are currently forming. More precisely: A special class of chameleons, the *leaders* (denoted by the set R_l) and whose identification will be detailed in the following sections, move to create a special configuration, named $\textsc{Ratio}(F(S_i))$, such that the circle $Q = SEC(\Gamma \setminus R_l)$ (i.e., the smallest circle enclosing all non-leaders) has a radius that is $1/F(S_i)$ times the distance of any leader chameleon to the center of this circle Q (refer to Figure 3). Since function $F()$ is injective, once $\textsc{Ratio}(F(S_i))$ has been created, all chameleons can uniquely agree on the pattern S_i that is being currently formed.

Expansion. The expansion process starts when we want to bring the chameleons from a configuration Γ (colored or uncolored) to an uncoloured configuration Γ' such that $\beta(\Gamma, \lambda) = \alpha(\Gamma')$ and the number of concentric circles is precisely $\alpha(\Gamma')$.

Let $Cir_1, \ldots, Cir_\alpha$ be the concentric circles populated by chameleons in Γ. Starting from the inside to the outside, for each circle Cir_i that contains multiple (chromatic) classes, we expand Cir_i by moving one class at a time, in an ordered fashion, to a slightly bigger circle, until all classes on Cir_i have been separated on different circles, each containing a single class. We now uncolor the chameleons on these circles, and expand Cir_{i+1}. This process will be denoted by $\textsc{Expansion}(\Gamma)$.

Lemma 3. *Let Γ be a coloured configuration, with coloring function λ, that has $\beta(\Gamma, \lambda)$ coloured classes.* $\textsc{Expansion}(\Gamma)$ *creates an uncoloured configuration Γ' with $\alpha(\Gamma') = \beta(\Gamma, \lambda)$.*

Contraction. Let the chameleons start from an uncoloured configuration Γ with $\alpha(\Gamma) = \alpha$ classes, each class located on a different circle. Let S_i be any pattern in $\mathcal{S}$. If $\alpha(S_i) < \alpha$, then we can activate the contraction process, as described below. Let $Cir_1, \ldots, Cir_\alpha$ be the concentric circles in Γ populated by chameleons, and $\delta = \lfloor \alpha / \alpha(S_i) \rfloor$. Contraction is achieved by collapsing consecutive groups of δ circles, from the outside to the inside, until there are only $\alpha(S_i)$ circles populated by chameleons. For the smallest circle C_i containing a single class, all chameleon on circles from Cir_{i+1} to $Cir_{\delta+i}$ rotate, one group at the time, so that there are no co-radial chameleons; Now, again one group at the time, these chameleons collapse on Cir_i. We iterate this process until we obtain a new configuration Γ' that has exactly $\alpha(S_i)$ populated circles. In the following, we will denote this process by $\textsc{Contraction}(\Gamma, S_i)$, with $\alpha(\Gamma) = \alpha$ and $\alpha(S_i) < \alpha$. We define as *density* of the contraction the maximum number of classes on the same circle at the end of this process, and we denote it by $\psi(\Gamma, S_i)$.

[3] Note that $F(S_i) \neq F(S_j)$ whenever $i \neq j$, even if S_i and S_j are isomorphic.

Lemma 4. *Let Γ be an uncoloured configuration with $\alpha(\Gamma) = \alpha$ and let P be a pattern with $\alpha(P) < \alpha$.* Contraction(Γ, P) *creates a coloured configuration Γ' having exactly $\alpha(P)$ concentric circles populated by chameleons.*

Pattern Formation. Given any pattern S_i belonging to a feasible choreography, we can use a combination of expansion and contraction to obtain a *backbone* of S_i, which is defined as a configuration that contains exactly $\alpha(S_i)$ populated concentric circles and on each circle Cir_i, the number of chameleons is a multiple of the symmetricity of S_i. An *incomplete backbone* is a backbone with one circle missing (i.e. with only $\alpha(S_i) - 1$ circles).

Let Γ_i be either a complete or incomplete backbone of S_i, and let S_i be the pattern in $\mathcal{S}$ to be formed. In the first phase of the pattern formation process, for every two classes of S_i that are on the same circle, the corresponding circles of Γ_i are merged, after an appropriate rotation so that no chameleons collide. After this process, the number of populated circles in Γ_i is equal to the number of populated circles in S_i; the circles of Γ_i are moved so as to coincide with the circles of S_i. We can assume the populated circle of S_i with the smallest radius already coincides with the populated circle of Γ_i with the smallest radius. The next lemma follows from the total ordering of classes, colours, and populated circles of both Γ_i and Γ'.

