Coloured Euler diagrams: a tool for visualizing dynamic systems and structured information

Paolo Bottoni[1] and Andrew Fish[2,⋆]

[1] Department of Computer Science, "Sapienza" University of Rome, Italy
bottoni@di.uniroma1.it
[2] School of Computing, Mathematical and Information Sciences,
University of Brighton, UK
Andrew.Fish@brighton.ac.uk

Abstract. Euler diagrams are a popular method for visualising sets and their relationships, exploited for resource management and visual logic specification, for example. We add the notion of colouring, provide a formal description of the extended system and demonstrate how coloured Euler diagrams provide adequate visualisations for concepts in the new bio-inspired model of Reaction Systems and for polyarchies, visualising multiple intersecting hierarchies.

1 Introduction

Venn and Euler diagrams were first introduced as an aid in syllogistic reasoning, but in recent times they have been utilised in various application domains, e.g. to represent genetic set relations [8], for file system management [3], or to represent the size of library database query results [15]. They can be viewed as a logic system in their own right, and incorporated into heterogeneous reasoning systems [14], or used to search for minimal proofs [13]. The various definitions of Venn and Euler diagrams (or Euler-like diagrams) in the literature slightly differ in syntax and semantics. We propose to extend the definition of an Euler diagram system to admit a notion of colouring, to which application-dependent semantics can be associated, in contrast with usage of colour as a secondary notation to highlight some information, without including it in the formal system.

Extending the formal system in this manner enables the precise use of these diagrams as visualisations incorporating information within this previously unutilised graphical dimension rather than via some other means that may lead to an increase in complexity of the notation or to more cluttered diagrams. We emphasize, however, that we use the term colouring in a mathematical sense of an assignation of numbers (with distinct numbers corresponding to distinct colours in the traditional sense), and whilst this notion maps in the natural manner to the use of actual colours on the diagram, this more abstract concept could in fact be represented graphically by different means if the situation required it. For instance, for colour-blind users, one could utilise an alternative graphical (or

⋆ Partially funded by UK EPSRC EP/E011160: Visualisation with Euler Diagrams.

concrete) representation making use of varying degrees of "dashned-ness" rather than colours such as red, green and blue, whilst utilising the same abstract model (and the same terminology of colouring at the abstract level). Moreover, we exploit vectors of colours, so that different forms of colouring can be used to visualise different aspects of the system (e.g. border color for identifying a database source, border thickness for expressing degree of relevance, and texture for identifying a keyword, in the presentation of the results of an OR query from federated databases, searching for documents presenting certain keywords).

The extension of the formal Euler diagram system to incorporate colouring has merit in that it is of theoretical interest to investigate ways of incorporating information within the diagrammatic notation, whether we are considering their use for simple data visualisation or perhaps more complicated uses such as the basis of diagrammatic logics. Their utility in the case of representing information that changes (e.g. set-based data that changes over time or diagrammatic logic statements that differ by diagrammatic logical inference rule application) is an avenue in which their usage may well bring user benefits, utilising colouring to indicate important changes between diagrams. The formalisation facilitates the use of the colouring concept within automated software systems.

We demonstrate the utility of coloured Euler diagrams as a representation system for both dynamical and static aspects within two different application domains. Firstly, we illustrate their usage for visualising notions such as modules along sequences of events, from Reaction Systems [4], a new bio-inspired model of computing where transformations rewrite the complete state of the system (so there is no resource counting). Secondly, we indicate the relationship of coloured Euler diagrams with hierarchies and with polyarchies [10], a representation of multiple intersecting hierarchies, utilised for categorisation in several domains.

We give intuition and motivations for Coloured Euler Diagrams (CEDs) in Section 2, and provide terminology and background definitions in Section 3 and a formal definition in Section 4. Section 5 gives an overview of Reaction Systems (*rs*) and discusses the visualisation of some *rs* features. In Section 6, we demonstrate their use in the domain of categorisation, comparing with visualisations such as Polyarchies which represent multiple intersecting hierarchies. Conclusions in Section 7 indicate other application areas for future investigation.

2 Motivation

Euler diagrams are a method of visually depicting a family of sets and their relationships. However, in many application domains, it is necessary to simultaneously present multiple families of sets. As an example, in an information system on the organisation of a multinational enterprise, one may categorise the personnel according to several dimensions, e.g. role, position in the organisational hierarchy, place of work. While each of these categories can be presented using Euler curves, there is no salient visual difference to support projection onto one of the categories (i.e. to distinguish the category type of the curves).

We explore here the visualisation of polyarchies, multiple overlapping hierarchies, each viewed as a family of sets. We propose to colour sets with a distinct colour for each hierarchy, with shared nodes coloured with a blend of the colours. In this way, one can exploit the spatial features of Euler diagrams, without having to resort to the use of multiple curve labels to indicate multiple categories.

Similarly, for dynamic structures, one wants to follow the individual evolution of families of phenomena, while maintaining a representation of the evolution of their relations. For example, for Reaction Systems discussed in this paper, interesting evolutions are those of states and of particular subsets, called modules.

Independent of the domain, two set-theoretic relationships related to this visualisation problem are those of *embedding* and *separation*, as defined in [4]. Informally, given a family of sets $\mathcal{L}$, a sub-family $\mathcal{F}$ is *embedded* in another sub-family $\mathcal{G}$ if all the component sets of $\mathcal{F}$ are contained in the intersection of the sets in $\mathcal{G}$. $\mathcal{F}$ is *separated* from $\mathcal{G}$ if it is embedded in $\mathcal{G}$ and there is a set $Y \in \mathcal{L}$ such that Y is contained in the intersection of all sets in $\mathcal{G}$, and the union of all sets in $\mathcal{F}$ is contained in Y. Providing a visual distinction between the two families may assist readers in assessing whether such relations exist.

In all of these cases, traditional visualisation through Euler diagrams does not distinguish between the families within a single diagram (unless one imposes extra labelling conventions). In our proposal, the incorporation of colouring to the Euler diagram system enables the indication of membership of a family via the use of colour, which in turn enables the representation of the set based relationships within the different families of sets as well as between these families. Furthermore, when considering dynamic information (for example variations over time in the composition of families of sets), using colouring provides a method of linking together the families of sets within a sequence of diagrams.

3 Set Systems and Colouring

We first recall standard notation for sets and set systems, and then define a general notion of coloured sets, subsets and set systems; the colourings can then be specialised according to the particular domain of application. We also define a method of deriving a colouring for subsets S from a colouring of sets X.

