

Lambdoid Emergence

René Vestergaard,^{*†} Jittisak Senachak,[†] and Mun'delanji Vestergaard[‡]

[†]School of Information Science, JAIST, Japan; [‡]School of Materials Science, JAIST, Japan.

February 14, 2008

We analyse lambdoid gene regulation using the *CEq formal method* for inferring emergent properties from *modal influence graphs*. In addition to presenting the most comprehensive mathematical modelling of *Bacteriophage lambda* to date, our approach allows us to articulate and explore rigourously stated and algebraically concise properties of the way the virus works, and to propose novel explanations of the nature of lambdoid switching, stability, anti-immunity, and more.

Keywords: CEq, algebraic biology, *Bacteriophage lambda*, emergence, gene regulation, systems biology.

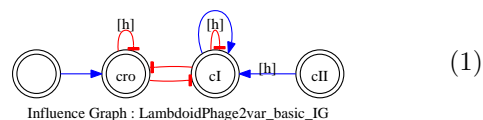
Proof assistants are formal methods, see Box 1, that help with formalising and verifying mathematical statements; they are mathematically rigid and build on long traditions in mathematical logic and theoretical computer science. The state-of-the-art is industrial grade and is supported by extensive meta-theory. Proof assistants do not teach mathematics but aim to increase trust and to facilitate the subsequent and often disruptive leveraging of the formalisation efforts. Success stories exist throughout theoretical computer science and even in pure mathematics [19, 26], and scientific mainstreaming is under way [20]. Formal proofs are composed of only elementary reasoning steps and are designed to be straightforward to check for correctness, irrespective of the manner and how hard they were to produce — this contrasts the situation with hand-written proofs.

^{*}Corresponding: René Vestergaard, JAIST, 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan; vester@jaist.ac.jp.

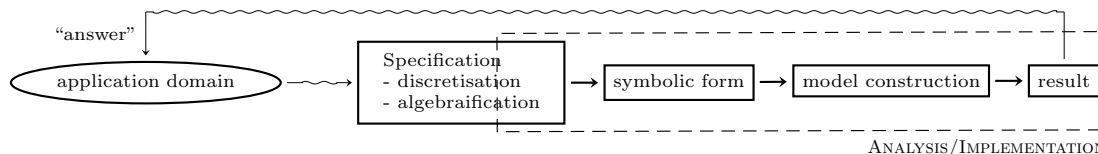
The CEq formal method [22] is similar to a proof assistant but is intended for biochemical applications, which makes things harder on three main fronts. One is the inescapable ‘formalist problem’: “have I formalised what I intended to formalise and how well does it fit reality?”. Another is the lack of an established foundation. Thirdly, there is the fact that mathematical properties are considered as inputs in the context of proof assistants. In biochemistry, reasoning often goes in the other direction by starting from basic facts and aiming for hypotheses that can be experimentally validated [5]. Loosely speaking, this means that reasoning and proving are distinct undertakings in biochemistry whereas they are degrees of the same thing in mathematics. One concept that can help bridge the two worlds of formal reasoning and biological science is that of *emergence*, which we claim that CEq’s eponymous *cascaded equilibria* and the attendant *CEq Emergence Assistant* address.

One particularly revealing challenge for a proof assistant is to slavishly apply it to a well-written textbook that can be expected to be correct. This article is a self-contained account of our efforts to do that with CEq and Ptashne [21]. The presentation uses CEq’s inherent algebraic style; in places we go beyond [21] and certain peripheral issues are omitted.

Bacteriophage lambda infects *Escherichia coli*.



A formal method consists of a mathematically well-defined analysis, a supporting tool implementation, and technologies and methodologies for successfully applying it to (real-world) problems and for relating its formal results back to the application domain. A discrete formal method typically takes its input in an algebraic specification language, i.e., a language consisting of objects and operations over them; it may be general purpose or domain-specific. The operations will have an intended (possibly user-specifiable) semantics, and a symbolic intermediate language is typically defined in which the semantics can be made explicit. A model construction is then employed to allow for either interactive or automated execution of the core part of the analysis that produces the final result.



Key: \rightsquigarrow is human activity; \rightarrow is (possibly user-directed) machine activity.

Box 1. Typical layout of a (discrete) ‘formal method’.

First discovered more than fifty years ago, the virus is a key model organism for the study of gene regulation because its two distinct ways of growth and an externally-triggered switch between them appear to be coded for in just two genes. The full story is only slightly more complex, and most of the phage’s central functionality comes from the gene interactions in (1), with the (blue) pointed arrow expressing activation and the (red) flat arrow repression. *Bacteriophage lambda* has also been actively used as a vector in cloning work and as the subject of gene manipulation and replacement, partly with the aim of understanding how the phage may have evolved the ability to switch [10, 4, 2, 3]. It was therefore surprising when it was recently discovered that lambdoid gene regulation employs more advanced machinery than previously thought necessary, including cooperativity over a distance of several thousand base pairs, and that this seemingly plays an important role [8]. More recently, our knowledge of lambdoid gene regulation has been further deepened [9, 18, 17, 3, 7].

Reading the literature, it is clear that lambdoid gene regulation is thought to be algebraic in nature, i.e., to be guided by inherent attributes of the involved entities. While many things need to be understood and complete formalisations of course are infeasible [1], (1) additionally suggests that lambdoid

gene regulation also is believed to be modular, i.e., that the *cro* and *cI* genes have a clean contact surface with their environment. CEq is a proposal for a formalisation of the referenced style of algebraic argumentation. Formalisation is important because it guarantees reproducibility and consistency across analyses, in particular as regulatory arguments become subtler in the face of richer and deeper understandings, and it becomes difficult to correctly and consistently take all relevant issues into account at the right times. This implies that formal methods typically address the systems perspective. More, formalised understandings allow for iteratively improved analyses, thought experiments [3], systematic explorations of combinatorial possibilities [11], and more.

Methods

We describe how CEq works and how we apply it to lambdoid gene regulation. The basic principles behind CEq were introduced earlier [23], although with limited flexibility. The current presentation marks a substantial maturation across the board, and introduces several notions to the specification language and analysis that go beyond the graph form of influence graphs. The aim has been to parameterise the CEq model construction where technical choices are made and to make CEq a viable option for working biochemists. The presentation of CEq we give is

not complete and covers only the fragment that is required for the analysis of lambdaoid gene regulation; everything we discuss has stabilised.

The CEq Formal Method closely follows Box 1 in its construction, with tool support provided by the CEq Emergence Assistant [22]. (We use CEq as a generic name for both the formal method and the tool). Having been designed in part as a biochemical formal method, all aspects of CEq have direct biochemical interpretations, as we shall see.

