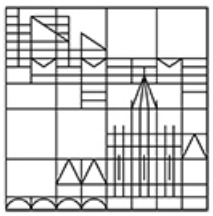
Fachbereich Informatik und
Informationswissenschaft

Universität
Konstanz



Lecture Notes

Network Dynamics: Opinion Dynamics

taught in Winter terms 2010, 2011, 2012

by

Sven Kosub

November 29, 2012

Version v1.6

Contents

How Can Extremism Prevail? 1

In order to give an impression on what opinion dynamics is about, we present a case study based on an article that received some echo in several scientific disciplines:

> Guillaume Deffuant, Frédéric Amblard, Gérard Weisbuch, Thierry Faure: How can extremism prevail? A study on the relative agreement interaction model. *Journal of Artificial Societies and Social Simulation*, 5(4), 2002.

The article addresses the question whether it is possible to identify certain parameters and parameter values that endogeneously govern the distribution of opinions within a human population. A particular goal is to look for values that allow extreme opinions to dominate eventually.

The study is representative for a physics-oriented approach to complex networks:

- Methodologically, it employs *agent-based modelling*. Agent-based modelling uses simplified interaction models and simulations to explore a nonlinear dynamical behavior of complex systems. Agent-based modelling is applied when kinetic models involving differential equation systems are inappropriate, e.g., due to the number and the heterogeneity of variables.
- It explains a complex phenomenon in a *stylized*, metaphorical fashion.

Apart from the methodological perspective, the concrete, original research motivation for the study presented lies in the influence "green" farmers have attained in the farming population.

1.1 The Relative Agreement model

We consider the following formal scenario: A population of n agents is given. An agent i is characterized by two variables:

- opinion $x_i \in [-1, 1]$
- uncertainty $u_i \in [0, 1]$

Thus, the actual opinion of the agents ranges in her opinion segment

$$S_i =_{\text{def}} [x_i - u_i, x_i + u_i],$$

the size of which is $(x_i + u_i) - (x_i - u_i) = 2u_i$.

We suppose a directed model of influence where any two agents use a communication channel. Agent i locally communicates to agent j over her communication channel, possibly causing changes in opinion and uncertainty of agent j. In this situation, agent i is the influencer of agent j and agent j is the influenced of agent i.

The effect of a communicative influence is given by an update rule which is assumed to be the same for all interaction pairs of agents. Figure 1.1 describes a situation of an interaction pair (i, j).

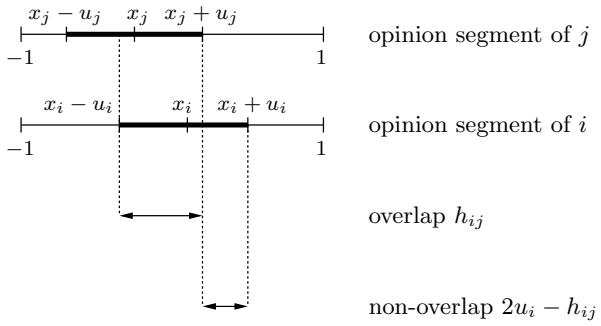


Figure 1.1: The Relative Agreement model

The update rule is based on the agreement along the opinion segments of agents i and j, i.e.,

$$h_{ij} - (2u_i - h_{ij}) = 2(h_{ij} - u_i),$$

in relation to the uncertainty of the influencer

$$\frac{2(h_{ij} - u_i)}{2u_i} = \frac{h_{ij}}{u_i} - 1.$$

The formal specification of the update rule is given by defining local transitions:

$$x_j \leftarrow x_j + \begin{cases} \mu\left(\frac{h_{ij}}{u_i} - 1\right)(x_i - x_j) & \text{if } h_{ij} \geq u_i \\ 0 & \text{otherwise} \end{cases}$$

$$u_j \leftarrow u_j + \begin{cases} \mu\left(\frac{h_{ij}}{u_i} - 1\right)(u_i - u_j) & \text{if } h_{ij} \geq u_i \\ 0 & \text{otherwise} \end{cases}$$

Here, μ is some decay constant, $0 < \mu < 1$.

1.2 Dyadic convergence

Proposition 1.1 *Let an interaction pair (i, j) be given. Let h_{ij} denote the overlap of the opinion segments of the actors i and j before interaction, and let h'_{ij} denote the overlap of the opinion segments of the actors i and j after interaction. Then, $h_{ij} \leq h'_{ij}$.*

Proof: Let (x_i, u_i) be the opinion/uncertainty pair of actor i, let (x_j, u_j) be the opinion/uncertainty pair of actor j. According to our update rule, if $h_{ij} \leq u_i$ then there are no changes, neither in the opinions nor in the uncertainties of both actors. That is, $h'_{ij} = h_{ij}$. So, let $h_{ij} > u_i$. Let x'_j denote actor j's opinion after interaction, and let u'_j denote actor j's uncertainty after interction. More specifically, we have

$$\begin{aligned} x'_j &= \left(1 - \mu\left(\frac{h_{ij}}{u_i} - 1\right)\right) x_j + \mu\left(\frac{h_{ij}}{u_i} - 1\right) x_i, \\ u'_j &= \left(1 - \mu\left(\frac{h_{ij}}{u_i} - 1\right)\right) u_j + \mu\left(\frac{h_{ij}}{u_i} - 1\right) u_i. \end{aligned}$$

The overlap h'_{ij} is given by

$$h'_{ij} = \min(x_i + u_i, x'_j + u'_j) - \max(x_i - u_i, x'_j - u'_j).$$

Note that $h_{ij} \leq 2u_i$. Thus, the update rules define convex combinations. By linearity, we easily examine the following cases:

1. Suppose $x_i + u_i \leq x_j + u_j$ and $x_i - u_i \geq x_j - u_j$. Therefore,

$$\begin{aligned} x_i + u_i &\leq x'_j + u'_j \leq x_j + u_j, \\ x_j - u_j &\leq x'_j - u'_j \leq x_i - u_i. \end{aligned}$$

   We obtain

$$\begin{aligned} h_{ij} &= x_i + u_i - (x_i - u_i), \\ h'_{ij} &= (x_i + u_i) - (x_i - u_i) = h_{ij} \end{aligned}$$

2. Suppose $x_i + u_i \leq x_j + u_j$ and $x_i - u_i \leq x_j - u_j$. Therefore,

$$\begin{aligned} x_i + u_i &\leq x'_j + u'_j \leq x_j + u_j, \\ x_i - u_i &\leq x'_j - u'_j \leq x_j - u_j. \end{aligned}$$

   We obtain

$$\begin{aligned} h_{ij} &= x_i + u_i - (x_j - u_j), \\ h'_{ij} &= (x_i + u_i) - (x'_j - u'_j) \geq (x_i + u_i) - (x_j - u_j) = h_{ij} \end{aligned}$$

3. Suppose $x_i + u_i \geq x_j + u_j$ and $x_i - u_i \leq x_j - u_j$. Therefore,

$$\begin{aligned} x_j + u_j &\leq x'_j + u'_j \leq x_i + u_i, \\ x_i - u_i &\leq x'_j - u'_j \leq x_j - u_j. \end{aligned}$$

   We obtain

$$\begin{aligned} h_{ij} &= x_j + u_j - (x_j - u_j), \\ h'_{ij} &= (x'_j + u'_j) - (x'_j - u'_j) \geq (x_i + u_i) - (x_j - u_j) = h_{ij} \end{aligned}$$

4. Suppose $x_i + u_i \geq x_j + u_j$ and $x_i - u_i \geq x_j - u_j$. Therefore,

$$\begin{aligned} x_j + u_j &\leq x'_j + u'_j \leq x_i + u_i, \\ x_j - u_j &\leq x'_j - u'_j \leq x_i - u_i. \end{aligned}$$

We obtain

$$\begin{aligned} h_{ij} &= x_j + u_j - (x_i - u_i), \\ h'_{ij} &= (x'_j + u'_j) - (x_i - u_i) \geq (x_j + u_j) - (x_i - u_i) = h_{ij} \end{aligned}$$

This proves the proposition. ■

Proposition 1.2 *Let the interaction pair (i, j) be given. For $k \in \mathbb{N}$, let $x_j^{(k)}$ be actor j's opinion after the k-th round of the directed interaction (i, j), and let $u_j^{(k)}$ be actor j's uncertainty after the k-th round of the directed interaction (i, j). Then,*

$$\lim_{k\to\infty} x_j^{(k)} = x_i, \qquad \lim_{k\to\infty} u_j^{(k)} = u_i.$$

Proof: We only prove the convergence in opinions. Since $h_{ij} \leq 2u_i$, we obtain as an upper bound on the opinion $x_j^{(k)}$ for $k \in \mathbb{N}_+$

$$x_j^{(k)} \leq (1 - \mu)x_j^{(k-1)} + \mu x_i,$$

and furthermore, by induction,

$$x_j^{(k)} \leq (1 - \mu)^k x_j + \left(1 - (1 - \mu)^k\right) x_i.$$

Hence, $\lim_{k\to\infty} x_j^{(k)} \leq x_i$. For the lower bound, we write

$$x_j^{(k)} = (1 - \mu A_{k-1})x_j^{(k-1)} + \mu A_{k-1} x_i,$$

where $A_k = \frac{h_{ij}^{(k)}}{u_i} - 1$ is the relative agreement after the k-th interaction. By Proposition 1.1, it holds that $A_k \leq A_{k+1}$ for all $k \in \mathbb{N}$. Thus, we can estimate

$$x_j^{(k)} \geq (1 - \mu A_0)x_j^{(k-1)} + \mu A_0 x_i,$$

and, again by induction,

$$x_j^{(k)} \geq (1 - \mu A_0)^k x_j + \left(1 - (1 - \mu A_0)^k\right) x_i,$$

Hence, $\lim_{k\to\infty} x_j^{(k)} \geq x_i$. This proves the proposition. ■

1.3 Extremists

In general populations of actors, it is not clear at all whether there is any convergence to a "stable" opinion/uncertainty pattern over several time steps. If unambiguous convergence is reachable, there are three important cases: convergence to the opinion poles, either positive or negative, or convergence to the middle. We are interest in studying convergence to extreme opinions.

Extremists are people with extreme opinions, i.e., opinions close to the boundaries measured by -1 and 1. Furthermore, the model of extremists within a population is based on two observations which are claimed to possess a certain anectodal evidence [7]:

1. "... people who have extreme opinions tend to be more convinced,"
2. "... people who have moderate initial opinions, often express a lack of knowledge (and uncertainty)."