Let X be a set; then 2^X is the power set of X. A *set system* on X is a pair (X, S) where $S \subset 2^X$ is a set of subsets of X (see [2] for instance). Let $\Delta \subset \mathbb{Z}^+$ denote the set of prime numbers and $DIV(x)$ the set of prime integer divisors of x for $x \in \mathbb{Z}$. We assume that $\emptyset \in S$ for all set systems; this corresponds to the requirement that the "outside zone" is present in the diagrams, so the term set system becomes synonymous with the abstract diagram from [6].

Definition 1. *Let $X = \{x_1, \ldots, x_n\}$ be a finite set, and let $S = \{s_1, \ldots, s_k\} \subset 2^X$. Then an X-colouring is a function $c : X \to \mathbb{Z}$, and an S-colouring is a function $c' : S \to \mathbb{Z}$. Let c be an X-colouring. Then the* natural extension to subsets *of c is $c' : 2^X \to \mathbb{Z}$, given by $c'(\Sigma) = \prod_{i \in I} c(x_i)$, where $\emptyset \neq \Sigma = \{x_i :$*

$i \in I\} \subset 2^X$, for some finite index set I, and $c'(\emptyset) = 0$. A prime colouring *on X is an X-colouring such that $c(X) \subset \Delta$ and an* injective colouring *of X is a colouring c which is injective.*

Taking the natural extension to subsets as a product of set colours, and using distinct prime numbers for the original colourings ensures unique colourings for subsets. Colourings of subsets are *reducible* if they can be derived as products.

Lemma 1. *The natural extension to subsets of any injective prime colouring on X is an injective colouring on 2^X.*

Definition 2. *Let c be an X-colouring and c' an S-colouring. Then c* extends *to c' (or c'* reduces *to c) if c' is a restriction of the natural extension to subsets of c. Let c' be an S-colouring. Then c' is* reducible *if there exists an X-colouring c such that c' reduces to c.*

Lemma 2. *Let c' be an S-colouring on a set X, such that $c'(\emptyset) = 0$, and $\forall \Sigma \in S$, if $\mid \Sigma \mid = k$ then $c'(\Sigma)$ is a product of k distinct primes. Then, any injective function $div : X \rightarrow \Delta$, s.t. $\forall \Sigma \in S, x_i \in \Sigma \Leftrightarrow div(x_i) \in DIV(c'(\Sigma))$ is a prime injective X-colouring that extends to c'.*

Proof (sketch). The "if and only if condition" tells us that every subset (member of S) colouring is a prime number which divides the colouring of a set (member of X) if the set is a member of the subset. So every prime number that is used to colour a set that is a member of a subset must also divide the colouring of that subset. Since a subset is coloured by exactly k distinct primes if it has k members, the subset colour is precisely the product of the k primes that colour the corresponding k sets.

The previous two Lemmas will allow us to derive various colourings and provide some means of consistency checking later on. To allow for more than one colouring associated to a set system, we will use a vector of colouring functions.

Definition 3. *Let (X, S) be a set system. A* colouring *of (X, S) is a vector $K = (c_1, \ldots, c_k)$ of functions, with $k \geq 1$, where each c_i is either an X-colouring or an S-colouring for $i \in \{1, \ldots, k\}$. A set system (X, S) with a colouring K is called a* coloured set system, *denoted by (X, S, K).*

Example 1. Let $X = \{E, F, G, H\}$, $S = \{\emptyset, \{E\}, \{F\}, \{G\}, \{F, G\}, \{G, H\}\}$, let $c_1 : X \rightarrow \mathbb{Z}$ be defined by $c_1(E) = c_1(F) = c_1(G) = 0$ and $c(H) = 1$, and let $c_2 : S \rightarrow \mathbb{Z}$ be defined by $c_2(\emptyset) = c_2(\{G, H\}) = 0$, $c_2(\{E\}) = 1, c_2(\{F\}) = 2, c_2(\{G\}) = 3$ and $c_2(\{F, G\}) = 5$. Then c_1 is an X-colouring function, c_2 is an S-colouring function and (X, S, K) is a coloured set system, where $K = (c_1, c_2)$.

4 Coloured Euler Diagrams

We provide a basic definition of Euler diagrams, in a manner similar to other works (although we specify the definition in topological terms in order to ensure precision), and introduce the new concept of Coloured Euler Diagrams.

Commonly, somewhat restrictive set-ups are adopted, typically consisting of the well-formed diagrams of [6], where a decision procedure is provided to indicate if there is a well-formed concrete diagram realising an abstract diagram and if so to produce a drawing of it. The well-formedness conditions are concrete level constraints imposed on the system, with the intention of reducing human comprehension errors, and so there is validity in trying to preserve them; the conditions for [6] are: curves are simple (i.e. no self-intersection); curves only intersect at a finite number of points (i.e. no concurrent line segments); with at most two curves meeting at any particular point and such that the curves cross transversely at that point (i.e. curves that meet must really cross); no two curves have the same label; and no region of the plane that is inside a set of curves and outside the remaining set of curve is disconnected (no split zones).

However, since we wish to consider diagrams that are general enough to allow any set system to be represented (i.e. any abstract model has a visualisation), we cannot use uniquely labelled simple closed curves, as in [6]. Instead, we use unions of simple closed curves (or more precisely, the region bounded by them) to represent the sets of a set system, with labels determining the association. We use the term *contour* for a set of curves with a label in common, agreeing with the usage in [6] when restricting to well-formed Euler diagrams.

In general, any set-up can utilise the proposed conceptual extension of colouring. For example, in [2] some well-formedness conditions are relaxed and the term Euler-like diagrams is adopted for variations such as allowing "holes". In [16] it is shown that all abstract diagrams for at most eight sets are drawable using such Euler-like diagrams. Recently, a methodology for generating general Euler diagrams has been developed [11], ensuring the production of a diagram, with a heuristic approach to repairing possible breaks in well-formedness.

Let C be a simple closed curve in the plane. Then $\overline{C}$ denotes the closed region of the plane bounded by C (homeomorphic to a disc), $int(\overline{C})$ the interior of $\overline{C}$, and $ext(\overline{C})$ the exterior of $\overline{C}$. We say that curves $C_1, \ldots, C_n$ are *closure-disjoint* if $\overline{C_1} \cap \ldots \cap \overline{C_n} = \emptyset$; and C_2 is *closure-contained in* C_1 if $\overline{C_1} \cap \overline{C_2} = \overline{C_2}$.