Input to CEq is made using the associated MIG (Modal Influence Graph) specification language that formalises and extends the graphical notion of *gene-regulation networks*, aka influence graphs. MIG is a textual language to easily accommodate the algebraic information that MIG adds to plain influence graphs, with the CEq tool having the ability to extract a graphical influence graph from a well-formed specification, see (1). A MIG specification consists of two main kinds of declarations, each occupying a line of text; one kind of declaration is for objects.

```
;Objs | States | Active | Abs | Attr
"g1" | 0,"1","h" | "1","h" | |
```

(A line with an initial ‘;’ is for comments and is not formally parsed.) A vertical bar is used to separate different types of information, with object declarations consisting of five types. The first is the name of the object given as a string of letters. The next three are lists of either integers or strings of letters that indicate i) the regulatable states of the object, ii) the default sub-set of i) where the object is able to influence other objects, aka its ‘active’ states, and iii) a disjoint set of non-regulatable or ‘absent’ states (we shall not use this facility here). Finally, there is room to specify that an object has particular attributes; we shall discuss some of these later. Experience has taught us the following best modelling practices.

- Let all objects represent, e.g., genes. In particular, avoid mixing abstraction levels by declaring both, e.g., genes and proteins as objects.
- Use states to account for the ways influences may be exerted or received: with genes as objects,

states should represent the distinct ways the expressed proteins may bind to the DNA, which typically means that states formalise the “steps” of a given gene’s regulatory response function.

A second kind of MIG declaration is for influences.

```
;Inf^cer | Inf^cee|A/R| Inhbtr | Mdlr
"g2" {"h"} | "g1" | + | "g0.1" |
```

This particular declaration says that g2 in its h-state activates g1, as indicated by ‘+’, and that g0 inhibits the activation in its l-state. Repression would be indicated by ‘-’. Without the information in {}, CEq would have used each of g2’s active states as an influencing state. A similar facility is available for influencees, where writing “g1” is short-hand for “g1” [0, “1”, “h”] and where the content within [] can be any sequence of states of the considered object in any order and with omissions and duplicates. The above example is thus equivalent to the following.

```
"g2" {"h"} | "g1" ["h","1",0] |-| "g0.1" |
```

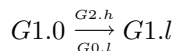
Lastly, we may specify modalities that qualify the meaning of the influence in question, as we shall see.

The intended semantics of influences is given explicit symbolic form in CEq’s ARS (Auto-Regulating Systems) intermediate formalism. Given a MIG specification, we consider the set of objects, \mathcal{O} , and for each o in \mathcal{O} the associated set of states, \mathcal{S}_o , and we denote by *entities*, or \mathcal{E} , the set of objects dotted with their states, e.g., {G0.l, G1.0, G1.l, G1.h, G2.h}. By default, CEq capitalises the initial letter of an entity when constructing \mathcal{E} , similar to the gene-protein distinction. This is done to stress that CEq analysis conceptually will lead to a result that belongs to a level of abstraction above the one in the MIG specification. Formally, an ARS relation over an \mathcal{E} consists of 4-ary relationships, called *reactions*, between subsets of \mathcal{E} . The four components are referred to as *substrates*, *products*, *catalysts*, and *inhibitors*.¹

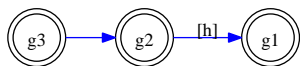
$$Ss \xrightarrow[Is]{Cs} Ps$$

¹When all reactions in a given ARS specification obey stoichiometric laws, the ARS will represent chemical reactions but the ARS formalism is intended for more general use and CEq is not, in fact, aware of any stoichiometry.

The MIG to ARS translation has the above influence give rise to the following reaction, and more.



Next, the Ceq model construction is used to primitivise the mechanisms by which influences work. The most basic way that feedback may change is for one object to change state, for example by G1.0 going to G1.l, and on to G1.h. Because the considered change can be effected by G2.h, g2 may influence anything that g1 influences. If some G3.l is able to bring about G2.h then also g3 will be able to influence anything that g1 influences, and so on. Ceq analysis implements these mechanisms by turning the ARS relation into a graph, called the *cascading model* (CM), where certain nodes contain both the substrate and the catalyst of a reaction. The general rule is that a CM will contain a combined substrate, catalyst node if the catalyst occurs as a product of another reaction, with one exception described below. First, we note that the example has the following influence graph.



For a given influence graph, Ceq can produce a number of different outcomes. The modalities that MIG add to influence graphs are used to make this (important!) choice, with the possible outcomes for the considered example looking as follows.²

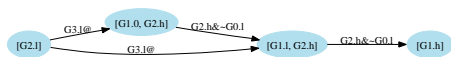
Residual, Approximative:



Residual, Definitive:

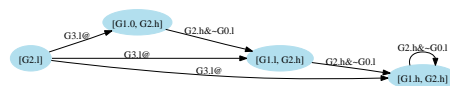


Pointwise, Approximative:



²The graphs are actual Ceq outputs.

Pointwise, Definitive:



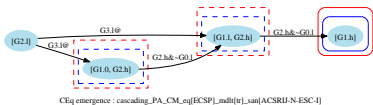
The outcomes differ in how cascading is implemented. The first two collapse the cascader only onto the first of several reactions that are catalysed by it, with the trailing reactions starting from nodes containing just the substrate (this is the exception mentioned above). The discussed two outcomes result from the MIG modality called ‘Residual’, with the alternative called ‘Pointwise’.³ The second and fourth models include the possibility that the catalyst actively keeps the influenced object in the extreme state. These two outcomes result from the MIG modality called ‘Definitive’, with the alternative called ‘Approximative’.⁴ After the creation of the outlined nodes, which we refer to as *points-of-interaction*, Ceq reconsiders all the reactions and creates nodes containing any product that has not already been put in a (targetable) node, such as G1.h in the two ‘approximative’ examples. In practice, this means that ‘approximative’ influences will be approximative in their ability to bring about the extreme state of the influencee. When finally inserting the reactions as edges between the appropriate pairs of constructed nodes, inhibitors will either prevent the insertion if there is a clash with the contents of the nodes, or they will remain in place and be indicated with a ~-prefix on the edge. The catalysts also remain in place and are indicated either as is or with a ‘@’-suffix if they do not occur as a product.

A CM is a comprehensive account of the ways information may propagate. Alternatively, we can see the graphs as indicating the degree of sustainability of DNA-binding configurations/concentration profiles. The general assumption that we will then make is that more highly sustainable profiles are more likely to underpin physiology, and the final analysis step therefore looks for sustainable situations. A node with out-degree zero amounts to a (locally) sustainable situation that no catalyst can facilitate a

³MIG synonyms include ‘Transitive’/‘nonTransitive’.

⁴MIG synonyms include ‘Reflexive’/‘irReflexive’.

change from. More generally, we will be interested in clusters of interconnected nodes that have no edges leaving them as a group. Graph-theoretically, maximal intraconnected clusters of nodes are referred to as *strongly connected components* (SCCs). Given a directed graph, $\langle \mathcal{V}, \rightarrow_e \rangle$, where $\rightarrow_e \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges between the graph’s vertices, we write $v_1 \rightarrow_e^* v_2$ if vertex v_1 can reach vertex v_2 in zero, one, or more \rightarrow_e -steps. A vertex, v , belongs to exactly one SCC, namely $[v] \triangleq \{v' \mid v \rightarrow_e^* v' \wedge v' \rightarrow_e^* v\}$. SCCs induce an equivalence relation over V , which we factor out for the purposes of sustainability identification in CMs: the *shrunk graph* of a given $\langle \mathcal{V}, \rightarrow_e \rangle$ is the (acyclic) graph between SCCs with edges given as $V_1 \curvearrowright_e V_2 \triangleq \exists v_1 \in V_1, v_2 \in V_2. v_1 \rightarrow_e v_2$. In particular, CEq identifies and indicates SCCs with out-degree zero in different shrunk graphs derived from the arrived-at CM as the final analysis step.