A simplification of these observations can be incorporated into the Relative Agreement model as follows. Let u_e be the uncertainties of the extremists, supposed to be small and the same for all extremists. Let u be the (identical) uncertainty of the moderate. According to our observations, it holds that $u > u_e$. Then, the population can be initially decomposed into three classes corresponding to their opinion/uncertainty pairs:

1. *positive extremists*: $x_i \approx 1$, $u_i = u_e$
2. *negative extremists*: $x_i \approx -1$, $u_i = u_e$
3. *moderates*: $x_i \approx 0$, $u_i = u$

Let p_e denote the fraction of extremists, either positive or negative, in the population. Depending on the fraction of actors belonging to these classes, an extremism bias can be defined. Let p_+ be the fraction of positive extremists, and let p_- be the fraction of negative extremists. Then, the *extremism bias* δ is given as

$$\delta = \frac{p_+ - p_-}{p_+ + p_-}$$

The simulation works in two phases:

1. For initalization, (a) choose n opinions uniformly at random from $[-1, 1]$ and set all n uncertainties to u, (b) for the $p_+ \cdot n$ most positive opinions and $p_- \cdot n$ most negative opinons, the uncertainties are set to u_e.
2. Iteratively choose a pair (i, j) of agents and let agent i exert influence on agent j according to the specified update rule.

The stylized simulation results can be divided into three stable scenarios: central clustering, bipolarization, single polarization. The following figures show diagram schemes for each of the three scenarios together with parameter settings such that the described behavior can be observed. The x-axis codes for time, i.e., number of iterations per actor, and the y-axis codes for opinions. A trajectory of an actor's opinion over the course of time runs inside the region bounded by the drawn curves. Common parameters for all figures (and the simulations) are $n = 200$, $\mu = 0.5$, $\delta = 0$, and $u_e = 0.1$. The initial uncertainty parameter u is increased from figure to figure.

In Figure 1.2, the initial uncertainty of the moderates is $u = 0.4$. It is an example of central convergence. The majority of the moderate actors are not attracted by the extreme opinions.

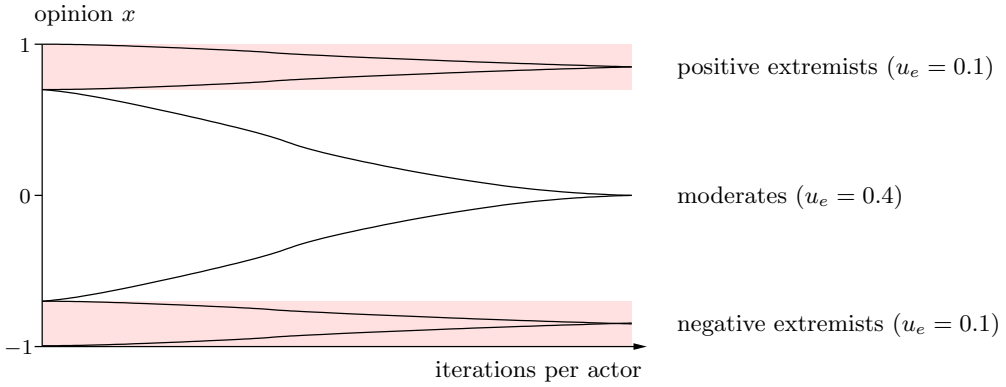


Figure 1.2: Scheme of central convergence.

In Figure 1.3, the initial uncertainty of the moderates is $u = 1.2$. It is an example of convergence to both extremes. The initially moderate actors split and become extremists.

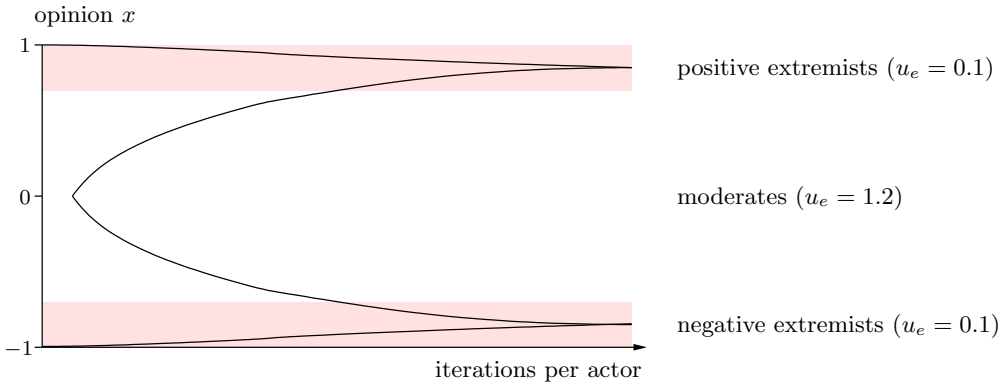


Figure 1.3: Scheme of bipolarization.

In Figure 1.4, the initial uncertainty of the moderates is $u = 1.4$. It is an example of convergence to one single extreme. In this case, the majority of the population is attracted by one of the extremes. This behavior can take place even when the number of

initial extremissts is the same at both extremes, which is claimed to have been a priori unexpected.

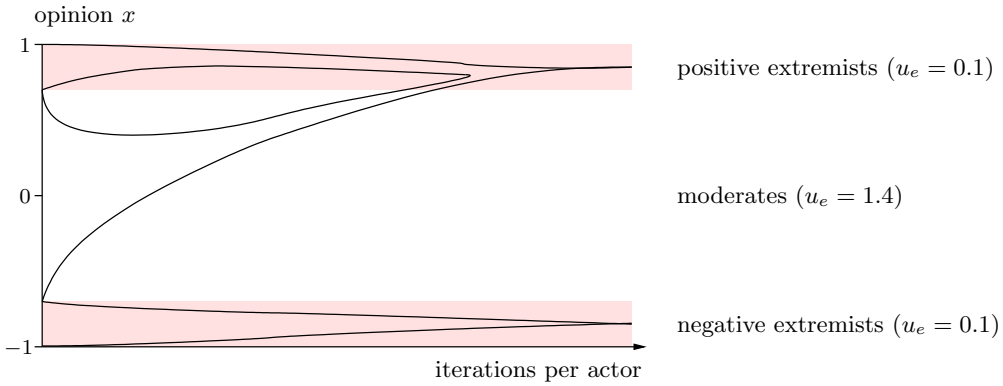


Figure 1.4: Scheme of single polarization.

Networks 2

2.1 Networks in a Static Perspective

2.1.1 Population

An *actor* x is any variable with values ranging in a set D called *attribute type*. If x takes on a value $z \in D$, then z is called *state* of x.

Definition 2.1 *1. A* population $X = \{x_1, \ldots, x_n\}$ *is a finite set of actors* $x_1, \ldots x_n$ *with attribute types* $D_1, \ldots D_n$.

2. A population $X = \{x_1, \ldots, x_n\}$ *with attribute types* $D_1 \ldots, D_n$ *is said to be* homogeneous *of attribute type* D *if* $D = D_1 = \cdots = D_n$.

In the following we restrict ourselves to homogeneous populations. Therefore we omit the word "homogeneous."

Let $X = \{x_1, \ldots x_n\}$ be a population with attribute type D. A *configuration* (assignment, interpretation) is a mapping $I : X \to D$, i.e., a configuration I assigns a state $I(x) \in D$ to each actor $x \in X$. As an alternative notation we also refer to a tuple $(z_1, \ldots, z_n) \in D^n$ such that $z_i = I(x_i)$ as a configuration.

2.1.2 Structure

The fundamental relation in network analysis is the *dyad*. A dyad relates two actors of a population. We use graph theory to describe dyadic structures of populations.

Definition 2.2 *Let* $X = \{x_1, \ldots, x_n\}$ *be a population of attribute type* D.

1. A structure *is a set* $E \subseteq X \times X$.

2. The elements of $X \times X$ *are called* dyads.

3. The elements of a structure E *are called* edges.

Structures can be directed, undirected, or mixed. They are allowed to have *annotations* (weights) with certain attribute types. That is, a structure E may be equipped with a weight function $w : E \to A$ where A is the attribute type.

2.1.3 Constraint

Constraints limit the set of possible configurations of populations.

Definition 2.3 *Let $X = \{x_1, \ldots, x_n\}$ be a population of attribute type D. A* constraint on X *is any relation $R \subseteq D^n$.*

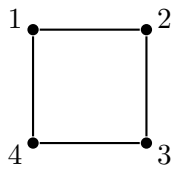
A configuration $I : X \to D$ is said to be *admissible* with respect to constraint R if and only if $(I(x_1), \ldots, I(x_n)) \in R$.

Example: We discuss constraints for some populations and structures.

- Let $X = \{1, 2, 3, 4\}$ be a population of type $D = \{0, 1\}$. Let $E = \mathrm{Circ}_4$ be a cycle containing the four actors. Define R to be the following constraint:

  $$R =_{\mathrm{def}} \{\ (0,0,0,0), (0,1,0,1), (1,0,1,0), (1,1,1,1)\ \}$$

  Then, constraint R corresponds to the set of solution of the following set of equations:

  $$\begin{aligned} x_1 &= x_4 \oplus x_1 \oplus x_2 \\ x_2 &= x_1 \oplus x_2 \oplus x_3 \\ x_3 &= x_2 \oplus x_3 \oplus x_4 \\ x_4 &= x_3 \oplus x_4 \oplus x_1 \end{aligned}$$

  Here, $\oplus$ denotes XOR. Clearly, x_1 does not directly depend on x_3 and x_2 does not directly depend on x_4. All other dependencies between the variable are represented by an edge in the structure. Later, we will call such an E interdependence structure for R.

- Again, let $X = \{1, 2, 3, 4\}$ be a population of type $D = \{0, 1\}$ and let $E = \mathrm{Circ}_4$ be a cycle containing the four actors. In contrast to the first example, suppose constraint R is given by the following set of equations:

  $$\begin{aligned} x_1 &= \overline{x_4} \wedge \overline{x_1} \wedge \overline{x_2} \\ x_2 &= \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \\ x_3 &= \overline{x_2} \wedge \overline{x_3} \wedge \overline{x_4} \\ x_4 &= \overline{x_3} \wedge \overline{x_4} \wedge \overline{x_1} \end{aligned}$$

  It is easily seen that in fact, R is empty. Thus, there is no admissible configuration.

- Let $X = \{x_1, \ldots, x_n\}$ be a population of type $D = \mathbb{R}$, and let E be unspecified. Then, R could be defined to be the convex polytope for the linear inequality constraints $A \cdot z \leq b$ such that $A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n$ and $z \geq 0$.

2.1.4 System

Definition 2.4 *Let D be an attribute type. A (homogeneous)* system *S is a triple (X, E, R) such that X is a population of attribute type D, E is a structure on X, and R is a constraint on X.*

Example: The triple

$$\big(\{1,2,3,4\}, \mathrm{Circ}_4, \big\{(0,0,0,0),(0,1,0,1),(1,0,1,0),(1,1,1,1)\big\}\big)$$

as in the example above is a system.

A *state* of S is any admissible configuration $I : X \to D$.

Note that if $R = D^n$, where n is the population size, then we omit R in the description of a system, ie., we write (X, E) instead of (X, E, D^n).

2.2 Networks in a Dynamical Perspective

2.2.1 Process

Definition 2.5 *A* process *P is any infinite sequence $(S_i)_{i\in\mathbb{N}}$ of systems S_i*

We also denote a process by $P = (X_i, E_i, R_i)_{i\in\mathbb{N}}$ when referring to the components of the systems.