Definition 4. *Let Λ be a countable alphabet of labels. Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ be a family of curves in the plane with a finite number of points of intersection and $l : \mathcal{C} \to 2^{\Lambda}$ a function assigning a set of labels to each curve in $\mathcal{C}$. For $\lambda \in \Lambda$, if the set $C'_\lambda = \{C_i \in \mathcal{C} \mid \lambda \in l(C_i)\} \subset \mathcal{C}$, of the curves with λ as a label, is non-empty then it is called the* contour *of λ. Let $\mathcal{C}' = \{C'_\lambda \mid \lambda \in \Lambda\}$ be the set of contours in $\mathcal{C}$. Let $L(C'_\lambda) = \bigcup_{C_i \in C'_\lambda} l(C_i)$, the set of labels for the curves in C'_λ, and $L(\mathcal{C}') = \bigcup_{C'_\lambda \in \mathcal{C}'} L(C'_\lambda)$. If $\{C_i \mid C_i \in C'_\lambda\}$ is closure-disjoint, for each $\lambda \in \Lambda$, then we say that $d = (\mathcal{C}, l)$ is an* Euler diagram*. The maximal connected sets of points of the plane with the curves in $\mathcal{C}$ removed are called the* minimal regions *of d, denoted $R(d)$. Let $\mathcal{Y} \subset \mathcal{C}'$ be a set of contours. Let z be the region of the plane inside $\mathcal{Y}$ but outside $\mathcal{C}' \setminus \mathcal{Y}$. If z is nonempty then it is called a* zone*:*

$$z = \bigcap_{c \in \mathcal{Y}} int\,(\overline{c}) \;\cap \bigcap_{c \in \mathcal{C}' \setminus \mathcal{Y}} ext\,(\overline{c}) \neq \emptyset.$$

The set of zones of d is denoted by $Z = Z(d)$.

In terms of the interpretation of the diagrams, as usual we have that if two regions determined by two contours do not overlap then the sets represented are disjoint, whilst if the region of one contour E_1 is wholly contained in the region of another contour E_2 then we have the subset relationship $E_1 \subset E_2$.

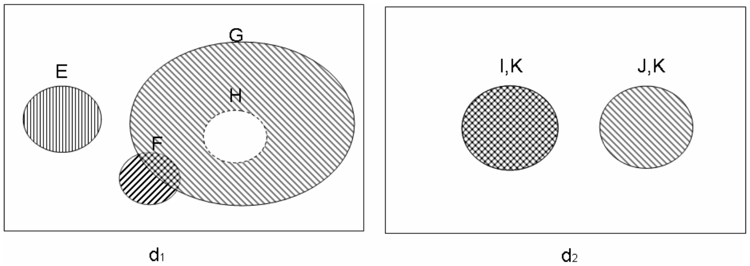

Fig. 1. Examples of coloured Euler diagrams. d_1 demonstrates both curve and region colouring; d_2 demonstrates the use of multiple labels and curves for set representation.

Example 2. Figure 1 shows two (coloured) Euler diagrams d_1 and d_2, where we have drawn a (rectangular) bounding box around each diagram and labelled the diagrams in order to distinguish where one diagram ends and the other begins. In d_1 we have four curves, each with one label (E, F, G, and H), which are all therefore contours; in fact these curves determine a well-formed Euler diagram of [6]. There are six zones in the diagram, described by their set of containing curve labels: $\emptyset$, $\{E\}$, $\{F\}$, $\{G\}$, $\{F, G\}$, and $\{G, H\}$. This diagram realises the set system of Example 1. In terms of set-theoretic relationships we infer: $H \subset G$, $E \cap F = \emptyset$, $E \cap G = \emptyset$, and $F \cap H = \emptyset$. Diagram d_2, with two curves and three contours, realises the set system (X, S), with $X = \{I, J, K\}$ and $S = \{\emptyset, \{I, K\}, \{J, K\}\}$. In this complex example X is a set of sets and we label curves with the names of the sets, so that they have a label in common. The contours for I and J coincide with the curves with that label in their label set. Applying a standard interpretation of Euler diagrams, we have that the two regions contained by the two curves represent the sets $I \cap K$ and $J \cap K$. As missing zones represent the empty set, we derive $(I \cap K) \cap (J \cap K) = \emptyset$, and so the diagram supports the inference $I \cup J = K$. One can observe the trade-off between the complexity of interpretation of these diagrams with multiple curves and labels and the ability to realise any set system.

With regard to the extension to colouring we demonstrate two realisations of colouring functions in Example 2: colouring the curves themselves, or utilising textures for diagram regions. For overlapping regions, one could choose independent textures, or some form of meshing of textures from the basic regions, giving a visual indicator of which regions are overlapping, similar to that in [12]. The formal definition of an instantiation function from the abstract colouring functions to such graphical attributes is omitted here for simplicity. Before we give the definition of colouring, we provide some consistency conditions: firstly, if any two curves with the same label set have the same colour, and there is a colouring of contours that induces this colouring on curves then we have curve-contour colour consistency; secondly, if a colouring of minimal regions can be induced from a colouring on zones which in turn can be induced by a colouring of contours then we have region-contour consistency.

Definition 5. *Let $\mathcal{C}$ be a set of labelled curves, forming a set of contours $\mathcal{C}'$ and let c be a colouring on $\mathcal{C}$ s.t. $\forall C_i, C_j \in \mathcal{C}$, we have $l(C_i) = l(C_j) \Rightarrow c(C_i) = c(C_j)$. Then $l(\mathcal{C}) \subset 2^{L(\mathcal{C}')}$ and c defines a colouring $c_1 : l(\mathcal{C}) \to \mathbb{Z}$. If c_1 is reducible to a $L(\mathcal{C}')$-colouring then we have* curve-contour colour consistency*. Let R be the set of minimal regions defined by $\mathcal{C}$, and suppose that c'' is a colouring of R. If c'' satisfies the properties that: (i) every minimal region of any zone z has the same image under c''; (ii) the Z-colouring determined by c'' is reducible to a $\mathcal{C}'$ colouring, then we have* region-contour colour consistency*.*

Remark 1. If all contours have unique single labels, then a 1-1 correspondence exists between contour labels and contours, and since every curve with the same label set has the same colour, also between curve label sets and curves. In this case colourings of $L(\mathcal{C}')$ and $\mathcal{C}'$ are equivalent, as well as those of $l(\mathcal{C})$ and of $\mathcal{C}$.