Lemma 5. *Starting from a backbone Γ_i of a given pattern $S_i \in \mathcal{S}$, the chameleons can always reach a new configuration Γ', where (i) the radius of the i-th populated circle in Γ' equals the radius of the i-th populated circle of S_i; and (ii) the number of chameleons on the i-th populated circle on Γ' is a multiple of the number of points on the i-th circle on S_i.*

Let us call the Γ' of previous lemma the *skeleton* of S_i; again, we will say that the skeleton of S_i is complete or incomplete depending on whether the backbone of S_i was complete or incomplete. Once the chameleons have formed the skeleton of S_i, the second phase of the pattern formation process consists in the actual formation of S_i. For those circles of the skeleton that contain more chameleons than the corresponding circle of S_i, the chameleons on these circles are assigned different colours, one per class, using a routine AssigncoloursToClasses(S_i). Now, the positions of the chameleons having the smallest colour on Cir_1 determine the final positions to be occupied by all chameleons in order to successfully form S_i (this follows from chirality, and total ordering of the colours). Once, the final positions have been determined, the chameleons reach them, one class at a time, moving within each circle ordered according to the colouring (multiplicities can be formed if required). At this point, all chameleon but the leaders have reached their final positions; let us call this configuration *almost final.*

The final step is to have the leaders to reach their final positions. Notice that, if the skeleton was complete, then the configuration without the leaders already forms S_i; thus, the leaders will just occupy positions occupied by another class on the outermost populated circle. Otherwise, the skeleton of S_i is missing one class in order to complete S_i; thus, the leaders will occupy these missing final

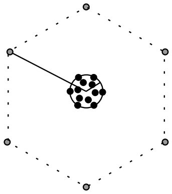

Fig. 3. The configuration Ratio(r), where the gray dots represent the leader chameleons. The small circle is Q (refer to Section 3.2).

positions, ending the pattern formation process. We will denote this process by PatternFormation(Γ_i, S_i), where Γ_i is a backbone configuration of S_i.

Lemma 6. *Starting from a skeleton Γ'_i of a given pattern $S_i \in \mathcal{S}$, the chameleons can always form S_i.*

4 Contraction-Free Choreographies (with Repetitions)

In this section, we consider sequences of patterns where a pattern may appear more than once in the sequence; however, all patterns have the same number of classes; i.e., $\forall i, j, \alpha(S_i) = \alpha(S_j) = \alpha$. We first provide a lower bound on the number of colours, k necessary to perform a choreography in this setting.

Theorem 2. *Given a contraction-free choreography $\mathcal{S}$ with α classes, the choreography described by $\mathcal{S}$ can be performed* only if *the number of available colors $k \geq \mu(P_i)^{\frac{1}{\alpha}}$, $\forall P_i \in \mathcal{S}$, where $\mu(P_i)$ is the number of occurrences of P_i in $\mathcal{S}$.*

Proof. Due to the oblivious nature of the chameleons, it is necessary to distinguish between different occurrences of the same pattern. This means that, since chameleons in the same equivalence class behave in the same way, and a pattern P_i has α classes, using exactly l colours it is possible to distinguish l^α repetitions. Hence, with less than $\mu(P_i)^{\frac{1}{\alpha}}$ distinct colours it is impossible to assign $\mu(P_i)$ different proper colourings for P_i so to distinguish the different occurrences.

We now show an algorithm that can form all feasible contraction-free sequences using almost the minimum number of colours. The protocol is outlined in Figure 4, where each step is assumed to be completed before the next one takes place. The chameleons start from an arbitrary initial uncoloured configuration I with *size*$(I) = n$. The $\alpha(I) = \alpha$ different classes in I can be lexicographically ordered by the chameleons; the first step of the algorithm is to have the chameleons in the smallest class to become *leaders*. The leaders colour themselves `GOLD`, and during the entire algorithm they will always maintain this special colour.

The leaders place themselves in the appropriate position so to create a Ratio$(F(S_i))$ configuration, as described in Section 3.2. When Ratio$(F(S_i))$

has been completed, the chameleons know that S_i is the next element of the sequence to be formed. At this point, they perform an expansion to obtain a backbone of S_i. The chameleons then invoke AssigncoloursToPatterns($\mathcal{S}$), to determine which new colour each should take next. This subroutine assigns colours to the points of each S_i so that each repetition in $\mathcal{S}$ of the same pattern P become distinct, with the condition that in each colouring, the elements of the same class are given the same colour. This assignment can be easily done because colours and classes are totally ordered. When this process is completed, PatternFormation() can be invoked, and S_i is formed.

```
Protocol RepeatedCoreography
/* $S_i$ is the next pattern to be formed, $1 \leq i \leq m$ */
  1. If no chameleons have GOLD colour, elect the leaders $R_l$, and colour them GOLD
  2. Create a Ratio($F(S_i)$) configuration $\Gamma$
  3. Expansion($\Gamma \setminus R_l$) creating configuration $\overline{\Gamma}$
  4. AssigncoloursToPatterns($\mathcal{S}$)
  5. PatternFormation($\overline{\Gamma}'$, $S_i$)
```