Let $\langle _ \rangle$ be the operation that removes all edges with inhibitors from a CM. Given a CM, M , CEq specifically identifies clusters of nodes that have out-degree zero in $\langle [M] \rangle$ and $[\langle M \rangle]$; we refer to these as *pre-equilibria* (thick red boxes) and *sub-equilibria* (thin blue boxes). The former includes clusters with out-degree zero in $[M]$, which we refer to as *equilibria* (proper). If a pre- or sub-equilibrium has an out-edge in M , we say that it is *collapsible* (dashed boxes). If all incoming edges to one of them carry inhibitors, we say that it is *preventable* (rounded corners). Informally, we are interested in collapsible pre- or sub-equilibria because they can be actively sustained by the presence of the inhibitors on the out-edges, and are atomic because of their intraconnectedness. Technically speaking, we have that pre-equilibria always contain at least one sub-equilibrium and that sub-equilibria never contain another sub- or pre-equilibrium. More generally, any hybrid pre-/sub-equilibrium that could be obtained by removing some inhibited edges before the $[_]$ operation will contain at least one sub-equilibrium. In other words, pre- and sub-equilibria will account

for all equilibrium behaviour, with no need to explore the combinatorially many hybrid forms of equilibria.

Although we do not elaborate on the mathematics here, CMs obey a *concurrent semantics* that permits different sub-graphs to co-exist in time, provided they are not contradictory. Co-existability in CMs accounts for *stochasticity* and more, as we shall see.

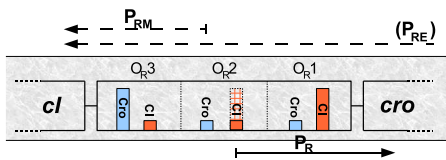
Lastly, we note that in particular auto-influences may not lead to what would be considered meaningful ARS reactions. An example is the following situation where we assume that the G1s refer to one individual. (We call this the ‘monolithicity’ assumption.)



CEq can automatically sanitise away these and similar cases as part of the analysis. The following sanitation steps are possible, and are used by default in the order listed,⁵ first for the MIG to ARS translation.

- A** No reaction may have an **absent** state as catalyst.
 - C** No reaction may have an inhibitor as **catalyst**.
 - S** No reaction may have an inhibitor as **substrate**.
 - R** No reaction may use an object at two different states as catalyst and substrate (example).
 - I** An inhibitor occurring with a different state in the substrate of a reaction is suppressed.
 - J** An inhibitor occurring with a different state in the catalyst of a reaction is suppressed.
- The possible ARS to CM sanitation is as follows.
- N** No **node** may contain an object in two states.
 - C** No edge may involve an object **changing** states that is not prescribed by the reaction.
 - S** No edge may **start** in a node where the catalyst occurs with a different state as a non-substrate.

⁵The annotated letters are indicated in CEq’s output in the order listed, with big-case meaning that the sanitation step is performed and small-case meaning that it is not.



Key: histograms show binding and affinities; arrows show transcription, with natural/mediated affinities solid/dashed.

The *cI* and *cro* genes are adjacent on the λ -chromosome. Between them are two *promoter* sites where RNA polymerase, RNAP, can attach and initiate transcription of separate DNA-strands, i.e., in opposite directions: P_{RM} for *cI* and P_R for *cro*; RNAP has natural affinity for P_R but not for P_{RM} . Superimposed on the promoters are three *operator* sites to which CI and Cro protein may bind in their default dimer form: O_{R3} , O_{R2} , O_{R1} ; O_{R3} is inside P_{RM} , O_{R1} is inside P_R , and O_{R2} is overlapping the two promoters in different ways. CI's binding is special in that the dimer has "arms" that reach around and

make contact on the other side of the DNA. CI has strong affinity for O_{R1} and weak for the others, while Cro has strong affinity for O_{R3} and weak for the others. With a dimer bound to either O_{R2} or O_{R1} , P_R is blocked and *cro* is not transcribed; similarly for O_{R3} , P_{RM} , and *cI*. CI dimer bound to O_{R1} will typically mediate the binding of a second CI dimer to O_{R2} by tetramerisation. The CI-tetramer i) will mediate RNAP binding to P_{RM} , resulting in *cI*-transcription, and ii) will typically octamerise with a CI-tetramer similarly bound to two operator sites a few thousand base pairs away on the λ -chromosome, which, in turn, iii) mediates tetramerised CI binding between O_{R3} and a third site at the remote operator when these are active at high concentration. Cro does not cooperate beyond dimerisation. High-concentration CII protein mediates *cI*-transcription from a downstream promoter site, P_{RE} , with the transcribing RNAP having to traverse the O_R region before reaching *cI*. RNAP from P_{RE} is blocked by (tetramerised) CI but not by Cro. We are not considering regulation of *cII*.

Box 2. Chemistry of *Bacteriophage lambda*'s *cI*, *cro* chromosome region, following Ptashne [21].

E No edge may end in a node where the catalyst occurs with a different state as a non-product.

I An inhibitor occurring with a different state in the non-substrate part of a start node is suppressed.

As mentioned above, a final mandatory constraint is that no edge may connect to start or end nodes that contain a non-suppressed inhibitor of the edge.

CEq instantiates Box 1 by formalising and/or introducing the discussed technologies as follows.

Specification: influence, attribute, modality, MIG.

Symbolic form: reaction, sanitation, ARS.

Model construction: cascading, sanitation, CM.

Result: pre/sub-equilibria, collapsible, preventable.

The complete formal definition of CEq, including the syntax of the MIG specification language, is available online [24].

Our basic MIG specification of lambdoid gene regulation is based on the account of the chemistry of the relevant region of the *Bacteriophage lambda* chromosome given in Box 2. (We use the term 'lambdoid' to indicate that, while we start from physical reality, we are not bound by any ex-silico constraints in making analytic changes to the basic specification, i.e., to stress that we are doing formal analysis.)