Definition 6. *Let $d = (\mathcal{C}, l)$ be an Euler diagram and R its set of minimal regions. A function $c : \mathcal{C} \to \mathbb{Z}$ is called a $\mathcal{C}$-colouring function and a function $c : R \to \mathbb{Z}$ is called an R-colouring function. A* colouring *of $d = (\mathcal{C}, l)$ is a vector $K = (c_1, \ldots, c_k)$ with $k \geq 1$, and each c_i either a $\mathcal{C}$-colouring function or an R-colouring function. An Euler diagram $d = (\mathcal{C}, l)$ with a colouring K is called a* coloured Euler diagram *(abbreviated CED), denoted $(\mathcal{C}, l, K)$, if each component of K satisfies either curve-contour colour consistency or region-contour colour consistency, according to its domain being $\mathcal{C}$ or R, respectively.*

The consistency relationships ensure that we have colouring functions on the set of contours and zones of the diagram, which therefore give rise to colouring functions on the underlying set system that the diagram is representing.

Example 3. From Figure 1, in terms of colouring, d_1 has a 2-component colouring vector $K = (c_1, c_2)$, the same as for the set system in Example 1, where c_1 is realised by border colouring (with correspondences $0 \longleftrightarrow$ **solid**, and $1 \longleftrightarrow$ **dashed**) and c_2 is realised by region texturing. The diagram d_2, with two curves and three contours, is equipped with a 1-component R-colouring function $c_1 : R \to \mathbb{Z}$ such that $c_1(\emptyset) = 0$, $c_1(\{I, K\}) = 1$, and $c_1(\{J, K\}) = 2$, realised using textures.

We relate the notions of coloured set system and concrete coloured Euler diagram, and show that any coloured set system can be realised by a concrete coloured Euler diagram. Diagram d_2 from Example 2 shows an example of an application of the proof strategy. Notice that consistency amongst colouring on curves and contours is incorporated into the definition of CEDs, whilst consistency between sets and contours is incorporated in the definition of realisation.

Definition 7. *Let (X, S) be a set system. Then an Euler diagram $d = (\mathcal{C}, l)$ is a* realisation *of (X, S) if there is a bijection $b : X \leftrightarrow \mathcal{C}'$ which induces a bijection $b_s : S \leftrightarrow Z(d)$. Also, (X, S) is called the* abstraction *of $d = (\mathcal{C}, l)$. Let $(d = \mathcal{C}, l, K')$ be a CED and (X, S, K) a coloured set system, such that $(\mathcal{C}, l)$ is a realisation of (X, S). Then we say that K'* realises the colouring K *of (X, S), if $\exists k \in \mathbb{Z}^+, \mid K \mid = \mid K' \mid = k$ and the bijection b respects the colouring vectors: $\forall i \in \{1, \ldots, k\}$ $c_i(x) = y \Leftrightarrow c'_i(b(x)) = y$, for $x \in X$, and $c'_i(s) = y \Leftrightarrow c_i(b_s(s)) = y$, for $s \in S$ (according to c_i's being a contour or a region colouring). We call $(\mathcal{C}, l, K')$ a* realisation *of (X, S, K) and (X, S, K) the* abstraction *of $(\mathcal{C}, l, K')$.*

In order to realise coloured set systems, and to reduce the amount of information needed to define a consistent colouring, we show how to induce colourings on curves and regions from contours in a diagram by: colouring curves with the product of the colours of the contours the curve belongs to; inducing region colouring from contour colouring via the zone colouring obtained as a natural extension to subsets.

Definition 8. *Let $d = (\mathcal{C}, l)$ be an Euler diagram and let $c : \mathcal{C}'(d) \to \mathbb{Z}$ be a colouring of the contour set of d. Then the* induced curve colouring $c_c : \mathcal{C}(d) \to \mathbb{Z}$ *is given by: $c_c(C) = \prod_{\{C' \in \mathcal{C}'(d) \mid C \in C'\}} c(C')$ for $C \in \mathcal{C}(d)$, whilst the* induced region colouring $c_r : \mathcal{R}(d) \to \mathbb{Z}$ *is obtained from the extension of c to zones by setting every minimal region's colour to be the colour of the zone containing it.*

Remark 2. In the following proof, and later on, we make use of the important notion of induced colourings: a set colouring (i.e an X-colouring) gives rise to a colouring of contours from which we can induce a colouring of curves, or in fact a colouring of zones, and hence minimal regions. If all of the colourings of a diagram are induced from contour colourings, then this ensures the consistency of all relations required for the diagram to be a CED; i.e. an Euler diagram $d = (\mathcal{C}, l)$ together with a vector of induced colouring functions is a CED.

Theorem 1. *(i) Let $d = (\mathcal{C}, l)$ be an Euler diagram realising a set system (X, S) and let c be an X-colouring. Then c gives rise to a colouring of contours, via the bijection of the realisation, which induces a curve or region colouring of d.*

(ii)Any coloured set system (X, S, K) can be realised as a CED $d = (\mathcal{C}, l, K)$.

Proof. The first part follows from definitions. For the second part, we first show that any set system can be realised. For every $x \in S \setminus \emptyset$, take one curve[3] labelled by x. This yields a Euler diagram $d = (\mathcal{C}, l)$ with the correct zone set. Map the S-colourings of (X, S) to R-colourings of $(\mathcal{C}, l)$, using the Z-colouring determined by S and the same colour for the set of minimal regions comprising a zone. Map the X-colourings to colourings of $\mathcal{C}'$, the contours of d, updating the colourings of curves with multiple labels (i.e. take the induced colouring on curves).

[3] The construction is uniform whether S is a set of distinct elements, or a set of sets. In the first case we consider each element as a singleton containing exactly it.

For families of sets depicted in a sequence of diagrams, as a sequence of contours, we define a contour sequence, tracking a set through a sequence of diagrams, and a colouring respects this contour sequence if it is consistent across the sequence. Extending to a family of these sequences gives rise to the $\mathcal{Y}^*$-respectful sequences, covering a sequence of diagrams if every contour appears in the $\mathcal{Y}^*$-respectful sequence.