Basically, we can identify at least three (non-excluding) important subtypes of processes:

- A process of type $(X_i, \emptyset, \emptyset)_{i\in\mathbb{N}}$ is called a *population process.*
- A process of type $(X_i, E_i, \emptyset)_{i\in\mathbb{N}}$ is called a *structure process.*
- A process of type $(X, E, R_i)_{i\in\mathbb{N}}$ is called a *state process.*

In this course, the focus is solely on *state processes.*

2.2.2 Trajectory

Systems running through a process can take several paths of state changes depending, e.g., on initial or environmental conditions. Such paths are called trajectories.

Definition 2.6 *Let $P = (X, E, R_i)_{i\in\mathbb{N}}$ be any state process with a population of size $\|X\| = n$. A* trajectory *is an infinite sequence $(I_j)_{j\in\mathbb{N}}$ of admissible configurations, i.e., $(I_j(x_1), \ldots, I_j(x_n)) \in R_j$ for all $j \in \mathbb{N}$.*

The set of possible trajectories of a process $(X, E, R_i)_{i\in\mathbb{N}}$ is $\bigtimes_{i=0}^{\infty} R_i$ and can thus be uncountable in general, even for a binary attribute type. An overwhelming fraction of these trajectories is certainly not realistic.

2.2.3 Global State Dynamic

A global (deterministic) state dynamic is a mechanism for selecting trajectories of a state process.

Definition 2.7 *Let $P = (X, E, R_i)_{i\in\mathbb{N}}$ be a state process of attribute type D and a population of size n.*

1. *A* global state dynamic *is a mapping $\mathbf{F} : D^n \times \mathbb{N}_+ \to D^n$.*
2. *A global state dynamic $\mathbf{F}$ is said to be* compatible *with P if $\mathbf{F}(R_0, t) \subseteq R_t$ for all $t \in \mathbb{N}$.*

A global state dynamic $\mathbf{F}$ produces the trajectory $(I, \mathbf{F}(I,1), \mathbf{F}(I,2), \dots)$ depending on the initial configuration I.

Example: Let $P_1 = (X, E, R_i^1)_{i\in\mathbb{N}}$ and $P_2 = (X, E, R_i^2)_{i\in\mathbb{N}}$ be state processes of type $D = \{0,1\}$, population $X = \{1,2,3\}$, structure $E = K^3$, and constraints

$$\begin{aligned} R_i^1 &=_{\text{def}} D^3, \\ R_i^2 &=_{\text{def}} \{\ (z_1,z_2,z_3) \in D^3 \mid 1 \le z_1+z_2+z_3 \le 2\ \}. \end{aligned}$$

Consider the following global state dynamics:

$$\mathbf{F}_1 \ : \ (z_1,z_2,z_3,t) \ \mapsto \ (1-z_1, 1-z_2, 1-z_3)$$

$$\mathbf{F}_2 \ : \ (z_1,z_2,z_3,t) \ \mapsto \ \begin{cases} (z_1,z_2,1-z_3) & \text{if } t \equiv 0 \bmod 3 \\ (1-z_1,z_2,z_3) & \text{if } t \equiv 1 \bmod 3 \\ (z_1,1-z_2,z_3) & \text{if } t \equiv 2 \bmod 3 \end{cases}$$

Note that $\mathbf{F}_1$ does not depend on parameter t. For $h = 5$, it produces the following set of trajectories (the rows of the table):

(z_1,z_2,z_3)	$\mathbf{F}_1(z_1,z_2,z_3,1)$	$\mathbf{F}_1(z_1,z_2,z_3,2)$	$\mathbf{F}_1(z_1,z_2,z_3,3)$	$\mathbf{F}_1(z_1,z_2,z_3,4)$
$(0,0,0)$	$(1,1,1)$	$(1,1,1)$	$(1,1,1)$	$(1,1,1)$
$(0,0,1)$	$(1,1,0)$	$(1,1,0)$	$(1,1,0)$	$(1,1,0)$
$(0,1,0)$	$(1,0,1)$	$(1,0,1)$	$(1,0,1)$	$(1,0,1)$
$(0,1,1)$	$(1,0,0)$	$(1,0,0)$	$(1,0,0)$	$(1,0,0)$
$(1,0,0)$	$(0,1,1)$	$(0,1,1)$	$(0,1,1)$	$(0,1,1)$
$(1,0,1)$	$(0,1,0)$	$(0,1,0)$	$(0,1,0)$	$(0,1,0)$
$(1,1,0)$	$(0,0,1)$	$(0,0,1)$	$(0,0,1)$	$(0,0,1)$
$(1,1,1)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$

It is obvious that $\mathbf{F}_1$ is compatible with both processes P_1 and P_2 (for each choice of h).

In contrast, $\mathbf{F}_2$ produces the following set of trajectories for $h = 5$.

(z_1, z_2, z_3)	$\mathbf{F}_2(z_1, z_2, z_3, 1)$	$\mathbf{F}_2(z_1, z_2, z_3, 2)$	$\mathbf{F}_2(z_1, z_2, z_3, 3)$	$\mathbf{F}_2(z_1, z_2, z_3, 4)$
$(0,0,0)$	$(1,0,0)$	$(0,1,0)$	$(0,0,1)$	$(1,0,0)$
$(0,0,1)$	$(1,0,1)$	$(0,1,1)$	$(0,0,0)$	$(1,0,1)$
$(0,1,0)$	$(1,1,0)$	$(0,1,0)$	$(0,1,1)$	$(1,1,0)$
$(0,1,1)$	$(1,1,1)$	$(0,1,1)$	$(0,1,0)$	$(1,1,1)$
$(1,0,0)$	$(0,1,1)$	$(1,0,0)$	$(1,0,1)$	$(0,0,0)$
$(1,0,1)$	$(0,0,1)$	$(1,0,1)$	$(1,0,0)$	$(0,0,1)$
$(1,1,0)$	$(0,1,0)$	$(1,0,0)$	$(1,1,1)$	$(0,1,0)$
$(1,1,1)$	$(0,1,1)$	$(1,0,1)$	$(1,1,0)$	$(0,1,1)$

We observe that $\mathbf{F}_2$ is compatible with P_1 but is compatible with P_2, since, e.g., $\mathbf{F}_2(0,1,1,1) = (1,1,1) \notin R_2^2$ although $(0,1,1) \in R_1^2$.

2.2.4 Local State Dynamic

Definition 2.8 *Let $P_1 = (X, E, R_i)_{i\in\mathbb{N}}$ be a state process of attribute type D and population size n.*

1. *A* local transition *on X is a mapping $f : D^n \to D$*
2. *An* update schedule *on X is a mapping $\alpha : \mathbb{N}_+ \to \mathcal{P}(X)$*
3. *A* local state dynamic *an X is a pair (F, α) such that $F = \{f_1, \ldots, f_n\}$ is a set of local transitions on X, where f_i is associated with actor $x_i \in X$, and α is an update schedule on X.*

Example: Consider again P_1 and the following two local state dynamics, given by the same set $F = \{f_1, f_2, f_3\}$ of local transitions such that for each $i \in \{1,2,3\}, z_1, z_2, z_3 \in \{0,1\}$

$$f_i \;:\; \{z_1, z_2, z_3\} \mapsto \begin{cases} z_i & \text{if } z_1 + z_2 + z_3 = 1 \\ 1 - z_i & \text{otherwise} \end{cases}$$

and the following two update schedules:

$$\alpha_1 \;:\; \mathbb{N}_+ \to \mathcal{P}(X) \;:\; t \mapsto \{1,2,3\}$$

$$\alpha_2 \;:\; \mathbb{N}_+ \to \mathcal{P}(X) \;:\; t \mapsto \begin{cases} \{3\} & \text{if } t \equiv 0 \bmod 3 \\ \{2\} & \text{if } t \equiv 1 \bmod 3 \\ \{1\} & \text{if } t \equiv 2 \bmod 3 \end{cases}$$

Schedule α_1 is called *synchronous* as the schedule allows all actors to update their states in one step in parallel. In contrast, schedule α_2 periodically selects

exactly one actor in each step for updating. Such a schedule is called *sequential*. Sequential updates are usually described by sequences of actors. In our example, α_2 can be written as $(2, 1, 3, 2, 1, 3, 2, 1, 3, \dots)$ or, due to its periodic structure, simply by the permutation (tuple) $(2, 1, 3)$.

A local state dynamic induces a global state dynamic.

Definition 2.9 *Let $P = (X, E, R_i)_{i\in\mathbb{N}}$ be a state process of attribute type D and population size n. Let (S, α) be a local state dynamic on X.*

1. *For each actor $x_i \in X$ and for each subset $U \subseteq X$ of actors,* activity function $\varphi_i[U]$ *is defined for configuration $\vec{z} = (z_1, \dots, z_n) \in D^n$ by*

$$\varphi_i[U](\vec{z}) =_{\text{def}} \begin{cases} f_i(z_1, \dots, z_n) & \text{if } x_i \in U \\ z_i & \text{if } x_i \notin U \end{cases}$$

2. *For each set $U \subseteq X$, the* global transition (function) $\mathbf{F}_S[U] : D^n \to D^n$ *is defined for configuration $\vec{z} = (z_1, \dots, z_n)$ by*

$$\mathbf{F}_S[U](\vec{z}) =_{\text{def}} \big(\varphi_1[U](\vec{z}), \dots, \varphi_n[U](\vec{z})\big)$$

3. *The global state dynamic $\mathbf{F}_{(S,\alpha)} : D^n \times \mathbb{N}_+ \to D^n$* induced by the local state dynamic *(S, α) is defined for $t \in \mathbb{N}_+$ by*

$$\mathbf{F}_{(S,\alpha)}(\cdot, t) =_{\text{def}} \left(\prod_{k=1}^{t} \mathbf{F}_S[\alpha(k)]\right)(\cdot),$$

i.e., by the composition of global transitions specified by the update schedule.