The CI and Cro proteins will each have two distinct binding configurations on λ -DNA, with strong-affinity binding significant already at 'low' concentrations and weak-affinity binding significant only at 'high' concentrations. The object declarations above the first dashed line in Figure 3 formalise these, and include the possibility of the proteins being at zero concentration, e.g., prior to infection. We also consider CII protein, for which we distinguish a 'high' state from the rest, as well as DNA-transcribing RNA polymerase. Protein-DNA interactions are generally weak and last a relatively short time when compared to the time it takes to effect functional changes in protein concentrations. We therefore specify 'Pointwise' as the default influencing modality, i.e., we specify

Pointwise, Approximative				
;Objs	States	Active	Absent	Attr
"cro"	0, "l", "h"	"l", "h"		
"cI"	0, "l", "h"	"l", "h"		resid_Icer
"cII"	"", "h"	"h"		
" "	"RNAP"	"RNAP"		

;Inf^cer	Inf^cee	A/R	Inhbtr	Mdlt
"cro" {"h"}	"cro"	-		
"cI"	"cro"	-		dfntv
" "	"cro"	+	"cI", "cro.h"	

"cI" {"h"}	"cI"	-		
"cro"	"cI"	-		
"cI"	"cI"	+	"cro", "cI.h"	
"cII" {"h"}	"cI"	+	"cI"	

;Seeds				
<"cI.0", "cro.0">				

Figure 3. Basic lambdoid phage gene regulation specified as a Modal Influence Graph (MIG)

that an influencer explicitly must be seen to have remained active in order to be able to effect a second state change.⁶ Exceptionally, we assert that the special circumstances surrounding CI’s DNA binding makes *cI* a ‘Residual’ influencer.⁶ Informally, this means that a particular *cI* influence will (appear to) remain in place for an extended period of time. Formally, all influences by an object with the ‘resid_Icer’ attribute will automatically be qualified with the ‘residual’ modality. In terms of what the influences can accomplish, we specify that they are ‘Approximative’ at extreme states by default because, with one or two exceptions, all influences will be counter-acted in ways that, e.g., prevent the proteins from returning to zero concentration or from having their concentration increase arbitrarily.⁶

The influences are specified between the dashed lines in Figure 3, first for influences on *cro* and then on *cI*. The influences are justified by direct reference to Box 2. The first influence reflects that Cro binds to O_{R1} at high concentrations, even if it only has weak affinity for the site, thereby preventing *cro*-transcription and, by natural decay, resulting in lower Cro-concentration. We note that the specified line is equivalent to the following under CEQ-sanitation.

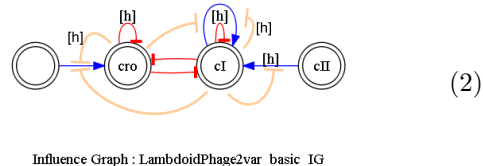
⁶We shall explore this issue extensively.

"cro" {"h"} | "cro" ["l", "h"] | - | |

We prefer the version in Figure 3 because it is operator-site driven in the style of Box 2 and does not second-guess what state changes may take place. We can confirm that the influence is approximative, as it is an auto-influence that naturally will cease as Cro transitions away from ‘high’ concentration. At any non-zero concentrations of CI, i.e., in *cI*’s active states, *cI* similarly represses *cro*, as formalised in the second line. Exceptionally, the special nature of CI’s DNA-binding combined with the relative instability of Cro means that the considered influence is ‘Definitive’, i.e., can make Cro reach (functionally) zero concentration.⁶ The third line accounts for the natural affinity of RNAP for *cro*-transcription; the transcription can not proceed in the presence of CI (at any concentration) or Cro at high concentration because of the O_{R1}/P_R overlap. In particular, we see that the auto-inhibition at high concentration justifies the ‘Approximative’ modality. The remaining influences are on *cI* and are similarly justified. In case of CII-mediated *cI*-transcription, only CI is an inhibitor via O_R -binding [21], possibly because of CI’s ‘arms’ that reach around the DNA, thereby conceivably blocking RNAP’s interstrand movement.⁶ The specification of *cI*’s auto-activation is equivalent to the following due to CEQ sanitation.

"cI" {"l"} | "cI" | + | "cro" |

We, again, prefer Figure 3’s operator-site driven style that corresponds more closely to Box 2. In particular, we note that repression in gene regulation typically will be a secondary effect of an inhibition that is efficient enough for the natural decay and dispersal of proteins to become dominant forces. This is the case for all four specified lambdoid repressions, as becomes clear if we let CEQ insert also inhibitions when extracting the influence graph from Figure 3.⁷



⁷The graph is from CEQ but has been edited for legibility.

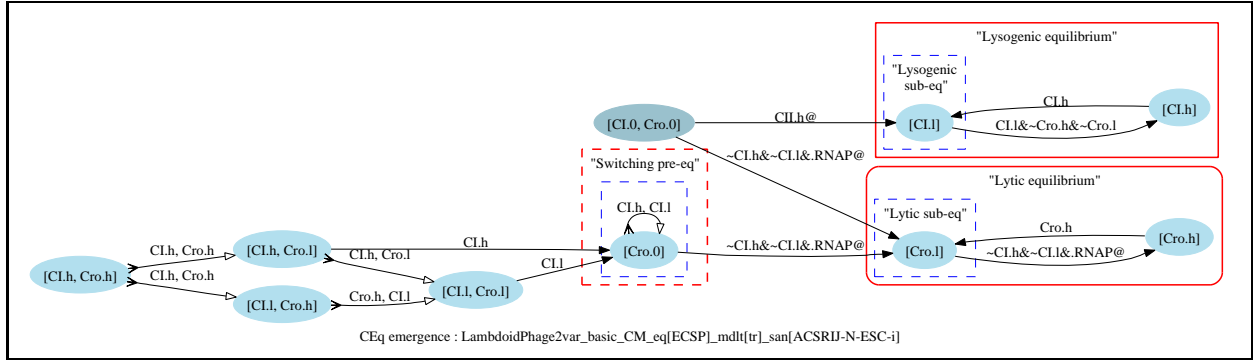


Figure 4. Ceq output for basic lambdoid phage gene regulation, see Figure 3 (with named equilibria)

Concretely, we are stating that short of (mediation of) proteolysis, protein-protein binding, or similar, it is not appropriate to include a repression in a gene-regulation specification without an attendant (possibly implicitly specified) inhibition.

Lastly, we specify at the end of Figure 3 that we wish our modelling to include the situation with both proteins at zero concentration, i.e., we consider the phage prior to and at the point of infection. We do this explicitly because, without proteins, the considered situation is not an (intrinsic) point-of-interaction. Differently said, we add the node to account for the barren state of an obligate parasite.

Results

Figure 4 displays the Ceq-analysis result of our basic lambdoid MIG specification, see Figure 3. We have named the pre-/sub-equilibria for ease of reference. The slightly darker colour of the $CI.0, Cro.0$ node indicates that the node would not have been inserted if it had not been seeded in the specification.

The lysogenic cycle is thought to be indefinitely maintainable in *Bacteriophage lambda*. The cycle is characterised by CI 's concentration fluctuating within some band as the bacterium host goes through its life-cycle and replicates the infecting virus along with itself. No other λ -genes are expressed during the lysogenic cycle, and the cycle is evidently related

to the “lysogenic equilibrium” in the figure. We see that the equilibrium makes it clear that no other protein is needed for CI to be able to actively adjust cI -expression to maintain its own concentration between the low and high extremes. Conversely, the presence of the lysogenic sub-equilibrium seemingly poses a problem for in silico indefiniteness but a closer look at the “switching pre-equilibrium” of Figure 4 illustrates why CI is also called ‘repressor’.