Definition 9. *Let $\mathcal{D} = (d_1, \ldots, d_n)$, with $d_h = (\mathcal{C}_h, l_h)$, be a sequence of Euler diagrams. For each d_h, $\mathcal{C}'_h$ is its contour set and (X_h, S_h) its set system abstraction. Let $\mathcal{Y} = (Y_1, \ldots, Y_n)$ be a sequence of elements*[4]*, with $Y_i \in X_i$, for $i \in \{1, \ldots, n\}$. Then $C'(\mathcal{Y})$, the* contour sequence *of $\mathcal{Y}$, is the sequence $(C'_{Y_1}, \ldots, C'_{Y_n})$ of contours in $\mathcal{D}$ s.t. C'_{Y_i} corresponds to Y_i under the bijection of the realisation. Let $\mathcal{C}'(\mathcal{Y})$ denote the set of contours in the sequence. Let each d_h be coloured by a colouring K^h. If $C'_{Y_j}, C'_{Y_k} \in \mathcal{C}'(\mathcal{Y}) \Rightarrow K^j(C'_{Y_j}) = K^k(C'_{Y_k})$, then we say that the colourings* respect the contour-sequence *$C'(\mathcal{Y})$. Let $\mathcal{Y}^* = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_k\}$ be a collection of k contour sequences. If the colourings respect the contour sequence of $\mathcal{Y}_i$ for all $i \in \{1, \ldots, k\}$, then we say that the sequence of CEDs $\mathcal{D}' = (d'_1, \ldots, d'_n)$, with $d'_h = (\mathcal{C}_h, l_h, K^h)$, is a $\mathcal{Y}^*$-*respectful CED sequence. *If $\bigcup_{j=1}^{k} \mathcal{C}'(\mathcal{Y}_j) = \bigcup_{i=1}^{n} \mathcal{C}'(d_i)$ then we say that $\mathcal{Y}^*$* covers *$\mathcal{D}$.*

Theorem 2. *Let $\mathcal{D} = (d_1, \ldots, d_n)$, with $d_h = (\mathcal{C}_h, l_h)$, be a sequence of Euler diagrams, such that (X_h, S_h) is the set system abstraction of d_h. Let $\mathcal{F}^* = \{\mathcal{F}_1, \ldots, \mathcal{F}_k\}$ be a collection of families of sets, such that each $\mathcal{F}_i$ defines a contour sequence of $\mathcal{D}$, and $\mathcal{F}^*$ covers $\mathcal{D}$. Then there is a colouring function that makes $\mathcal{D} = (d_1, \ldots, d_n)$ into a $\mathcal{F}^*$-*respectful CED sequence.

Proof. Assign a unique prime colour to each family $\mathcal{F}_i$, thereby assigning a colour to the contour sequence for that family. Then take the colourings in each diagram d_h of the CED sequence to be those induced from the contour colourings. This yields an $\mathcal{F}^*$-respectful CED sequence, as required.

5 Representing Reaction Systems with CEDs

In this Section we propose the use of CEDs to represent some significant notions from the recently proposed computational model of Reaction Systems [5, 4]. A Reaction System (rs) is an ordered pair $\mathcal{A} = (X, A)$, where X is a finite (background) set, and A is a finite set of reactions of the form $a = (R_a, I_a, P_a)$, s.t. $R_a, I_a, P_a \subseteq X$ for each $a \in A$. R_a is called the set of reactants, I_a of inhibitors, and P_a of products. A reaction a is enabled in a state T if $R_a \subseteq T$ and $I_a \cap T = \emptyset$. If a is enabled in T, then P_a is produced as the new state, replacing T, denoted by $res_{\mathcal{A}}(T) = P_a$, or $T \overset{\mathcal{A}}{\rightarrow} P_a$; otherwise the result of applying a to T is the empty set. Note that a can be enabled only if $R_a \cap I_a = \emptyset$.

[4] Note that the X_h are the set of elements of the set system, but these elements Y_i are themselves sets in the applications and we consider them as such here. Of course, there is a natural map from elements to singleton sets that can be utilised if required.

Figure 2 shows an informal graphical representation of a set of reaction rules on background set $X = \{1, 2, 3, 4, 5\}$, as a respectful CED sequence for the family $\{R, I, P\}$, augmented with information of set membership. Here we provide a legend on the left of the figures indicating the colour coding.

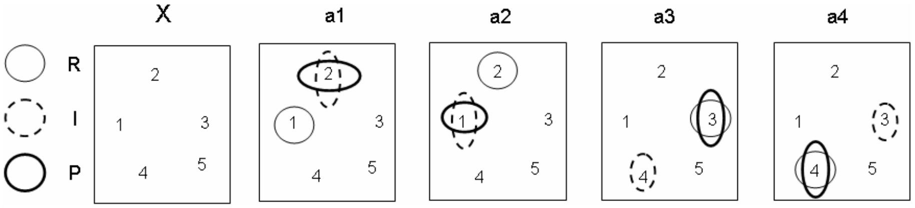

Fig. 2. Four reaction rules $a_1, \ldots, a_4$ on a background set X. For example, as $R_{a_1} = \{1\}, I_{a_1} = \{2\}$ and $P_{a_1} = \{2\}$, in a state containing 1 but not 2, rule a_1 produces 2.

These notions are immediately extended to sets: given a set A, $en_A(T) \subset A$ is the set of reactions from A enabled in T. The effect of applying A to T is then the cumulative effect $res_{\mathcal{A}}(T) = \bigcup_{a \in en_A(T)} P_a$. For simplicity, we consider here only self-sustaining processes, i.e. without any contribution from the environment. Hence, we identify the evolution of the process with the sequence of states produced by the *rs*. A process π is characterised by a finite sequence $sts(\pi) = (W_0, \ldots, W_n)$, s.t. for $0 < i \leq n$, $W_i \subseteq X, W_i = res_{\mathcal{A}}(W_{i-1})$.

An *extended reaction system* (or *ers*) is an ordered pair $\mathcal{E} = (\mathcal{A}, R)$, where $\mathcal{A} = (X, A)$ is an *rs*, and $R \subset 2^X \times 2^X$. R is a restriction relation s.t. in the sequence $(W_0, \ldots, W_n)$ each pair $(W_i, W_{i+1}) \in R$. $\mathcal{E}$ is also denoted (X, A, R).