Note that $f \cdot g$ is the function defined by $(f \cdot g)(x) = g(f(x))$. The following shall elucidate the above definition in detail. For $t = 3$ and $\vec{z} \in D^n$, we have

$$\begin{aligned} \mathbf{F}_{(S,\alpha)}(\vec{z}, 3) &= \left(\prod_{k=1}^{3} \mathbf{F}_S[\alpha(k)]\right)(\vec{z}) = \left(\mathbf{F}_S[\alpha(1)] \cdot \prod_{k=2}^{3} \mathbf{F}_S[\alpha(k)]\right)(\vec{z}) \\ &= \left(\prod_{k=2}^{3} \mathbf{F}_S[\alpha(k)]\right)\big(\mathbf{F}_S[\alpha(1)](\vec{z})\big) = \Big(\mathbf{F}_S[\alpha(2)] \cdot \mathbf{F}_S[\alpha(3)]\Big)\big(\mathbf{F}_S[\alpha(1)](\vec{z})\big) \\ &= \mathbf{F}_S[\alpha(3)]\big(\mathbf{F}_S[\alpha(2)]\big(\mathbf{F}_S[\alpha(1)](\vec{z})\big)\big) \end{aligned}$$

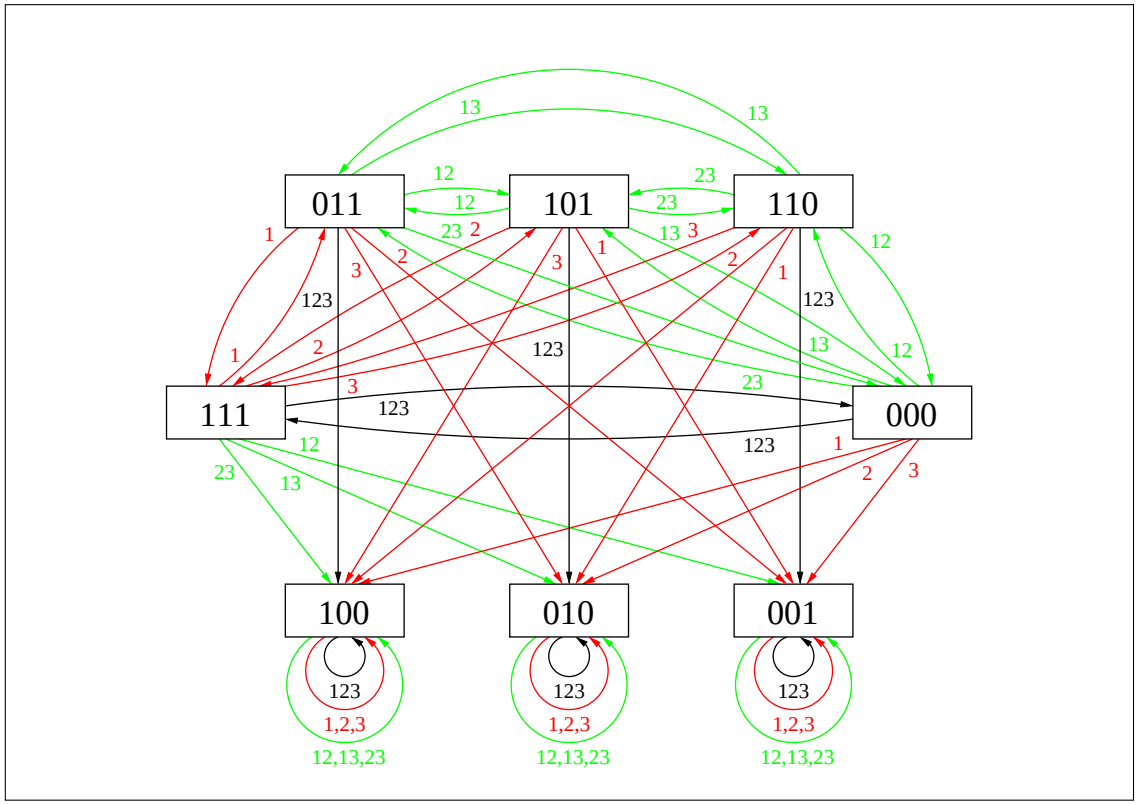


Figure 2.1: Global transition function

Also notice that activity functions and global transition functions do not depend on schedules.

Example: We continue the example of local state dynamics discussed above. Let $U_1 = \{1, 2\}$ and $U_2 = \{1, 2, 3\}$ be subsets of populations. Then we obtain the following activity functions:

- $\varphi_1[U_1] = f_1$, $\varphi_2[U_1] = f_2$, and $\varphi_3[U_1] = \mathrm{id}$;
- $\varphi_1[U_2] = f_1$, $\varphi_2[U_2] = f_2$, and $\varphi_3[U_2] = f_3$.

The global transition function looks as follows:

- $\mathbf{F}_S[U_1](z_1, z_2, z_3) = (f_1(z_1, z_2, z_3), f_2(z_1, z_2, z_3), z_3)$; concrete function values are, e.g.,

$$\begin{aligned} \mathbf{F}_S[U_1](1,1,1) &= (0,0,1) \\ \mathbf{F}_S[U_1](1,0,1) &= (0,1,1) \\ \mathbf{F}_S[U_1](0,0,1) &= (0,0,1) \end{aligned}$$

- $\mathbf{F}_S[U_2](z_1, z_2, z_3) = (f_1(z_1, z_2, z_3), f_2(z_1, z_2, z_3), f_3(z_1, z_2, z_3)$; concrete values are, e.g.,

$$\begin{aligned} \mathbf{F}_S[U_2](1,1,1) &= (0,0,0) \\ \mathbf{F}_S[U_2](0,0,0) &= (1,1,1) \end{aligned}$$

Figure 2.1 visualizes the global transition functions for $F = \{f_1, f_2, f_3\}$ completely. The global state dynamic induced by (S, α_1) is as follows:

(z_1, z_2, z_3)	$\{1,2,3\}$	$\{1,2,3\}$	$\{1,2,3\}$	$\{1,2,3\}$	...
(0,0,0)	(1,1,1)	(0,0,0)	(1,1,1)	(0,0,0)	...
(0,0,1)	(0,0,1)	(0,0,1)	(0,0,1)	(0,0,1)	...
(0,1,0)	(0,1,0)	(0,1,0)	(0,1,0)	(0,1,0)	...
(0,1,1)	(1,0,0)	(1,0,0)	(1,0,0)	(1,0,0)	...
(1,0,0)	(1,0,0)	(1,0,0)	(1,0,0)	(1,0,0)	...
(1,0,1)	(0,1,0)	(0,1,0)	(0,1,0)	(0,1,0)	...
(1,1,0)	(0,0,1)	(0,0,1)	(0,0,1)	(0,0,1)	...
(1,1,1)	(0,0,0)	(1,1,1)	(0,0,0)	(0,0,0)	...

That is, (S, α_1) can generate oscillating trajectories.

The global state dynamic induced by (F, α_2) is as follows:

(z_1, z_2, z_3)	$\{2\}$	$\{1\}$	$\{3\}$	$\{2\}$	...
(0,0,0)	(0,1,0)	(0,1,0)	(0,1,0)	(0,1,0)	...
(0,0,1)	(0,0,1)	(0,0,1)	(0,0,1)	(0,0,1)	...
(0,1,0)	(0,1,0)	(0,1,0)	(0,1,0)	(0,1,0)	...
(0,1,1)	(0,0,1)	(0,0,1)	(0,0,1)	(0,0,1)	...
(1,0,0)	(1,0,0)	(1,0,0)	(1,0,0)	(1,0,0)	...
(1,0,1)	(1,1,1)	(0,1,1)	(0,1,0)	(0,1,0)	...
(1,1,0)	(1,0,0)	(1,0,0)	(1,0,0)	(1,0,0)	...
(1,1,1)	(1,0,1)	(0,0,1)	(0,0,1)	(0,0,1)	...

We observe that any further update does not change the resulting configurations. Thus, (S, α_2) is highly stable as each trajectory reaches a fixed point.

2.3 Network Equivalence

2.3.1 Functional Equivalence

Let (X, E, R) be any state process with population size $n = \|X\|$ and attribute type D. Let L be a set of local transitions with interdependence structure E.

A schedule $\alpha : \mathbb{N}_+ \to X(= \mathcal{P}_1(X))$ is said to be *sequential* if and only if

1. $\alpha(t) = a(t + k \cdot n)$ for all $k \in \mathbb{N}$,
2. α is surjective,
3. $\alpha(t_1) \neq \alpha(t_2)$ for all $t_1, t_2 \in \{1, \ldots, n\}$ such that $t_1 \neq t_2$.

A sequential schedule can be described by a permutation $\pi : X \to X$. A local state dynamic (L, α) over a sequential update schedule α can be written as (L, π), where π is the permutation that generates α in the sense of the definition above.

Definition 2.10 *Let L be a set of local transitions on X, $\|X\| = n$. Let $\pi, \pi' : X \to X$ be a permutation on X. Then, π and π' are said to be* functionally equivalent*, $\pi \equiv_{\mathrm{f}} \pi'$ in symbols, if and only if*

$$\mathbf{F}_{(L,\pi)}(\cdot, k \cdot n) = \mathbf{F}_{(L,\pi')}(\cdot, k \cdot n)$$

for all $k \in \mathbb{N}_+$.

Clearly, it is logically equivalent to require the equation $\mathbf{F}_{(L,\pi)}(\cdot, n) = \mathbf{F}_{(L,\pi')}(\cdot, n)$ for π and π' to be functionally equivalent.

A deeper analysis of the notion of functional equivalence is based on update orders given by permutations. Let X be a population, without loss of generality, $X = \{1, \ldots, n\}$, let E be an interdependence structure, and let S_X denote the symmetric group of X, i.e., the set all permutations $\pi : X \to X$. For different $\pi, \pi' \in S_X$, we say that π and π' are *adjacent* (with respect to (X, E)) if and only if there is a k such that $\{\pi(k), \pi(k+1)\} \notin E$ and $\pi(i) = \pi'(i)$ for $i \notin \{k, k+1\}$. In other words, π and π' are adjacent with respect to (X, E) iff π' is obtained by swapping consecutive elements, not neighbored in E, in the permutation order of π.

Proposition 2.11 *Let $\pi, \pi' \in S_X$ be adjacent with respect to (X, E). Let k be such that $\{\pi(k), \pi(k+1)\} \notin E$ and $\pi(i) = \pi'(i)$ for all $i \notin \{k, k+1\}$. Then,*

$$\mathbf{F}_L[\pi(k)] \cdot \mathbf{F}_L[\pi(k+1)] = \mathbf{F}_L[\pi(k+1)] \cdot \mathbf{F}_L[\pi(k)]$$

for all sets L of local transition functions with interdependence structure E.

Proof: Since $\{\pi(k), \pi(k+1)\} \notin E$, $\pi(k)$ is fictive in $f_{\pi(k+1)}$ and $\pi(k+1)$ is fictive in $f_{\pi(k)}$. That is, we can replace the $\pi(k)$-th argument in $f_{\pi(k+1)}$ as well as the $\pi(k+1)$-st argument in $f_{\pi(k)}$ arbitrarily. Suppose L is a set of local transitions. Let $\vec{z} = (z_1, \ldots, z_n)$

be any configuration. Assume that, without loss of generality, $\pi(k) < \pi(k+1)$. Then,

$$\begin{aligned}
&\big(\mathbf{F}_L[\pi(k)] \cdot \mathbf{F}_L[\pi(k+1)]\big)(\vec{z}) \\
&\quad = \mathbf{F}_L[\pi(k+1)]\big(\mathbf{F}_L[\pi(k)](\vec{z})\big) \\
&\quad = \mathbf{F}_L[\pi(k+1)](z_1, \ldots, f_{\pi(k)}(z_1, \ldots, z_n), \ldots, z_n) \\
&\quad = (z_1, \ldots, f_{\pi(k)}(z_1, \ldots, z_n), \ldots, f_{\pi(k+1)}(z_1, \ldots, f_{\pi(k)}(z_1, \ldots, z_n), \ldots, z_n), \ldots, z_n) \\
&\quad = \mathbf{F}_L[\pi(k)](z_1, \ldots, f_{\pi(k+1)}(z_1, \ldots, z_n), \ldots, z_n) \\
&\quad = \mathbf{F}_L[\pi(k)]\big(\mathbf{F}_[\pi(k+1)](\vec{z})\big) \\
&\quad = \big(\mathbf{F}_L[\pi(k+1)] \cdot \mathbf{F}_L[\pi(k)]\big)(\vec{z})
\end{aligned}$$

This proves the proposition. ■

Update graph. The *update graph* (or *ugraph*, for short) $U = U(X, E)$ consists of vertex set S_X and edge set $\{(\pi, \pi') \mid \pi \text{ and } \pi' \text{ are adjacent}\}$

Example: We determine update graph for Circ_4, with population $X = \{1, 2, 3, 4\}$.