CEq Assertion 1 (Lysogenic Cycle) *Lambdoid cI will typically have the ability to i) maintain steady CI concentration and ii) actively keep off cro-expression; when i) and ii) co-exist, the resulting lysogenic cycle can not be interrupted per se.*

Item i) refers to what we call the lysogenic equilibrium in the figure and ii) refers to the switching pre-equilibrium. The assertion raises the question of how the lysogenic cycle is entered into and suggests that the cycle will be affected by the ‘collapsibility’ of the switching pre-equilibrium. We shall explore these issues in Ceq Assertions 3, 4, and 5.

The lytic attack of *Bacteriophage lambda* is swift, typically taking less than one hour before roughly a hundred virus progeny are released by lysis of the host. The attack is controlled in the two initial stages by cro . First, the concentration of Cro increases, triggering rapid DNA-replication among other things. In the second stage, the concentration of Cro decreases

again, to allow for coating of the phage progeny from the first stage ahead of their release. The lytic attack is captured in the figure by the “lytic equilibrium”, with the rounded corners indicating ‘preventability’, i.e., that all incoming edges can be inhibited.

CEq Assertion 2 (Lytic Attack) *cro will involuntarily commit a typical lambdoid phage to a lytic attack when and only when CI is absent for long enough to let Cro reach (sufficiently) high concentration.*

The assertion amounts to a literal reading of the lytic equilibrium and its surrounds in Figure 4. The increase in Cro’s concentration is effected by RNAP’s natural affinity for *cro*-transcription, which any amount of CI can block. The nature and location of the lytic sub-equilibrium combined with the ‘preventability’ of the lytic equilibrium imply the ‘only when’ part of the assertion. For the ‘when’ part, we note that the lytic equilibrium and the two nodes feeding it cover all situations where CI is absent. The lytic attack is *not* stable in the indefinite sense of the lysogenic cycle, and it is physiologically important that only one increase/decrease cycle is required for the success of the (destructive) lytic attack [16].

Anti-immunity of a lambdoid phage is a state that is characterised by an inability to undertake either lytic or lysogenic growth. Anti-immunity is anticipated by the co-existable lytic and lysogenic sub-equilibria in Figure 4 but, *ex silico*, the state is common only in some lambdoid variants.

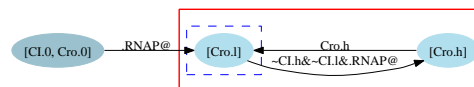
CEq Assertion 3 (Lysogenic Precedence) *Co-existing lytic and lysogenic equilibria will transition to the lysogenic cycle in a typical lambdoid phage.*

When the two equilibria co-exist, the pertinent dynamics is accounted for by the leftmost part of Figure 4. In particular, the lysogenic and lytic sub-equilibria will become co-existing and although they then appear to deadlock, the edge out of the $[CI.l, Cro.l]$ node in Figure 4 dictates that Cro’s concentration will be lowered, which will have the phage transition to the lysogenic equilibrium co-existing with the switching pre-equilibrium, i.e., to the lysogenic cycle. We will return with a wider-ranging analysis of these dynamics in CEq Assertion 6.

The initial decision that a lambdoid phage is faced with upon infection of a host is whether to pursue the lysogenic cycle or initiate the lytic attack at that point. By design, this issue is addressed in the $[CI.0, Cro.0]$, or *pre-infection*, node in Figure 4.

CEq Assertion 4 (Initial Decision) *A typical lambdoid phage will initiate its lytic attack upon infection unless i) the host is already a lysogen or ii) CII’s concentration becomes sufficiently high to allow the phage to enter the lysogenic cycle.*

We first note that CEq Assertion 2 establishes that the lytic attack is, indeed, the default response of a barren lambdoid phage. Next, we observe that the two edges out of the pre-infection node are not complementary. In case the phage is successful in pursuing both, CEq Assertion 3 shows that the lysogenic cycle will result, thus establishing the only non-obvious part of ii). In case of i), the host already contains CI and the *initial-decision lytic pathway* from the pre-infection node to the lytic equilibrium is blocked, due to inhibition. For arguing that this phenomenon is specific to secondary infections, i.e. that we are discussing λ -immunity of lysogens, we recall that CEq sanitises the generated CMs to ensure that analyses are performed in environments where no object is simultaneously in multiple states. The situation of a phage infecting a lysogen, i.e., a host with a pre-existing λ -infection, breaks the monolithicity assumption and the analysis in Figure 4 has been done using all but one of the default sanitation steps, as indicated by the non-capital ‘i’ at the end of the analysis label. As seen earlier, this sanitation step would have suppressed the CI inhibition on the edge out of the pre-infection node towards the lytic equilibrium. With default CEq sanitation, i.e., when analysing a monolithic host, the lytic cycle is identified as non-preventable, see the CM-excerpt below.⁸



⁸To simplify, we perform all remaining CEq-analyses under the monolithicity assumption, i.e., with default sanitation.

towards the lytic attack, CI concentration is reestablished to functional levels, the lytic and lysogenic sub-equilibria will become co-existing, i.e., the phage will transition to the proto-anti-immune state instead. This completes the justification of all edges in Figure 6. (There would be an additional edge from the lytic attack to proto-anti-immunity in case the lytic attack was not destructive after the first increase/decrease cycle.) What remains more generally is to understand in more detail the mechanisms that govern anti-immunity and switching.

Control of anti-immunity is localised in the factors that affect the $CI.l, Cro.l$ node, making the issue relatively straightforward to analyse with CEq.

CEq Assertion 6 *The ability of cI to repress cro is a key regulator of a lambdoid phage’s ability to commit to a mode of growth, with higher minimal Cro -concentration more likely to result in anti-immunity.*

The assertion is anticipated by the discussion following CEq Assertion 3, with formal evidence coming from counter-factually formalising cI ’s influence on cro as ‘Approximative’ rather than ‘Definitive’, see Figure 7. The specification change makes the switching pre-equilibrium disappear and makes anti-immunity an equilibrium in its own right.

Conversely, anti-immunity would not be a factor if the $CI.l, Cro.l$ situation could not arise. Formal evidence is given in Figure 8, where cro is specified to still inhibit cI -transcription but not sufficiently strongly to result in repression of CI’s concentration.

CEq Assertion 7 *The ability of cro to repress cI is a key regulator of a lambdoid phage’s ability to admit an anti-immune state, with less repression less likely to lead to (proto-)anti-immunity.*

A pathway analysis for Figure 8 leads to the same result as Figure 6 only without proto-anti-immunity, in part because the lytic sub-equilibrium now involves Cro at zero concentration and therefore is not co-existable with the lysogenic sub-equilibrium. The assertion may at first appear counter-intuitive. However, some amount of cro -repression of cI will have

no adverse effect in practice by CEq Assertion 3 and can thus either i) be used to balance the threshold of the switch by enabling safe fluctuation of genetic lysogeny, i.e., by making epigenetic switching less arduous, ii) be purely coincidental with no evolutionary pressure to have it eliminated, or iii) code for functionality we have not considered here.