Let $\mathcal{E} = (X, A, R)$ be an *ers*, $\mathcal{W} = (W_0, \ldots, W_n)$ a finite sequence of states for some process in $\mathcal{E}$, and $\omega = (Q_i, \ldots, Q_j)$ a sequence of sets s.t. $Q_i \subset W_i, \ldots, Q_j \subset W_j$ and $Q_i = res_{\mathcal{A}}(Q_{i-1})$, for $0 < i \leq j \leq n$ (see Figure 3). Under some technical conditions (see [4]), ω is called an *event* and each Q_i a *module* of ω in W_i. A *snapshot* $\mathcal{S}_k$ of ω at step k is the set of modules of W_k.

Respectful CED sequences with induced colourings provide an immediate representation of modules. We state here the required properties of the sequences, and claim that explicit constructions exist (not shown here due to lack of space), using prime injective colourings and the notion of inducing to generate them.

Definition 10. *Let $\omega = (Q_i, \ldots, Q_j)$ be an event of $\mathcal{E}$ with $\mathcal{W}$ the corresponding sequence of states. Let $\mathcal{D} = (d_0, \ldots, d_{j-i})$ be a finite sequence of CEDs, s.t. each d_h, with $0 \leq h \leq j - i$, is a realisation of $(\{W_{i+h}, Q_{i+h}\}, T, K^h)$ for some colouring K^h, $T \subset 2^{\{W_{i+h}, Q_{i+h}\}}$, and the curves for the contours Q'_{i+h} are closure-contained in the curves for the contours W'_{i+h}, with curves identified if the corresponding sets are equal. We call $\mathcal{D}$ a* representation *of ω if the two resulting contour sequences $\mathcal{W}', \mathcal{Q}' \in \mathcal{C}'(\mathcal{D})$ are such that $\mathcal{Y}^* = \{\mathcal{W}', \mathcal{Q}'\}$ covers $\mathcal{D}$, and $\mathcal{D}$ is a $\mathcal{Y}^*$-respectful CED sequence.*

Remark 3. Here, and in the following, when constructing CED sequences, the curve and region colourings are induced from the contour colourings. So, if some of the Q_i and W_i are equal (so their curves coincide) the curve colouring used is obtained from the contour colourings (taking the product of the colours of the contours). We also use different types of colouring (curve versus region) to indicate the different types of families (states versus modules).

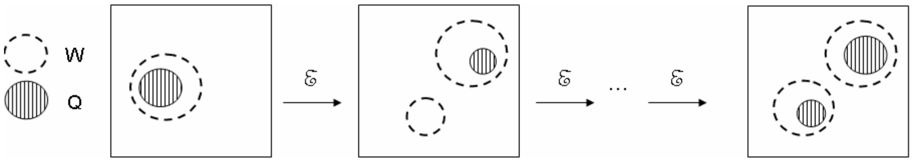

Fig. 3. A graphical representation of states (W_i) and modules (Q_i) using CEDs; here each subsequent diagram represents a single evolution of the system.

Events can merge in one state and must remain merged thereafter. Let ω and ω^1 be two events of $\mathcal{E}$. We say that ω *merges* ω^1 in W_i if Q_i is a module of ω in W_i, Q_i^1 a module of ω^1 in W_i and $Q_i \subseteq Q_i^1$. Then ω and ω^1 remain merged as both processes evolve through $W_{i+1}, W_{i+2}, \ldots, W_j$, i.e. $Q_k \subset Q_k^1$ for all $k \in \{i, \ldots, j\}$. We provide a CED characterisation of merging as follows, from which the representation in Figure 4 derives.

Definition 11. *Let ω, $\mathcal{D}$, $\mathcal{W}$, $\mathcal{Q}$, $\mathcal{W}'$, $\mathcal{Q}'$ be as in Definition 10 and ω^1 another event on the same $\mathcal{W}$, with a corresponding family of modules $\mathcal{Q}^1$. Then we say that a sequence of diagrams $\mathcal{D}^2$ is a* representation for the merging *of ω and ω^1 iff: 1) there exists a sequence of diagrams $\mathcal{D}^1$ which is a representation of ω^1 s.t. the families of contours $\mathcal{W}'$ and $\mathcal{W}'^1$ coincide in $\mathcal{D}$ and $\mathcal{D}^1$, with distinct colourings for contours in $\mathcal{Q}'$ and $\mathcal{Q}'^1$ and same colourings for $\mathcal{W}$. 2) There exists an operation $*$ which combines pairs of the diagrams, in sequential order, preserving contours and contour colourings, and which merges curves which represent the same set, s.t. $\mathcal{D}^2 = \mathcal{D} * \mathcal{D}^1$. 3) $\mathcal{D}^2$ is a respectful sequence of diagrams for $\mathcal{Y}^* = \{\mathcal{W}', \mathcal{Q}', \mathcal{Q}'^1\}$. 4) $\mathcal{Y}^*$ covers $\mathcal{D}^2$.*

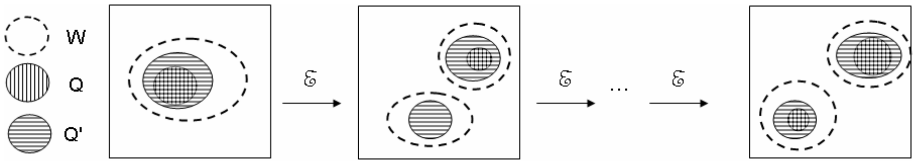

Fig. 4. A representation of the merging of events showing two sequences of modules Q_i and Q'_i with the same set of states W_i.

A further refinement considers a set Q which is a module for different states W_k, W_l^1 from two state sequences τ and τ^1. If $W_k \subseteq W_l^1$, then the successor of Q

in τ^1 (i.e. the module obtained from Q by the application of the set of reactions) is a subset of the successor of Q in τ. We define a colouring scheme for combining diagrams which represent different state sequences in the same *ers*.