Based on the update graph, we define an equivalence relation on S_X with respect to $U = U(X, E)$:

$$\pi \sim_U \pi' \Longleftrightarrow_{\text{def}} \pi \text{ and } \pi' \text{ are connected by a path in } U$$

Proposition 2.12 *Let $G = (X, E)$ be an undirected graph, $\|X\| = n$. Let $\pi, \pi' \in S_X$ and let $U = U(X, E)$ be the update graph. If $\pi \sim_U \pi'$ then*

$$\mathbf{F}_{(L,\pi)}(\cdot, n) = \mathbf{F}_{(L,\pi')}(\cdot, n)$$

for all sets L of local transition functions with interdepence structure E.

Proof: The proof is by induction on the distance d between permutations in the update graph $U = U(X, E)$. The distance $d_U(\pi, \pi')$ is defined to be the length of a shortest path from π to π' in U.

- *Base of induction*: Let $d = 0$. So, $d_U(\pi, \pi') = 0$, i.e., $\pi = \pi'$.

- *Induction step*: Let $d_U(\pi,\pi') = d_U(\pi',\pi) = d > 0$. Let $(\pi_0,\ldots,\pi_{d-1},\pi_d)$ be a shortest path in U such that $\pi_0 = \pi'$ and $\pi_d = \pi$. It follows that π_{d-1} and π_d are adjacent with respect to U. Thus, there is a k such that $\{\pi(k),\pi(k+1)\} \notin E$ and $\pi(i) = \pi_{d-1}(i)$ for all $i \notin \{k,k+1\}$. We obtain for any set L of local transition functions and $\vec{z} \in D^n$

$$
\begin{aligned}
&\mathbf{F}_{(L,\pi)}(\vec{z},n) \\
&= \left(\prod_{j=1}^{n} \mathbf{F}_L[\pi(j)]\right)(\vec{z}) \\
&= \left(\prod_{j=1}^{k-1} \mathbf{F}_L[\pi(j)] \cdot \mathbf{F}_L[\pi(k)] \cdot \mathbf{F}_L[\pi(k+1)] \cdot \prod_{j=k+2}^{n} \mathbf{F}_L[\pi(j)]\right)(\vec{z}) \\
&= \left(\prod_{j=1}^{k-1} \mathbf{F}_L[\pi(j)] \cdot \mathbf{F}_L[\pi(k+1)] \cdot \mathbf{F}_L[\pi(k)] \cdot \prod_{j=k+2}^{n} \mathbf{F}_L[\pi(j)]\right)(\vec{z}) \\
&\qquad\qquad \text{(by Proposition 2.11)} \\
&= \left(\prod_{j=1}^{k-1} \mathbf{F}_L[\pi_{d-1}(j)] \cdot \mathbf{F}_L[\pi_{d-1}(k)] \cdot \mathbf{F}_L[\pi_{d-1}(k+1)] \cdot \prod_{j=k+2}^{n} \mathbf{F}_L[\pi_{d-1}(j)]\right)(\vec{z}) \\
&= \left(\prod_{j=1}^{n} \mathbf{F}_L[\pi_{d-1}(j)]\right)(\vec{z}) \\
&= \mathbf{F}_{(L,\pi_{d-1})}(\vec{z},n) \\
&= \mathbf{F}_{(L,\pi')}(\vec{z},n) \qquad\qquad \text{(by induction assumption)}
\end{aligned}
$$

This proves the proposition. ∎

We consider the equivalence class $[\pi]_U$ of a permutation π with respect to $U = U(X,E)$, i.e.,

$$[\pi]_U =_{\text{def}} \{\pi' \mid \pi \sim_U \pi'\},$$

together with the quotient set with respect to the equivalence relation $\sim_U$

$$S_X/\sim_U = \{[\pi]_U \mid \pi \in S_X\}.$$

Proposition 2.13 *Let $G = (X,E)$ be an undirected graph and let $U = U(X,E)$ be the update graph. Then, there exists a bijective mapping*

$$f_G \;:\; S_X/\sim_U \;\to\; \mathrm{Acyc}(G),$$

where $\mathrm{Acyc}(G)$ is the set of all acyclic orientations of G.

Proof: We first construct an appropriate mapping $\tilde{f}_G : S_X \to \mathrm{Acyc}(G)$. Any permutation $\pi \in S_X$ induces a linear ordering $\leq_\pi$ on X by

$$i \leq_\pi j \iff_{\mathrm{def}} \pi(i) \leq \pi(j).$$

Any linear ordering $\leq_\pi$ on X induces an acyclic orientation: for each $\{i, j\} \in E$ set

$$i \to j \iff_{\mathrm{def}} i <_\pi j$$

Let $\tilde{f}_G$ map each permutation to the according orientation. We have to argue that $\tilde{f}_G(\pi) = \tilde{f}_G(\pi')$ for $\pi \sim_U \pi'$. It suffices to show $\tilde{f}_G(\pi) = \tilde{f}_G(\pi')$ for adjacent permutations π, π' (proof of the general case is then by induction): If π and π' are adjacent, they differ in exactly two consecutive entries not connected by an edge in E. Thus, $\tilde{f}_G(\pi) = \tilde{f}_G(\pi')$.

Now, define $f_G : S_X/\sim_U \to \mathrm{Acyc}(G)$ by $f_G([\pi]_U) =_{\mathrm{def}} \tilde{f}_G(\pi)$. Observe that f_G is injective (exercise!). It remains to show that f_G is surjective. Consider an acyclic orientation of G. For vertex $i \in X$ define the rank of i as follows:

$$\mathrm{rank}(i) \quad =_{\mathrm{def}} \quad \text{length of a longest directed path to } i \text{ (with respect to the given acyclic orientation)}$$

We should note that $\mathrm{rank}(i) = \mathrm{rank}(j)$ implies $\{i, j\} \notin E$ for $i \neq j$. We define

$$H =_{\mathrm{def}} \{h \mid \mathrm{rank}^{-1}(h) \neq \emptyset\}$$

and for $h \in H$

$$\mathrm{rnk}^{-1}(h) =_{\mathrm{def}} (i_1, \ldots, i_{m_h}),$$

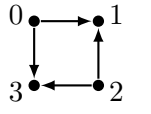
where $\mathrm{rank}(i_j) = h$ and $i_j < i_k$ for $j < k$. Furthermore, consider

$$\big[\big(\mathrm{rnk}^{-1}(0), \mathrm{rnk}^{-1}(1), \ldots, \mathrm{rnk}^{-1}(t)\big)\big]_U$$

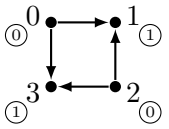
with $t = \max H$. Then, clearly, f_G maps $\big[\big(\mathrm{rnk}^{-1}(0), \ldots, \mathrm{rnk}^{-1}(t)\big)\big]_U$ to the given orientation. Thus, f_G is surjective. Hence, f_G is bijective. ■

Example: Consider Circ_4

$$[(0,2,1,3)]_U \qquad \longmapsto \qquad$$



$$[(\underbrace{0,2}_{\mathrm{rnk}^{-1}(0)}, \underbrace{1,3}_{\mathrm{rnk}^{-1}(1)})]_U \qquad \longmapsto \qquad$$



Proposition 2.14 *For any undirected graph $G = (X, E)$, $\|X\| = n$, and any set L of local transition functions with interdepence structure E,*

$$\| \left\{ \mathbf{F}_{(L,\pi)}(\cdot, n) \mid \pi \in S_X \right\} \| \leq \|\mathrm{Acyc}(G)\|;$$

and the bound is sharp.

Proof: Using Proposition 2.12 and Proposition 2.13, we obtain the following:

$$\left\| \left\{ \mathbf{F}_{(L,\pi)}(\cdot, n) \mid \pi \in S_X \right\} \right\| \leq \|\{[\pi]_U \mid \pi \in S_X\}\| = \|S_X / \sim_U\| = \|\mathrm{Acyc}(G)\|$$

Sharpness is left as an exercise. This proves the proposition. ■

> **Example:** It holds that $\|\mathrm{Acyc}(\mathrm{Circ}_n)\| = 2^n - 2$, since only two of the 2^n possible orientations of Circ_n are not acyclic. Thus, there are at most $2^n - 2$ essentially different local state dynamics on Circ_n.

How to compute $\|\mathrm{Acyc}(G)\|$?

Chromatic polynomial. Let $G = (V, E)$ be an undirected graph. A *vertex coloring* with k colors $1, \ldots, k$ is a mapping $f : V \to \{1, \ldots, k\}$ such that $f(u) \neq f(v)$ if $\{u, v\} \in E$. Define $P_G(k)$ to be the number of different vertex colorings with k colors of G. The pausible choices for the number of colors are $0, 1, \ldots, n$. Thus, we know the function values of P_G for $n + 1$ arguments. Hence, there is a uniquely determined normal polynomial (i.e., the leading coefficient in the expanded form of the polynomial is 1) of degree n which takes on these specified function values. We identify P_G with this polynomial, and we call $P_G(x)$ the *chromatic polynomial* of graph G.

> **Example:** Let $G = K^n$. It holds that $P_G(k) = 0$ for $k \in \{0, 1, \ldots, n-1\}$. Moreover, $P_G(n) = n!$. Thus, the chromatic polynomial of G is given by
>
> $$P_G(x) = \prod_{j=0}^{n-1} (x - j).$$

Lemma 2.15 *Let G, H be undirected graphs.*

1. *If G is a one-vertex graph, $P_G(x) = x$.*
2. *$P_{G \oplus H}(x) = P_G(x) \cdot P_H(x)$*
3. *$P_G(x) = P_{G-e}(k) - P_{G/e}(x)$*

Example: Let T be a tree with n vertices. Let u be an arbitrary leaf of T and $e = \{u, v\}$ be the edge connecting u with T. Then, it holds

$$\begin{aligned} P_T(x) &= P_{T-e}(x) - P_{T/e}(x) \\ &= P_{T'}(x) \cdot x - P_{T''(x)} \end{aligned}$$

Here, T' is a tree with $n-1$ vertices, T'' is a tree with $n-1$ vertices. Actually, $T' \simeq T''$. We conclude

$$P_T(x) = P_{T'}(x) \cdot (x-1).$$

By iteration, we obtain $P_T(x) = x(x-1)^{n-1}$.