Control of the switch is localised in the factors that affect the $Cro.0$ node, centring on the specifics of cI influencing. cI is special by i) being a ‘Residual’ influencer and ii) having a ‘Definitive’ influence on cro . An additional non-standard feature is that cI iii) inhibits its own cII -mediated activation by CI blocking an already initiated RNAP-traversal of the lambdoid chromosome, rather than occupying the relevant promoter site prior to the fact. It has recently become clear that CI’s long-range cooperativity makes cI inhibition and repression of cro a non-negligible factor and cooperativity more generally will, we surmise, play at least a key supporting role for i)–iii). More specifically, the formal notion of residual influence leads us to the following assertion.

CEq Assertion 8 *Any inertia associated with CI’s (cooperative) DNA-binding will be a key regulator of the efficiency of the lambdoid switch, with less inertia leading to less efficient switching.*

By inertia we mean reluctance to enter into or break out of the functional form of DNA-binding. Inertia is more likely to be an issue for CI because of the three involved forms of cooperativity: (DNA-independent) dimerisation, (short-range) tetramerisation, and (long-range and DNA-super-coiling) octamerisation. The idea behind the assertion is that more inertia would imply that CI concentration will be dysfunctionally low when the CI-binding that is responsible for a given repression ultimately fails, and Cro -led prophage induction will therefore be more likely to succeed. Experimentally, the assertion could be established by measuring repression as a function of CI concentration changing in lambdoid real-time, and observing a Δ depending on whether it is increasing or decreasing change. The Δ would capture CI’s joint reluctance to cease and initiate repression.

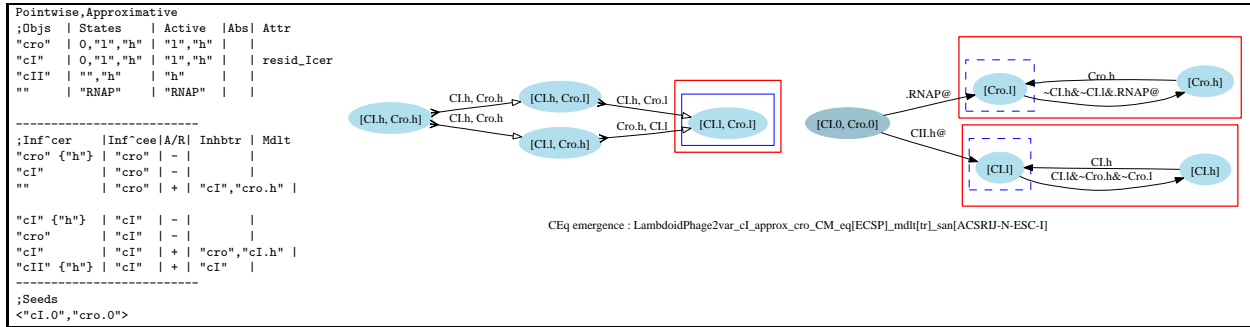


Figure 7. CEq input + output for lambdaoid phage with approxi-mative cI -influence on cro

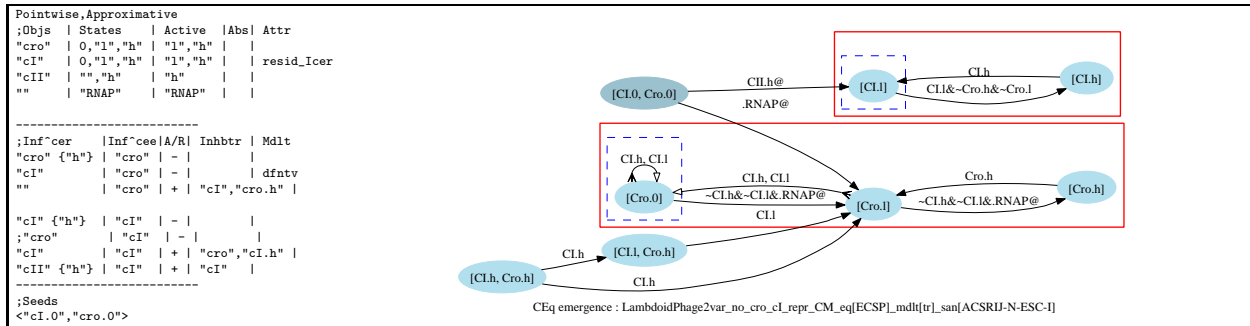
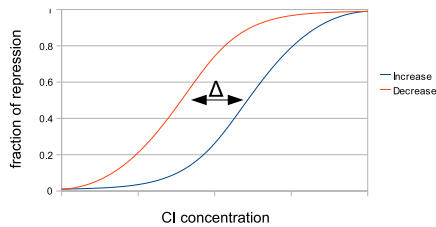


Figure 8. CEq input + output for lambdaoid phage without cro -repression of cI



We already saw evidence for the assertion in Figure 7 if we loosely relate inertia with the permanency of CI-mediated influences, i.e., the CEq-evidenced residuality and definitiveness of regulation by cI . Additional evidence comes from counter-factually formalising cI as a pointwise influencer, see Figure 9. The presented CM has Cro.0 paired up with functional CI in the locality where previously it had Cro.0 in a stand-alone capacity, with the result that the original switching

pre-equilibrium instead takes the form of a specialised lysogenic cycle, which is consistent with a small Δ .

CEq Assertion 9 *The specifics of cI -influencing, i)–iii), are sufficient and (without replacement) necessary to explain the lambdaoid switching asymmetry.*

Formal evidence for ‘sufficiency’ is supplied by having cro perform i), ii), and possibly iii) instead of cI , see Figure 10, with the result that switching becomes “lytic-to-lysogenic”.⁹ As for ‘necessity’, we have already considered the effects of separately relaxing CI’s ability to do i) and ii), see Figures 9 and 7. If they are relaxed at the same time, the result is the same as in Figure 7. If we relax also iii), we get the essentially non-physiological result in Figure S1.

⁹We consider this particular CEq argument to be especially elegant because it nicely show-cases CEq’s functionality and because the conclusion can not obviously be reached ex silico.

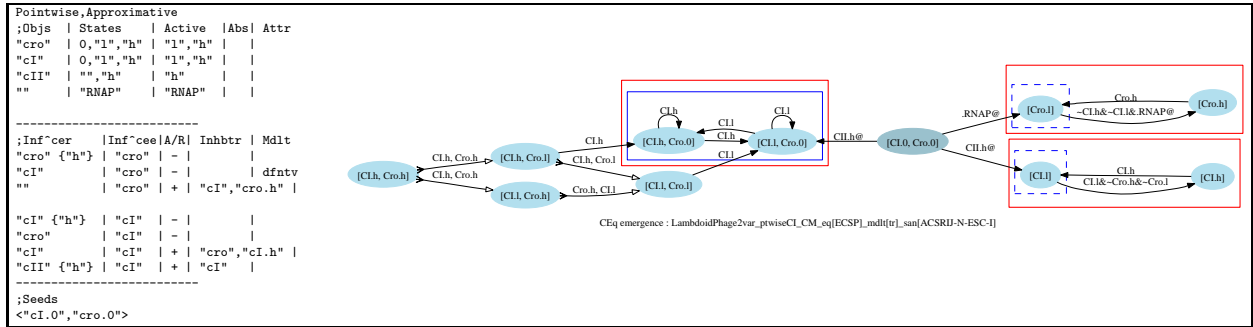


Figure 9. CEq input + output for lambdoid phage with pointwise cI -influence on cro

In either case, the result differs materially from standard functionality of a typical lambdoid phage.