Definition 12. *Let ω, ω_1 be two events from sequences of states $\mathcal{W}$ and $\mathcal{W}^1$, with families of modules $\mathcal{Q}$ and $\mathcal{Q}^1$. A sequence $\mathcal{D}^2$ is a* representation for common events *iff: 1) There exist two sequences of CEDs $\mathcal{D}$ and $\mathcal{D}^1$ which are representations for ω and ω_1, respectively. 2) All colourings for contours in $\mathcal{Q}'$, $\mathcal{Q}'^1$, $\mathcal{W}'$, $\mathcal{W}'^1$ are distinct. 3) $\mathcal{D}^2 = \mathcal{D} * \mathcal{D}^1$, with $*$ as in Definition 11. 4) $\mathcal{D}^2$ is a respectful CED sequence for $\mathcal{Y}^* = \{\mathcal{W}', \mathcal{W}'^1, \mathcal{Q}', \mathcal{Q}'^1\}$. 5) $\mathcal{Y}^*$ covers $\mathcal{D}^2$*

In Figure 5, curve colouring (dashed versus non-dashed) distinguishes the W_i states and texture (horizontal vs. vertical lines) distinguishes modules. The colour for the common module is different from those of the states and of the other modules, and is shown by combining the colourings for the different events.

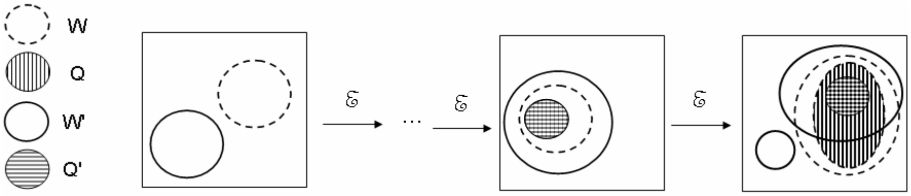

Fig. 5. Representing common events: Q and Q' are modules for W and W' resp.

In [4], it is shown that each snapshot in a sequence of snapshots is a partial order, with top and bottom elements. A fundamental property of snapshots is that, given two consecutive snapshots $\mathcal{S}_k, \mathcal{S}_{k+1}$ and $\mathcal{F}, \mathcal{G}$ two families of subsets in $\mathcal{S}_k$, if $\mathcal{F}$ is embedded in $\mathcal{G}$, and each module in $\mathcal{F}$ and $\mathcal{G}$ has a successor in $\mathcal{S}_{k+1}$, then $\mathcal{F}'$ is separated from $\mathcal{G}'$ in $\mathcal{S}_{k+1}$, as shown in Figure 6.

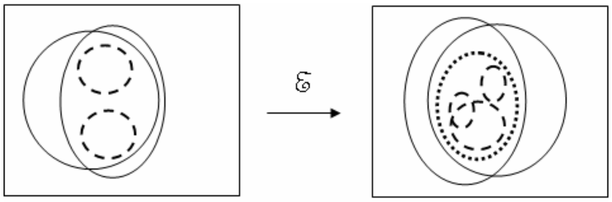

Fig. 6. From embedding to separation. Left: S_k with $\mathcal{F}$, dashed, embedded in G, bold. Right: S_{k+1} with $\mathcal{F}$ separated from $\mathcal{G}$ by the dotted curve.

6 Information Structure Visualisations

We describe some information structure visualisations and indicate how CEDs could be used in their place. A basic structure is a hierarchy, modeled as a rooted, directed tree. Viewing how a single hierarchy changes over time is an important task in some domains (e.g. the TimeTube visualization uses Disk Trees to represent the evolution of the web ecology [1]). Placing hierarchies next to each other and connecting common nodes with lines was a method used to visualise taxonomies in [7]. Multiple intersecting hierarchies that share common nodes have been called Polyarchies [10]. Visualization techniques include the visual pivot, which shows how hierarchies relate to each other in the context of various entities: the specified entities are highlighted, the new hierarchy appears and rotates into place and the old hierarchy rotates out of place and fades away, leaving the new hierarchy, with the user being shown the transition. A stacked link view is also adopted, showing multiple hierarchies with lines linking the common nodes (an example is shown on the left of Figure 7). This enables a similar animation for viewing several related hierarchies, which preserves the previous view. In [9], Polyarchies are viewed as edge coloured multigraphs, effectively considering them as the union of rooted trees, each corresponding to a single hierarchy. CEDs can be used to represent hierarchies: a stacked link view can be represented as a sequence of CEDs with colour indicating commonality arising from the common nodes of the hierarchies; this then extends to a visualisation using a CED sequence instead of a Polyarchy in stacked link view.

Definition 13. *Let $T = (V, E, r)$ be a tree, with vertex set V and directed edge set E, rooted at $r \in V$. E^t is the transitive closure of E. Let d_T be an Euler diagram with contour set $\mathcal{C}'$ and one curve for each contour, s.t. there is a bijection $b : V \rightarrow \mathcal{C}'$ for which $(v_1, v_2) \in E^t$ iff the curve for $b(v_1)$ is contained in the interior of the curve for $b(v_2)$. Then d_T, together with any C-colouring function c assigning the same value to all curves, is a* coloured Euler diagram *of T.*

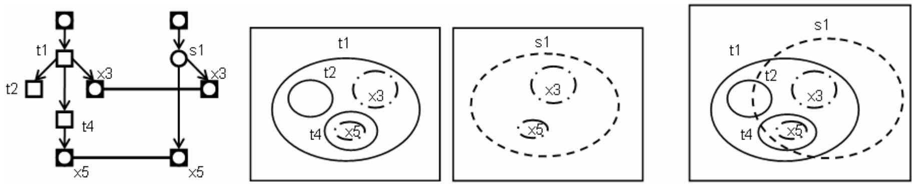

Fig. 7. Left: A polyarchy, consisting of two hierarchies T_1 and T_2 joined by lines indicating nodes in common (i.e. this is a stacked link view of T_1 and T_2). Middle: A pair of coloured Euler diagrams d_1 and d_2, each one representing the corresponding hierarchy with the colour (dotted curves) indicating the nodes in common. Right: A single combined diagram d_3 representing the polyarchy.