Thus, each tree with n vertices has the same chromatic polynomial independent of its structure. Moreover, a graph G with n vertices is a tree if and only if $P_G(x) = x(x-1)^{n-1}$.

Lemma 2.16 *Let G be an undirected graph. Suppose there are graphs G_1, G_2 such that $G = G_1 \cup G_2$ and $G_1 \cap G_2 = K^n$. Then,*

$$P_G(x) = \frac{P_{G_1}(x) \cdot P_{G_2}(x)}{P_{K^n}(x)}$$

Proof: Each vertex coloring f of G corresponds to exactly one pair (f_1, f_2) of colorings of G_1 and G_2 which are identical on K^n. So, let f_1 be a k-coloring of G_1. Then, there are $P_{G_2}(k)/P_{K^n}(k)$ k-colorings of G_2 which are identical on K^n with f_1. This proves the lemma. ■

Example: We want to compute, once more, the chromatic polynomial for K^n. We start with the following recursion:

$$\begin{aligned} P_{K^n}(x) &= P_{K^n-e}(x) - P_{K^n/e}(x) \\ &= \frac{P_{K^{n-1}}(x)^2}{P_{K^{n-2}}(x)} - P_{K^{n-1}}(x) \\ &= \frac{P_{K^{n-1}}(x)}{P_{K^{n-2}}(x)} \; (P_{K^{n-1}}(x) - P_{K^{n-2}}(x)) \end{aligned}$$

By induction we can prove that $P_{k^n}(x) = x^{\underline{n}}$:

- *Base of induction*: We have two case here, $n \in \{1, 2\}$: $P_{K^1}(x) = x = x^{\underline{1}}$ and $P_{K^2}(x) = x(x-1) = x^{\underline{2}}$.
- *Induction step*: For $n > 2$, we have

$$\begin{aligned} P_{K^n}(x) &= \frac{x^{\underline{n-1}}}{x^{\underline{n-2}}} \cdot \left(x^{\underline{n-1}} - x^{\underline{n-2}}\right) && \text{(by induction assumption)} \\ &= (x - (n-1) + 1) \cdot x^{\underline{n-2}} \cdot ((x - (n-1) + 1) - 1) \\ &= x^{\underline{n-2}} \cdot (x - (n-2)) \cdot (x - (n-1)) \\ &= x^{\underline{n}} \end{aligned}$$

We give a different interpretation of $P_G(x)$.

Proposition 2.17 *Let $G = (V, E)$ be an undirected graph. Then, $P_G(k)$ is equal to the numbers of pairs (f, O) where $f : V \to \{1, \ldots, k\}$ and O is an orientation of G such that*

1. *the orientation O is acyclic,*
2. *if $u \to v$ in orientation O then $f(u) > f(v)$.*

Proof: Consider a pair (f, O) satisfying (i), (ii). From (ii) it follows that $f(u) \neq f(v)$ for $\{u, v\} \in E$. Thus, f is a vertex coloring with k colors. Moreover, (ii) implies (i). Conversely, if f is a vertex coloring with k colors then f defines a unique acyclic orientation O by $u \to v$ if and only if $f(u) > f(v)$. Hence, the number of allowed pairs (f, O) is the number of vertex colorings with colors $1, \ldots, k$ and is, thus, $P_G(k)$. ■

Proposition 2.17 suggests the following modification: Let $G = (V, E)$ be an undirected graph and let $k \in \{1, \ldots, n\}$ where $n = \|V\|$. Define $\overline{P}_G(k)$ to be the number of pairs (f, O) where $f : V \to \{1, \ldots, k\}$ and O is an orientation of G such that:

1. the orientation O is acyclic,
2. if $u \to v$ in orientation O then $f(u) > f(v)$.

We say that the function f is compatible with O if f satisfies the second conditions.

Lemma 2.18 *Let G, H be undirected graphs.*

1. *If G is one-vertex graph then $\overline{P}_G(x) = x$.*
2. *$\overline{P}_{G \oplus H}(x) = \overline{P}_G(x) \cdot \overline{P}_H(x)$*
3. *$\overline{P}_G(x) = \overline{P}_{G-e}(x) + \overline{P}_{G/e}(x)$ for any $e \in E$*

Proof: The first two statements are obvious.

In order to show the third statement, let $f : V \to \{1, \ldots, k\}$ be a mapping and let O be an acyclic orientation of $G-e$ compatible with f, where $e = \{u, v\} \in E$. Let O_1 be the orientation of G obtained by adjoining $u \to v$ to O, and O_2 that is obtained by adjoining $v \to u$ to O. We show that for each pair (f, O) exactly one of O_1 and O_2 is an acyclic orientation compatible with f, except for $\overline{P}_{G/e}(k)$ of the pairs, in which case both O_1 and O_2 are acyclic orientations compatible with f. Thus, $\overline{P}_{G-e}(k) = \overline{P}_G(k) - \overline{P}_{G/e}(k)$. We consider the following three cases:

- If $f(u) > f(v)$ then O_2 is not compatible with f while O_1 is compatible. Moreover, O_1 is acyclic, since if $u \to v \to w_1 \to w_2 \to \cdots \to u$ were a directed cycle in O_1, we would have $f(u) > f(v) \geq f(w_1) \geq f(w_2) \geq \cdots \geq f(u)$, which is a contradiction.

- If $f(u) < f(v)$ then we can argue symmetrically to the first case.
- If $f(u) = f(v)$, both O_1 and O_2 are compatible with f. Then, at least one of them is acyclic; if not: O_1 contains a cycle $u \to v \to w_1 \to w_2 \to \cdots \to u$ and O_2 contains a cycle $v \to u \to w'_1 \to w'_2 \to \cdots \to v$. Hence, O contains a cycle $v \to w_1 \to w_2 \to \cdots \to u \to w'_1 \to w'_2 \to \cdots \to u$ which is not possible.

It remains to prove that O_1 and O_2 are acyclic for exactly $\overline{P}_{G/e}(k)$ pairs (f, O) with $f(u) = f(v)$. Define $\Phi(f, O) =_{\text{def}} (f', O')$ such that $f' : V(G/e) \to \{1, \ldots, k\}$ (note that $f(u) = f(v)$) and O' is an acyclic orientation of G/e compatible with f'. Let z be the vertex obtained by identifying u and v. Define f' to be the following function:

$$f'(w) =_{\text{def}} \begin{cases} f(w) & \text{if } w \in V \setminus \{u, v\} \\ f(u) & \text{if } w = z \end{cases}$$

Define O' by $w_1 \to w_2$ in O' if and only if $w_1 \to w_2$ in O. Then, Φ is a bijection. This proves the proposition. ■

Theorem 2.19 (Stanley 1973) *For each graph $G = (V, E)$ such that $\|V\| = n$,*

$$\overline{P}_G(x) = (-1)^n P_G(-x).$$

Proof: Using the recursive rules according to Lemma 2.18 and Lemma **??**, we prove the statement by induction on the number n of vertices.

- *Base of induction*: Let $n = 1$. Then, $\overline{P}_G(x) = x = (-1)^1(-x) = (-1)^1 P_G(-x)$.
- *Induction step*: Suppose $n > 1$. Again, we argue inductively, in this case however, on the number of edges. For the base of induction, let G be the empty graph on n vertices. Then, $\overline{P}_G(x) = x^n = (-1)^n(-x)^n = (-1)^n P_G(-x)$. For the induction step, suppose $\|E\| \geq 1$. Then, for some edge $e \in E$

$$\begin{aligned} \overline{P}_G(x) &= \overline{P}_{G-e}(x) + \overline{P}_{G/e}(x) \\ &= (-1)^n \; P_{G-e}(-x) + (-1)^{n-1} \; P_{G/e}(-x) \\ &= (-1)^n \; \big(P_{G-e}(-x) - P_{G/e}(-x)\big) \\ &= (-1)^n \; P_G(-x) \end{aligned}$$

This proves the theorem. ■

Corollary 2.20 $\|\text{Acyc}(G)\| = (-1)^n \; P_G(-1)$.

Proof: It holds that $\|\mathrm{Acyc}(G)\| = \overline{P}_G(1) = (-1)^n \; P_G(-1)$. ∎

Example: We want to compute $\|\mathrm{Acyc}(\mathrm{Circ}_n)\|$ for $n \geq 3$. First, we prove that $P_{\mathrm{Circ}_n}(x) = (x-1)^n + (-1)^n(x-1)$ by induction on $n \geq 3$.

- *Base of induction*: For $n = 3$, we calculate

$$\begin{aligned} P_{\mathrm{Circ}_3}(x) &= x(x-1)(x-2) \\ &= x^3 - 3x^2 + 2x \\ &= x^3 - 3x^2 + 3x - 1 - (x-1) \\ &= (x-1)^3 + (-1)^3(x-1) \end{aligned}$$

- *Indution step*: For $n > 3$, we calculate

$$\begin{aligned} P_{\mathrm{Circ}_n}(x) &= P_{\mathrm{Circ}_n - e}(x) - P_{\mathrm{Circ}_n / e}(x) \\ &= x(x-1)^{n-1} - \big((x-1)^{n-1} + (-1)^{n-1}(x-1)\big) \\ &= (x-1)^n(x-1) - (-1)^{n-1}(x-1) \\ &= (x-1)^n + (-1)^n(x-1) \end{aligned}$$

Now, from Corollary 2.20, we obtain $\|\mathrm{Acyc}(\mathrm{Circ}_n)\| = 2^n - 2$ by considering two distinctive cases:

- If n is even then $\overline{P}_{\mathrm{Circ}_n}(1) = P_{\mathrm{Circ}_n}(-1) = 2^n - 2$
- If n is odd then $\overline{P}_{\mathrm{Circ}_n}(1) = -P_{\mathrm{Circ}_n}(-1) = -\left(-2^n - (-2)\right) = 2^n - 2$

Proposition 2.21 *Unless* P = NP*, there is no algorithm for computing the number of acyclic orientations of a given graph with n vertices, which runs in time polynomial in n.*

2.3.2 Black-Box Equivalence

Want to extend functional equivalence to arbitrary (periodic) local state dynamics, i.e., to arbitrary (periodic) schedules.

Definition 2.22 *Let $G = (X, E)$ be an undirected graph. Let $\alpha : \{1, \ldots, T\} \to \mathcal{P}(X)$ and $\alpha' : \{1, \ldots, T'\} \to \mathcal{P}(X)$ be update schedules having period lengths $T, T' \in \mathbb{N}_+$. Then, we say that α and α' are* black-box equivalent*, in symbols $\alpha \equiv_{\mathrm{bb}} \alpha'$, if and only if for all attribute types D and all sets L of local transition functions with attribute type D and interdependence structure E,*

$$\mathbf{F}_{(L,\alpha)}(\cdot, T) = \mathbf{F}_{(L,\alpha')}(\cdot, T').$$

Note that for all permutations π, π', it holds that $\pi \sim_{U(X,E)\pi'} \iff \pi \equiv_{\mathrm{bb}} \pi'$.