Discussion

CEq has clear parallels with Kauffman/Thomas-style (KT) gene-regulation analysis [13, 25, 14] but was independently conceived, and the two analyses are technically and pragmatically different. The most perspicuous difference is in the nodes being considered: CEq uses an influence-driven set of nodes while KT analysis uses the state-space set of nodes, independently of what influences are being considered. As shown in Figure 11, CEq can emulate KT analysis by using the CEq approach to edge-sanitisation, edge-insertion, equilibrium-identification, etc., but applied to the state-space set of nodes. The analysis in Figure 11 includes *RecA*, i.e., uses the MIG specification in Figure 5, and consists of four disconnected subgraphs because, as specified, *recA* and *cII* are not regulated upon and thus do not change their state. The four subgraphs closely correspond to our CEq analyses and we expect general projections from state spaces to CMs may exist. The state-space graph does not prima facie address the issues we analysed by considering collapsibility and preventability of equilibria or the presence and absence of particular nodes, such as λ -immunity, switching, and anti-immunity. We shall return to this issue in more detail shortly.

Equally importantly, the KT state space is of exponential size in the number of objects being considered ($\prod_{o \in \mathcal{O}} |\mathcal{S}_o|$) whereas CMs at most

will be of quadratic size in the number of states ($(\sum_{o \in \mathcal{O}} |\mathcal{S}_o|)^2 = |\mathcal{E}|^2$), with experience showing that the size is often much smaller than this bound. Computations that exhibit exponential growth are often characterised as *infeasible* because there typically will be reasonably hard and reasonably low bounds on the size of problems that can be addressed in practice. KT-analysis, for example, requires $|\mathcal{S}_o|$ times more resources if we add object o with states \mathcal{S}_o to an existing specification, and the largest KT analysis we have heard reported involves less than 100 objects [15]. By contrast, CEq analyses a particular example containing around 2,000 transcription factors out of a total of 30,000 genes, with 70,000+ influences in 13 hours on a PC. The specification is of human-genome size and is semi-realistic in that it was arrived at by auto-interweaving an influence graph that was extracted from a data-set obtained from automated sequencing of gene-disrupted *Bacillus subtilis* [12]. (The analyses in this article take fractions of a second.)

It seems fair to suggest that the size issue has been a factor behind the K-function formalism used in KT analysis in place of influence graphs, because particular classes of functions can be KT-analysed efficiently (but still within the 100-object limit reported above). In CEq, where model size is not a limiting factor, we have instead focused on usability and transparency in setting up the MIG specification language: MIG is based on and extends the community-developed language of influence graphs that is already well-understood and has independently proved its worth

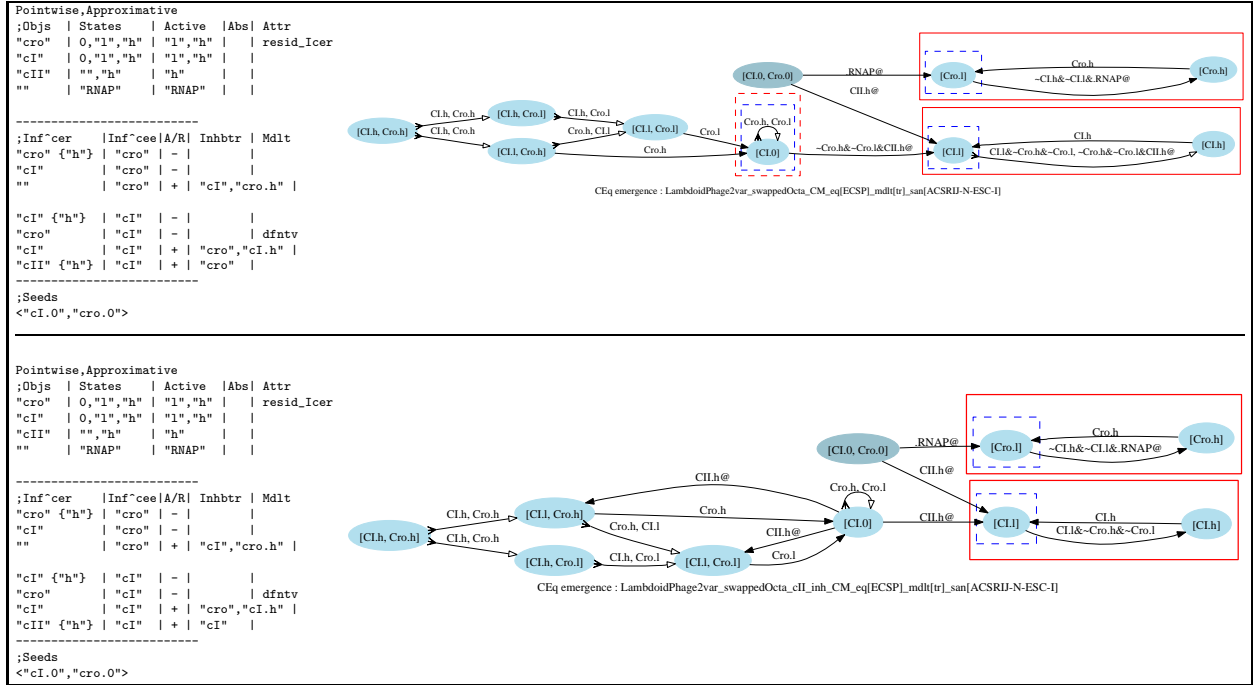


Figure 10. CEq input + output with swapped Cro, CI modalities and swapped/same inhibition of *cII*

to working biochemists. In terms of the expressivity of MIG vs K functions, we note that the latter can not faithfully accommodate situations where one object is subject to equally powerful positive and negative influences, and instead must prioritise one over the other. Example situations where prioritisation would be inappropriate include protein signalling: a single-phosphorylated MAPK protein, for example, is comparably (but not equally) likely to further phosphorylate and to dephosphorylate, with either asymmetry being pathological [23]. On a similar general note, it is not obvious how KT-analysis would deal with ARS-steps that do more than change the state of an object, as it by no means is obvious what states, e.g., B should be allowed to be in in start nodes and similar for A in end nodes in the following example.

$$A.a \xrightarrow{C.c} B.b$$

CEq, on the other hand, makes no assumption about what becomes what, and would simply introduce the

appropriate nodes to deal with the example.

The MIG modalities play a particularly central role in CEq and, for contrast, Figure S2 displays our basic lambdaoid specification analysed with the other possible default modalities, with each analysis significantly departing from key lambdaoid physiology. We believe the MIG modalities go some way towards answering open questions about the importance of modes of protein-DNA interaction [11].