Definition 14. *Let T_1 and T_2 be rooted trees with a set of vertices $V = V(T_1) \cap V(T_2)$ in common. Then $G(T_1, T_2)$, the* stacked link view *of T_1 and T_2, is the partially directed graph $T_1 \sqcup T_2$ together with an undirected edge set between the pairs of vertices in T_1 and T_2 in common. A* visualisation of the stacked link view *of T_1 and T_2 is a CED sequence d_1, d_2 such that d_1 and d_2 are coloured Euler diagrams for T_1 and T_2 respectively, using one same colour c for the nodes in V, and not using c for any node in the symmetrical difference $(V(T_1) \cup V(T_2)) \setminus V$.*

Lemma 3. *Let d_{T_1} and d_{T_2} be the CEDs of T_1 and T_2 respectively, with the T_i as in Definition 14 Let nodes in T_1 be coloured with a_1 and those in T_2 with a_2, a_1 and a_2 being distinct prime integers. Let d_1 be d_{T_1} with colouring altered so that the colour of curves of vertices in V is $a_1 * a_2$ whilst those not in V are coloured a_1; similarly, d_2 is d_{T_2} except that the colour of curves of vertices in V is $a_1 * a_2$. Then the coloured CED sequence $d_G = (d_1, d_2)$ of $G(T_1, T_2)$ is a visualisation of the stacked link view of T_1 and T_2.*

Example 4. Figure 7 shows a polyarchy consisting of two hierarchies T_1 and T_2 joined by lines indicating shared nodes. We have augmented the usual representation of trees with a form of colouring, using shapes [square or circle] to represent nodes from each hierarchy, and an overlapping square and circle to represent nodes in common. The respectful CED sequence which is the visualisation of the stacked link view of T_1 and T_2 is shown in the middle of the figure as d_1 and d_2. The right hand side of Figure 7 shows d_3, an example of the combination of the two diagrams d_1 and d_2, which offers a potential opportunity to reduce the length of diagram sequences, when it is possible.

Theorem 3. *Let P be any polyarchy consisting of multiple trees in a stacked link view. Then there is a coloured Euler diagram sequence visualising P.*

Proof (sketch). Let $T_1, \ldots, T_n$ denote the n hierarchies of the polyarchy. For each T_i define an ED as usual. Assign a unique prime colour p_i to each T_i. For a node n common to exactly the hierarchies $T_{n_1}, \ldots, T_{n_j}$, assign to n the colour $p_{n_1} * \ldots * p_{n_j}$. Then each node of any T_i corresponds to a contour of the CED, and if the node is shared amongst hierarchies, the corresponding contours in each diagram have the same colour.

7 Conclusion

We have extended the notion of Euler diagrams to incorporate colourings, providing extra dimensions for representing various domain features, and demonstrated their applicability as visualisations of concepts in two different domains: Reaction Systems and Information Structure visualisations. We draw representations of coloured Euler diagrams where the colouring functions are instantiated by graphical attributes such as actual colour, texture, dashedness, etc, and the choice of such instantiations can be important in terms of human perception and understanding, but they have a common encoding in a mathematical sense.

Colouring functions can be applied to alternative Euler based diagram systems. Colouring is applied to set systems and diagrams and we have shown that any coloured set system can be realised as a coloured Euler diagram using the set-up in this paper. Also, we have presented methods to induce colourings of curves and regions from a given specification of colourings on sets. One extra benefit of this is to reduce the amount of information required for such colouring specifications which may prove useful in implemented systems.

Possible directions for future work include: integration of colouring in logical systems, either to assist human readers in comprehension of logical steps involving conjunction of two diagrams, or to allow representation of attributes; extending the representation devised for polyarchies to representation of hypergraphs; representation of clusters, highlighting their changes over time along a sequence of diagrams. On the theoretical side, we plan to extend the proposal with notions of diagram combination and matching.

References

1. Ed H. Chi, James Pitkow, Jock Mackinlay, Peter Pirolli, Rich Gossweiler, and Stuart K. Card. Visualizing the evolution of web ecologies. In *Proc. CHI '98*, pages 400–407. ACM Press/Addison-Wesley, 1998.
2. Stirling C. Chow. *Generating and Drawing Area-Proportional Euler and Venn Diagrams.* PhD thesis, University of Victoria, 2007.
3. R. De Chiara, U. Erra, and V. Scarano. VennFS: A Venn diagram file manager. In *Proc. InfoVis 2003*, pages 120–126. IEEE Computer Society, 2003.
4. Andrzej Ehrenfeucht and Grzegorz Rozenberg. Events and modules in reaction systems. *Theoretical Computer Science*, 376(1-2):3–16, 2007.
5. Andrzej Ehrenfeucht and Grzegorz Rozenberg. Reaction systems. *Fundamenta Informaticae*, 75(1-4):263–280, 2007.
6. J. Flower, A. Fish, and J. Howse. Euler diagram generation. *Journal of Visual Languages and Computing*, 19:675–694, 2008.
7. Martin Graham and Jessie Kennedy. Extending taxonomic visualisation to incorporate synonymy and structural markers. *Inform. Visualiz.*, 4(3):206–223, 2005.
8. H. Kestler, A. Muller, T. Gress, and M. Buchholz. Generalized Venn diagrams: A new method for visualizing complex genetic set relations. *Journal of Bioinformatics*, 21(8):1592–1595, 2005.
9. M.J. McGuffin and M.C. Schraefel. A comparison of hyperstructures: Zzstructures, mspaces, and polyarchies. In *Proc. 15th ACM Hypertext and Hypermedia*, pages 153–162. ACM Press, 2004.
10. George Robertson, Kim Cameron, Mary Czerwinski, and Daniel Robbins. Polyarchy visualization: Visualizing multiple intersecting hierarchies. In *Proc. ACM SIGCHI*, pages 423–430. ACM Press, 2002.
11. P. Rodgers, L. Zhang, and A. Fish. General euler diagram generation. In *Proc. Diagrams 2008*, volume 5223 of *LNAI*, pages 13–27. Springer, 2008.
12. Paolo Simonetto, David Auber, and Daniel Archambaul. Fully automatic visualisation of overlapping sets. *Computer Graphics Forum*, 28:967–974, 2009.
13. G. Stapleton, J. Masthoff, J. Flower, A. Fish, and J. Southern. Automated theorem proving in Euler diagrams systems. *J. of Autom. Reas.*, 39(4):431–470, 2007.
14. N. Swoboda and G. Allwein. Using DAG transformations to verify Euler/Venn homogeneous and Euler/Venn FOL heterogeneous rules of inference. *Software and System Modeling*, 3(2):136–149, 2004.
15. J. Thièvre, M. Viaud, and A. Verroust-Blondet. Using euler diagrams in traditional library environments. In *Proc. Euler Diagrams 2004*, volume 134 of *ENTCS*, pages 189–202, 2005.
16. A. Verroust and M.-L. Viaud. Ensuring the drawability of Euler diagrams for up to eight sets. In *Proc. Diagrams 2004*, volume 2980 of *LNAI*, pages 128–141, Cambridge, UK, 2004. Springer.