In the following, we want to determine in a certain sense "best" simulations (according to $\equiv_{\mathrm{bb}}$) for serial update schedules, i.e., schedules $\alpha : \{1, \ldots, T\} \to X$. The case for arbitrary schedules is still open.

Mathematical Tools A

In this chapter we discuss relevant terminology and notation for sets, relations, and graphs, some fundamental algorithms, and a few other mathematical preliminaries.

A.1 Sets and relations

We denote the set of integers by $\mathbb{Z}$, the set of non-negative integers by $\mathbb{N}$, and the set of positive integers by $\mathbb{N}_+$. $\mathbb{Z}_2$ denotes the Galois field GF[2].

Sets

The empty set is denoted by $\emptyset$. For an arbitrary set A, $\mathcal{P}(A)$ denotes the power set of A, i.e., the family of all subsets of A, and $\mathcal{P}_+(A)$ denotes the set $\mathcal{P}(A) \setminus \{\emptyset\}$. For an arbitrary finite set A, its cardinality is denoted by $\|A\|$. Let A and B be any sets. Then $A \setminus B$ denotes the difference of A with B, i.e., the set of all elements that are in A but not in B. $A \times B$ denotes the cartesian product, i.e, the set of all pairs (a, b) with $a \in A$ and $b \in B$. For $m \in \mathbb{N}_+$, define $A^m =_{\text{def}} \underbrace{A \times \cdots \times A}_{m \text{ times}}$. Let M be any fixed basic set. For a set $A \subseteq M$, its complement in the basic set M is denoted by $\overline{A}$, i.e., $\overline{A} = M \setminus A$. A multiset A is allowed to contain elements many times. The multiplicity of an element x in a multiset A is the number of occurrences of x in A. The cardinality of a multiset A is also denoted by $\|A\|$.

Functions

Let M and M' be any sets, and let $f : M \to M'$ by any function. The domain of f which we denote by D_f is the set of all $x \in M$ such that $f(x)$ is defined. A function f is total if the domain of f is M. For a set $A \subseteq D_f$, let $f(A) = \{f(x) \mid x \in A\}$ denote the image of A under f. In particular, the range of f which is denoted by R_f is the set $f(D_f)$. For a set $A \subseteq M$, the restriction of a total function f to A is denoted by $f[A]$. The inverse of f is denoted by f^{-1}, i.e, $f^{-1} : M' \to \mathcal{P}(M)$ such that for all $y \in M'$, $f^{-1}(y) = \{x \in M \mid f(x) = y\}$. If $f^{-1}(y)$ is at most a singleton then we omit the braces. The pre-image of A under f is the set $f^{-1}(A) = \{x \in M \mid f(x) \in A\}$.

We use two notations for composition of functions. If f and f' are functions with $f : M \to M'$ and $f' : M' \to M''$, then $(f' \circ f)$ is the function mapping from M to

M'' which is defined for all $x \in M$ as $(f' \circ f)(x) \quad =_{\text{def}} \quad f'(f(x))$. In contrast, we use $f \cdot f'$ to denote $f' \circ f$.

A function $f : M \to M'$ is bijective if f is surjective, i.e., $R_f = M'$ and injective, i.e., for all $y \in R_f$, $f^{-1}(y)$ is a singleton. Suppose $M' = M$ and M is finite. In this case a bijective function f is a permutation. Suppose $M = \{1, 2, \ldots, n\}$. A *cycle* $(i_1\ i_2\ \ldots\ i_k)$ of length k of the permutation $\pi : M \to M$ is a sequence $(i_1, i_2, \ldots, i_k)$ such that $\pi(i_j) = i_{j+1}$ for $1 \leq j < k$ and $\pi(i_k) = i_1$. Each permutation allows a decomposition into cycles.

Orders

In more detail the following can be found in any textbook (e.g., [18, 5]) about theory of orders and lattices.

Let P be any set. A *partial order* on P (or order, for short) is a binary relation $\leq$ on P that is reflexive, antisymmetric, and transitive. The set P equipped with a partial order $\leq$ is said to be a *partially ordered set* (for short, *poset*). Usually, we talk about the poset P. Where it is necessary we write $(P, \leq)$ to specify the order. A poset P is a *chain* if for all $x, y \in P$ it holds that $x \leq y$ or $y \leq x$ (i.e., any two elements are comparable with respect to $\leq$). Such an order is also called a *total order*. A poset P is an *antichain* if for all $x, y \in P$ it holds that $x \leq y$ implies that $x = y$ (i.e., no two elements are comparable with respect to $\leq$).

We consider $\mathbb{N}$ to be ordered by standard total order on the natural numbers. If a set A is partially ordered by $\leq$ then A^m can be considered to be ordered by the *vector-ordering*, i.e., $(x_1, \ldots, x_m) \leq (y_1, \ldots, y_m)$ if and only if for all $i \in \{1, \ldots, m\}$, $x_i \leq y_i$.

An important tool for representing posets is the *covering relation* $\prec$. Let P be a poset and let $x, y \in P$. We say that x is covered by y (or y covers x), and write $x \prec y$, if $x < y$ and $x \leq z < y$ implies that $x = z$. The latter condition is demanding that there be no element z of P with $x < z < y$. A finite poset P can be drawn in a diagram consisting of points (representing the elements of P) and interconnecting lines (indicating the covering relation) as follows: To each element x in P associate a point $P(x)$ in the picture which is above all points $P(y)$ associated to elements y less than x, and connect points $P(x)$ and $P(y)$ by a line if and only if $x \prec y$. A poset can have different representation by diagrams.

Let P and P' be posets. A map $\varphi : P \to P'$ is said to be *monotone* (or order-preserving) if $x \leq y$ in P implies $\varphi(x) \leq \varphi(y)$ in P'. We say that φ is an *(order-)isomorphism* if φ is monotone, injective, and surjective. Two posets P and P' are *isomorphic*, in symbols $P \cong P'$, if there exists an isomorphism $\varphi : P \to P'$. Isomorphic poset shall be considered to be not essentially different: Two finite posets are isomorphic if and only if they can be drawn with identical diagrams.

Words

Sometimes we make no difference between m-tuples $(x_1, \ldots, x_m)$ over a finite set M and words $x_1 \ldots x_m$ of length m over M. Such finite sets are called alphabets. Let Σ be a finite alphabet. Σ^* is the set of all finite words that can be built with letters from Σ. For $x, y \in \Sigma^*$, $x \cdot y$ (or xy for short) denotes the concatenation of x and y. The empty word is denoted by ε. For a word $x \in \Sigma^*$, $|x|$ denotes the length of x. For $n \in \mathbb{N}$, Σ^n is the set of all words $x \in \Sigma^*$ such that with $|x| = n$. For a word $x = x_1 \ldots x_n \in \Sigma^*$ any word $x_1 \ldots x_k$ such that $k \leq n$ is called a prefix of x. We use regular expressions to describe subsets of Σ^* (see, e.g., [21]).

A.2 Graph theory

A graph $G = (V, E)$ consists of a set V of vertices and a set E of edges joining pairs of vertices. The vertex set and edge set of a graph G are denoted by $V(G)$ and $E(G)$, respectively. The cardinality of V is usually denoted by n, the cardinality of E by m. If two vertices are joined by an edge, they are adjacent and we call them *neighbors*. Graphs can be undirected und directed. In undirected graphs, the order in which vertices are joined is irrelevant. An undirected edge joining vertices $u, v \in V$ is denoted by $\{u, v\}$. In directed graphs, each directed edge has an *origin* and a *destination*. An edge with origin $u \in V$ and destination $v \in V$ is represented by an ordered pair (u, v). For a directed graph $G = (V, E)$, the *underlying undirected graph* is the undirected graph with vertex set V that has an undirected edge between two vertices $u, v \in V$ if (u, v) or (v, u) is in E.

Multigraphs

In both undirected and directed graphs, we may allow the edge set E to contain the same edge several times, i.e., E can be a multiset. If an edge occurs several times in E, the copies of that edge are called *parallel edges*. Graphs with parallel edges are also called *multigraphs*. A graph is called *simple*, if each of its edges in contained in E only once, i.e., if the graph does not have parallel edges. An edge joining a vertex to itself, is called a *loop*. A graph is called *loopless* if it has no loops. In general, we assume all graphs to be loopless unless specified otherwise.

Degrees

The *degree* of a vertex v in an undirected graph $G = (V, E)$, denoted by d_v, is the number of edges in E joining v. If G is a multigraph, parallel edges are counted according to their multiplicity in E. The set of neighbors of v is denoted by $N(v)$. $N^0(v)$ denotes the vertex set $N(v) \cup \{v\}$. If the graph under consideration is not clear from the context, these notations can be augmented by specifying the graph as an index. For example, $N_G(v)$ denotes the neighborhood of v in G.

Subgraphs

A graph $G' = (V', E')$ is a *subgraph* of the graph $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. Sometimes we denote this by $G' \subseteq G$. It is a *(vertex-)induced subgraph* if E' contains all edges $e \in E$ that join vertices in V'. The induced subgraph of $G = (V, E)$ with vertex set $V' \subseteq V$ is denoted by $G[V']$. The *(edge-)induced subgraph* with edge set $E' \subseteq E$, denoted by $G[E']$, is the subgraph $G' = (V', E')$ of G, where V' is the set of all vertices in V that are joined by at least one edge in E'.

Walks, paths, and cycles

A *walk* from x_0 to x_k in a graph $G = (V, E)$ is a sequence $x_0, e_1, x_1, e_2, x_2, \ldots, x_{k-1}, e_k, x_k$ alternating between vertices and edges of G, where $e_i = \{x_{i-1}, x_i\}$ in the undirected case and $e_i = (x_{i-1}, x_i)$ in the directed case. The length of a walk is the number of edges on the walk. As shorthands we use $(x_0, x_1, \ldots, x_k)$ and $(e_1, e_2, \ldots, e_k)$ to denote a walk. The walk is called a *path* if $x_i \neq x_j$ for $i \neq j$. A walk with $x_0 = x_k$ is called a *cycle* if $e_i \neq e_j$ for $i \neq j$. A cycle is a *simple cycle* if $x_i \neq x_k$ for $0 \leq i < j \leq k - 1$.

Special graphs

A *tree* is a connected (for a definition see below) undirected graph not containing a cycle. An undirected graph $G = (V, E)$ is called *complete* if it contains all possible pairs of vertices as edges. A complete graph with n vertices is denoted by K^n. A K^n is called a *clique*. A K^2 is a graph of two vertices with one edge joining them. A K^3 is also called a *triangle* or *triad*. A graph without edges is called *empty*. An *independent set* within a graph $G = (V, E)$ is a vertex set $U \subseteq V$ such that $G[U]$ is empty. A graph $G = (V, E)$ is called *bipartite* if there are independent vertex sets $V_1, V_2 \subseteq V$ such that V_1 and V_2 are disjoint and $V_1 \cup V_2 = V$. We denote by $E(V_1, V_2)$ the set of edges joining vertices from V_1 with vertices from V_2. If $E(V_1, V_2) = V_1 \times V_2$ then G is called a *complete bipartite graph*. Such a graph is denoted by K_{n_1,n_2} if V_1 consists of n_1 vertices and V_2 of n_2 vertices. A $K_{1,n}$ is also called a *star*. For two graphs $G = (V, E)$ and $G' = (V', E')$ we denote by $G \oplus G'$ the graph consisting of the disjoint union of the graphs G and G'.