Fig. 4. The protocol to execute contraction-free choreographs

Theorem 3. *Any contraction-free choreography $\mathcal{S}$ can be performed by a group of chameleons with $k = \mu(\mathcal{S})^{\frac{1}{\alpha(\mathcal{S})-1}} + 1$ colours.*

Proof. By Lemma 2, a class can be unambiguously selected as leaders. According to the RepeatedCoreography protocol, the leaders get coloured GOLD that will never change during the dance: this class acts as coordinator for the entire algorithm. To start the formation of a pattern, the leaders create a Ratio($F(S_i)$) configuration which, by construction, unambiguously identifies what is the next pattern to be performed. Since each P consists of α classes, and GOLD is reserved only for the leaders, $\mu(P)$ repetitions of $P \in \mathcal{S}$ can be distinguished using $\mu(P)^{\frac{1}{\alpha-1}} + 1$ colours for the classes. Since distinct patterns in $\mathcal{S}$ can be coloured independently of each other, for the entire process $\mu(\mathcal{S})^{\frac{1}{\alpha-1}} + 1$ colours suffice.

5 Repetition-Free Choreographies (with Contractions)

In this section, we handle the case when the patterns in $\mathcal{S}$ might not have the same number of equivalent classes (i.e. there could be contractions); however there are no patterns that appear more than once in $\mathcal{S}$.

We assume that the initial configuration I is such that $size(I) = n$, and that $\forall i, n \geq size(S_i)$. We first give a lower bound on the number k of colours necessary for the chameleons to be able to perform a choreography in this setting.

Theorem 4. *Given a repetition-free choreography $\mathcal{S}$, the dance described by $\mathcal{S}$ can be performed* only if *the number of available colors $k \geq \frac{maxS(\mathcal{S})}{minS(\mathcal{S})}$, where $maxS(\mathcal{S}) = \max\{size(P_i)\}$, and $minS(\mathcal{S}) = \min\{size(P_i)\}$.*

Proof. Let $S_{\max}$ be a pattern in $\mathcal{S}$ such that $\mathit{size}(S_{\max}) = \mathit{maxS}(\mathcal{S})$, and let $S_{\min}$ be a pattern in $\mathcal{S}$ such that $\mathit{size}(S_{min}) = \mathit{minS}(\mathcal{S})$. By contradiction, let us assume k is smaller than the bound in the theorem and there is an algorithm that performs $\mathcal{S}$ using k colors. In particular, the algorithm correctly forms $S_{\min}$. Since $n \geq \mathit{maxS}(\mathcal{S}) > k \cdot \mathit{minS}(\mathcal{S})$, it follows from the pigeon-hole principle that there are points occupied by more than k chameleons (which cannot all have distinct colors). Any two chameleons that are colocated and have the same color may not be separated by a deterministic algorithm. Thus, in any subsequent configuration, the chameleons may occupy at most $k \cdot \mathit{minS}(\mathcal{S}) < \mathit{maxS}(\mathcal{S})$. This implies that $S_{\max}$ may not be formed and the theorem follows.

We now prove that we can form all feasible repetition-free sequences by using almost the minimum number of colours (see Figure 5). In contrast with the previous case, the colours are not necessary to distinguish among patterns in the sequence, but among chameleons from different classes that happen to contract to points of multiplicity, and thus need to break the contraction at a later time.

```
Protocol ContractedCoreography
/* $S_i$ is the next pattern to be formed, $1 \leq i \leq m$ */
  If no chameleons have GOLD colour, elect the leaders $R_l$, and colour them GOLD
  Create a Ratio($F(S_i)$) configuration $\Gamma$
  Call Expansion($\Gamma \setminus R_l$) creating configuration $\overline{\Gamma}$
  Call Contraction($\overline{\Gamma} \setminus R_l, S_i$) creating a backbone $\Gamma'$
  PatternFormation($\Gamma', S_i$)
```

Fig. 5. The protocol to execute repetition-free choreographs

The chameleons start by electing the leaders, colouring them GOLD, and having them form configuration Ratio($F(S_i)$). Then, the chameleons start an expansion process that brings them into a configuration with α equivalence classes each in a different circle (Section 3.2) and with the colours of all chameleons except the leaders being OFF. When all non-leaders are OFF, they perform a contraction to create the backbone of S_i, and the PatternFormation() starts.