CEq Assertion 10 *‘Pointwise’ and ‘Approximative’ concisely formalise the default modes of protein-DNA interaction in lambdaoid gene regulation.*

CEq Assertion 11 *‘Residual’ and ‘Definitive’ concisely formalise CI’s (special) DNA-interaction and the wider role it plays in lambdaoid gene regulation.*

Our notion of equilibrium proper coincides with the KT notion of limit cycle/steady state [6].

informal notion. We are currently formalising and automating it more generally but, as with all construction of formal methods, extensive and detailed efforts are required in terms of definitions and theory, and in compiling a library of suitable test cases.

CEq Assertion 15 *Co-existability of pre-/sub-equilibria is fundamental to understanding lambdaoid stability in CEq. The fact that the lysogenic equilibrium can co-exist with the cro-centred switching pre-equilibrium makes the lysogenic cycle stable in CEq. The absence of a cI-centred notion of pre- or sub-equilibrium that can co-exist with the lytic equilibrium and rule out the lytic sub-equilibrium makes the lytic attack not stable in CEq.*

Co-existability can be considered directly over CMs due to the inherently partial nature of their nodes. It is unclear how the lower-level notion of co-existing objects that is enforced in state spaces may impact on the higher-level co-existence notion we consider.

Conclusion

One question that may come to mind after our analysis of lambdaoid gene regulation is how the simple concept of cascading i) can lead to small specification changes resulting in sometime subtle, sometime substantial changes in the CEq-produced cascading models and ii) can appear to lead to meaningful results. We did, after all, address a lysogenic cycle, a lytic attack, the precedence between them, how the two initially are chosen between, how one may become the other, how neither may result, and how the switch seems to be dictated in large part by the permanency of a particular protein-DNA interaction. And, we did this starting from what amounts to (1).

The answer to i) is simple: non-monotonicity. From a MIG specification, CEq first constructs an ARS relation, and then first the nodes and then the (labelled) edges of a graph that is then analysed for sustainability. When looking more closely, we see that each of the steps from MIGs to (co-existable) equilibria have both negative and positive dependencies on the previous steps, which means that sometimes more leads to less and sometimes to more, and similarly for less: more *is* different, and so is less.

As for ii), we note that the philosophy behind CEq is to **trust the implied meaning of influence graphs** — they come from within the community and have independently proved their worth. Informally, CEq does this by parsimoniously deriving a formal method from them as prescribed by Box 1.

Methodologically, we can note that modularity is respected by cascading in the sense that the various CMs we consider contain just CI and Cro in their nodes although other genes also exert influences, i.e., CEq specifically targets emergent properties from objects that are influenced upon.

We are currently pursuing compositionality of modular specifications, more expressivity, e.g., with primitive support for coregulation, integration with automated sequencing analysis, and more.

Acknowledgements RV would like to dedicate this article with fondness and veneration to Dines Bjørner on the occasion of his retirement.

References

- [1] P.W. Anderson. More is different. *Science*, 177(4047), 1972.
- [2] S. Atsumi and J.W. Little. Regulatory circuit design and evolution using phage lambda. *Genes Dev.*, 18(17), 2004.
- [3] S. Atsumi and J.W. Little. Role of the lytic repressor in prophage induction of phage lambda as analyzed by a module-replacement approach. *Proc Natl Acad Sci USA*, 103(12), 2006.
- [4] Allan Campbell. The future of bacteriophage biology. *Nature Reviews Genetics*, 4, 2003.
- [5] Luca Cardelli. Can a systems biologist fix a Tamagotchi? Position Paper for the Gilles Kahn Colloquium, January 2007. Available at <http://lucacardelli.name/>.
- [6] C. Chettaoui, F. Delaplace, P. Lescanne, M. Vestergaard, and R. Vestergaard. Rewriting game theory as a foundation for state-based models of gene regulation. In *Proceedings of*

- the Fourth International Conference on Computational Methods in Systems Biology, Lecture Notes in Bioinformatics 4210*, 2006.
- [7] D.L. Court, A.B. Oppenheim, and S.L. Adhya. A new look at bacteriophage lambda genetic networks. *J Bacteriol.*, 189(2), 2007.
- [8] I.B. Dodd, A.J. Perkins, D. Tsemitsidis, and J.B. Egan. Octamerization of lambda CI repressor is needed for effective repression of P(RM) and efficient switching from lysogeny. *Genes Dev.*, 15(22):3013–3022, 2001.
- [9] I.B. Dodd, K.E. Shearwin, and J.B. Egan. Revisited gene regulation in phage lambda. *Curr Opin Genet Dev*, 15, 2005.
- [10] D.I. Friedman and D.L. Court. Bacteriophage lambda; alive and well and still doing its thing. *Current Opinion in Microbiology*, 4, 2001.
- [11] C.C. Guet, M.B. Elowitz, W. Hsing, and S. Leibler. Combinatorial synthesis of genetic networks. *Science*, 296(5572), 2002.
- [12] Kunihiko Hiraishi and Hirofumi Doi. Estimation of gene regulation using expression profiles by gene disruption and comprehensive sequence analysis on gene regulatory regions. Poster 1P-067, MBSJ Forum, 2006.
- [13] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467, 1969.
- [14] S. A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.
- [15] Reinhard Laubenbacher and et al. Discrete Visualizer of Dynamics. <http://dvd.vbi.vt.edu/>.
- [16] John W Little, Donald P Shepley, and David W Wert. Robustness of a gene regulatory circuit. *The EMBO Journal*, 18, 1999.
- [17] J.W. Little. Threshold effects in gene regulation: when some is not enough. *Proc Natl Acad Sci USA*, 102(15), 2005.
- [18] Christine B. Michalowski and John W. Little. Positive autoregulation of ci is a dispensable feature of the phage lambda gene regulatory circuitry. *Journal of Bacteriology*, 187(18), 2005.
- [19] NewScientist.com. Computer generates verifiable mathematics proof. <http://www.newscientist.com/article.ns?id=dn7286>, April 2005. Report by Will Knight on Gonthier’s formal verification of a proof of the Four Colour Theorem using Coq.
- [20] POPLmark. The POPLmark Challenge. http://alliance.seas.upenn.edu/~plclub/cgi-bin/poplmark/index.php?title=The_POPLmark_Challenge. A challenge to make theorem-proving a feasible option for routine formalisation of articles submitted to the ACM POPL conferences.
- [21] Mark Ptashne. *A Genetic Switch: Phage Lambda Revisited*. Cold Spring Harbor Laboratory Press, 3rd edition, 2004.
- [22] Jittisak Senachak, Mun’de Vestergaard, and René Vestergaard. The CEq formal method. <http://cascade.jaist.ac.jp>.
- [23] Jittisak Senachak, Mun’delanji Vestergaard, and René Vestergaard. Cascaded games. In *Proceedings of the Second International Conference on Algebraic Biology, Lecture Notes in Computer Science 4545*, 2007.
- [24] Jittisak Senachak and René Vestergaard. The CEq formal method defined. <http://cascade.jaist.ac.jp/CoMEq/docs/definition.pdf>. 16 pages.
- [25] René Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3):563–585, 1973.
- [26] Freek Wiedijk. Formalizing 100 theorems. <http://www.cs.ru.nl/~freek/100/>. Status of the formalisation of the “Top 100+ Mathematical Theorems”; at 80% completion, Jan, 2008.