Graph classes

Two graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic*, denoted by $G \simeq G'$, if there is a bijective mapping $\varphi : V \to V'$ such that for all vertices $u, v \in V$ the following is true: in the case that G and G' are directed graphs it holds that $(u, v) \in E \Leftrightarrow (\varphi(u), \varphi(v)) \in E'$, and in the case that G and G' are undirected graphs it holds that $\{u, v\} \in E \Leftrightarrow \{\varphi(u), \varphi(v)\} \in E'$. A set of graphs is called a *graph class* if for each graph G in the class all graphs isomorphic to G belong to the class as well.

A.3 Algorithmics

Most results of this work relate to algorithms. In the following we mention essential problems and concepts which are needed more than once.

For two functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$ we say that f is in $O(g)$ if there are constant $n_0, c \in \mathbb{N}_+$ such that for all $n \geq n_0$, $f(n) \leq c \cdot g(n)$. We say that f is in $\Omega(g)$ if g is in $O(f)$. We say that f is in $\Theta(g)$ if f is in $O(g) \cap \Omega(g)$.

Connected components

An undirected graph $G = (V, E)$ is *connected* if every vertex can be reached from every other vertex, i.e., if there is a path from every vertex to every other vertex. A graph consisting of a single vertex is also taken to be connected. Graphs that are not connected are called *disconnected*. For a given undirected graph $G = (V, E)$, a *connected component* of G is an induced subgraphs $G' = (V', E')$ that is connected and maximal, i.e., there is no connected subgraph $G'' = (V'', E'')$ such that $V'' \supset V'$. Checking whether a graph is connected and finding all its connected components can be done in time $O(n + m)$ using depth-first search or breadth-first search.

A directed graph $G = (V, E)$ is *strongly connected* if there is a directed path from every vertex to every other vertex. A *strongly connected component* of a directed graph G is an induced subgraph that is strongly connected and maximal. The strongly connected components of a directed graph can be computed in time $O(n + m)$ using a depth-first search.

NP-*completeness*

It is important to consider the running-time of an algorithm for a given problem. Usually, one wants to give an upper bound on the running time of the algorithm for inputs of a certain size. If the running-time of an algorithm is $O(n^k)$ for some $k \in \mathbb{N}$ and for inputs of size n, we say that the algorithm runs in polynomial time. For graph problems, the running-time is usually specified as a function of n and m, the number of vertices and edges of the graph, respectively. For many problems, however, no polynomial-time algorithm has been discovered. Although one cannot rule out the possible existence of polynomial-time algorithms for such problems, the theory of NP-completeness provides means to give evidence for the computational intractability of a problem.

A decision problem is in the complexity class NP if there is a nondeterministic Turing machine that solves the problem in polynomial time. That is to say that the answer to a problem instance is “yes” if there exists a solution in the set of all possible solutions to the instance which is of polynomial size. Moreover, the test whether a potential solution is an actual solution must be performed in polynomial time. Note that a decision problem

is usually considered to consist of the set of the "yes"-instances. A decision problem is NP-hard if every problem in NP can be reduced to it via a polynomial-time many-one reduction. (A polynomial-time many-one reduction from a set A to a set B is a function computable in polynomial time such that for all instances x, $x \in A \Leftrightarrow f(x) \in B$.) Problems that are NP-hard and belong to NP are called NP-*complete*. A polynomial-time algorithm for an NP-hard problem would imply polynomial-time algorithms for all problems NP—something that is considered very unlikely. Therefore, the NP-hardness of a problem is considered substantial evidence for the computational difficulty of the problem.

A standard example of an NP-complete problem is 3SAT, i.e., checking whether a given propositional formula given as a 3CNF has a satisfying assignment. To be more precise, a kCNF is a formula $H = C_1 \wedge \cdots \wedge C_m$ consisting of clauses C_i each of which has the form $C_i = l_{i1} \vee l_{i2} \vee \cdots \vee l_{ik}$ where l_{ij} is either a positive or a negative literal. A positive literal is some variable, say x_k, and a negative literal is the negation of some variable, say $\overline{x_k}$.

The class of complements of NP sets is denoted by coNP, i.e., coNP $= \{\overline{A} | A \in \text{NP}\}$.

For optimization problems (where the goal is to compute a feasible solution that maximizes or minimizes some objective function), we say that the problem is NP-*hard* if the corresponding decision problem (checking whether a solution with objective value better than a given value k exists) is NP-hard.

#P-*completeness*

A complexity class closely related to NP is the class #P which has been introduced in [34, 33] to provide evidence for the computational intractability of counting problems. The class #P consists of all problems of the form "compute $f(x)$" where $f(x)$ is the number of accepting paths of a nondeterministic Turing machine running in polynomial time. Equivalently, a #P-functions counts the number of solutions to instances of an NP-problem. We say that a function f is #P-*complete* if it belongs to #P and every function $g \in$ #P is polynomial-time Turing reducible to f, i.e., g can be computed by a deterministic polynomial-time Turing machines which is allowed to make queries to f and answering these queries is done within one step (see, e.g., [21, 20]). The canonical example of a #P-complete problem is #3SAT, i.e., counting the number of satisfying assignments of a propositional formula given as a 3CNF. One of the most prominent #P-complete problem is counting the number of perfect matchings in a bipartite graph [33]. As in the case of NP, if there is a polynomial-time algorithm for computing some #P-complete function from #P then there are polynomial-time algorithms for all #P-functions—which is equally considered unlikely. In particular, such a polynomial-time algorithm would imply that P = NP.

[]

Bibliography

[1] F. Amblard and G. Deffuant. The role of network topology on extremism propagation with relative agreement opinion dynamics. *Physica A*, 343:725–738, 2004.

[2] Y. Bar-Yam. *Dynamics of Complex Systems*. Studies in Nonlinearity. Addison-Wesley, Reading, MA, 2003.

[3] C. Baraldi, G. Corsi, and E. Esposito. *GLU. Glossar zu Niklas Luhmanns Theorie sozialer Systeme*. Suhrkamp Verlag, Frankfurt am Main, 1997.

[4] D. Cartwright and F. Harary. Structural balance: A generalization of Heider's theory. *The Psychological Review*, 63(5):277–293, 1956.

[5] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Orders*. Cambridge University Press, Cambridge, 1990.

[6] G. Deffuant. Comparing extremism propagation patterns in continuous opinion models. *Journal of Artificial Societies and Social Simulation*, 9(3), 2006.

[7] G. Deffuant, F. Amblard, G. Weisbuch, and T. Faure. How can extremism prevail? A study on the relative agreement interaction model. *Journal of Artificial Societies and Social Simulation*, 5(4), 2002.

[8] G. Deffuant, G. Weisbuch, F. Amblard, and T. Faure. Simple is beautiful ... and necessary. *Journal of Artificial Societies and Social Simulation*, 6(1), 2003.

[9] B. Derrida and Y. Pomeau. Randomized networks of automata: A simple annealed approximation. *Europhysics Letters*, 1(2):45–49, 1986.

[10] R. Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer-Verlag, Berlin, 3rd edition, 2003.

[11] J. C. Dittmer. Consensus formation under bounded confidence. *Nonlinear Analysis*, 47:4615–4621, 2001.

[12] M. Edwards, S. Huet, F. Goreaud, and G. Deffuant. Comparing an individual-based model of behaviour diffusion with its mean field aggregate approximation. *Journal of Artificial Societies and Social Simulation*, 6(4), 2003.

[13] C. Elsenbroich. Explanation in agent-based modelling: Functions, causality or mechanisms? *Journal of Artificial Societies and Social Simulation*, 15(3):1, 2012.

[14] N. E. Friedkin. *A Structural Theory of Social Influence*. Structural Analysis in the Social Sciences. Cambridge University Press, Cambridge, 1998.

[15] N. E. Friedkin and E. C. Johnsen. Social influence and opinions. *Journal of Mathematical Sociology*, 15(3–4):193–2005, 1990.

[16] N. E. Friedkin and E. C. Johnsen. Social influence networks and opinion change. *Advances in Group Processes*, 16:1–29, 1999.

[17] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

[18] G. A. Grätzer. *General Lattice Theory*. Akademie-Verlag, Berlin, 1978.

[19] R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence: Models, analysis, and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3):2, 2002.

[20] L. A. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2002.

[21] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, MA, 2nd edition, 2001.

[22] K. Kacperski and J. A. Hołyst. Phase transitions as a persistent feature of groups with leaders in models of opinion formation. *Physica A*, 287:631–643, 2000.

[23] S. Kassin, S. Fein, and H. R. Markus. *Social Psychology*. Houghton Mifflin Company, Boston, MA, 8th edition, 2008.

[24] S. A. Kauffman. *The Origins of Order. Self-Organization and Selection in Evolution*. Oxford University Press, Oxford, 1993.

[25] S. Kosub. Computational analysis of complex systems. Habilitation, Fakultät für Informatik, Technische Universität München, 2007.

[26] R. Laubenbacher and B. Pareigis. Update schedules of sequential dynamical systems. *Discrete Applied Mathematics*, 154(6):980–994, 2006.

[27] N. Linial. Hard enumeration problems in geometry and combinatorics. *SIAM Journal on Algebraic and Discrete Methods*, 7(2):331–335, 1986.

[28] N. Luhmann. *Soziale Systeme. Grundriß einer algemeinen Theorie*. Suhrkamp Verlag, Frankfurt am Main, 1984.

[29] N. Luhmann. *Die Gesellschaft der Gesellschaft*. Suhrkamp Verlag, Frankfurt am Main, 1997.

[30] H. S. Mortveit and C. M. Reidys. *An Introduction to Sequential Dynamical Systems*. Springer-Verlag, New York, NY, 2008.

[31] H. G. Schuster. *Deterministisches Chaos*. VCH, Weinheim, 1994.

[32] R. P. Stanley. Acyclic orientations of graphs. *Discrete Mathematics*, 5(2):171–178, 1973.

[33] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[34] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[35] G. von Randow. When the centre becomes radical. *Journal of Artificial Societies and Social Simulation*, 6(1), 2003. Translated by K. G. Troitzsch.

[36] G. Voutsadakis. Threshold agent networks: An approach to modelling and simulation. *Applied Mathematics and Computation*, 142(2–3):521–543, 2003.

[37] W. Weaver. Science and complexity. *American Scientist*, 36:536–544, 1948.

[38] G. Weisbuch, G. Deffuant, and F. Amblard. Persuasion dynamics. *Physica A*, 353:555–575, 2005.

[39] S. Wolfram. *A New Kind of Science*. Wolfram Media, Champaign, IL, 1994.