Theorem 5. *Any non-repeating choreography $\mathcal{S}$ can be performed by chameleons with $\frac{maxS(\mathcal{S})}{minS(\mathcal{S})} + 1$ colours.*

6 Arbitrary Choreographies

We now have all the necessary tools to solve the most general case when $\mathcal{S}$ can contain both repetitions and contractions. The algorithm has the same structure of ContractedCoreography. The only crucial difference is in the function used in AssigncoloursToClasses(). While in ContractedCoreography colours were assigned only to the contracting classes, the function now determines how to assign colours, from a minimal set, to classes when forming a specific occurrence of P in $\mathcal{S}$ so as to distinguish different repetitions as well as contracting classes. The optimal number of colours $k^*(\mathcal{S})$, although difficult to express in a closed formula, is easily computable and easily bounded:

$$1 + \mu(\mathcal{S})^{\frac{1}{\alpha(\mathcal{S})-1}} \left\lceil \frac{maxS(\mathcal{S})}{minS(\mathcal{S})} \right\rceil \geq \quad k^*(\mathcal{S}) \quad \geq Max \left\{ \mu(\mathcal{S})^{\frac{1}{\alpha(\mathcal{S})}}, \left\lceil \frac{maxS(\mathcal{S})}{minS(\mathcal{S})} \right\rceil \right\}$$

7 Asynchronous Chameleons

The results we have presented so far have been established for `SSYNC` chameleons. However, as recently shown in [3], any result for `SSYNC` iguanid lizards can be achieved by `ASYNC` chameleons with a constant number of colours. This means that the results of the previous section still hold in `ASYNC` with just an increase in the multiplicative constant of the number of colours. In particular:

Theorem 6. *Oblivious* `ASYNC` *chameleons with $O(k^*(\mathcal{S}))$ colours can perform any sequence of tableaux provided that the symmetricity of each tableau divides that of the starting configuration.*

References

1. Ando, H., Suzuki, I., Yamashita, M.: Formation and agreement problems for synchronous mobile robots with limited visibility. In: Proc. of the 1995 IEEE Symp. on Intelligent Control, pp. 453–460 (1995)
2. Chatzigiannakis, I., Markou, M., Nikoletseas, S.: Distributed circle formation for anonymous oblivious robots. In: Proc. of 3rd Workshop on Efficient and Experimental Algorithms, pp. 159–174 (2004)
3. Das, S., Flocchini, P., Prencipe, G., Santoro, N., Yamashita, M.: The power of lights: Synchronizing asynchronous robots using visible bits. In: Proc. of 32nd ICDCS, pp. 506–515 (2012)
4. Das, S., Flocchini, P., Santoro, N., Yamashita, M.: On the computational power of oblivious robots: forming a series of geometric patterns. In: Proc. of 29th PODC, pp. 267–276 (2010)
5. Défago, X., Souissi, S.: Non-uniform circle formation algorithm for oblivious mobile robots with convergence toward uniformity. TCS 396(1-3), 97–112 (2008)
6. Dieudonné, Y., Labbani-Igbida, O., Petit, F.: Circle formation of weak mobile robots. ACM Trans. on Autonom. and Adapt. Sys. 3(4), 1–16 (2008)
7. Flocchini, P., Prencipe, G., Santoro, N.: Distributed Computing by Oblivious Mobile Robots. Morgan&Claypool (2012)
8. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Arbitrary pattern formation by asynchronous oblivious robots. TCS 407(1-3), 412–447 (2008)
9. Fujinaga, N., Yamauchi, Y., Kijima, S., Yamashita, M.: Asynchronous pattern formation by anonymous oblivious mobile robots. In: Aguilera, M.K. (ed.) DISC 2012. LNCS, vol. 7611, pp. 312–325. Springer, Heidelberg (2012)
10. Ligon, R.A., McGraw, K.J.: Chameleons communicate with complex colour changes during contests. Biology Letters 9(6) (2013)
11. Sugihara, K., Suzuki, I.: Distributed algorithms for formation of geometric patterns with many mobile robots. Journal of Robotics Systems 13, 127–139 (1996)
12. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots: Formation of geometric patterns. SIAM Journal on Computing 28(4), 1347–1363 (1999)
13. Yamashita, M., Suzuki, I.: Characterizing geometric patterns formable by oblivious anonymous mobile robots. TCS 411(26-28), 2433–2453 (2010)