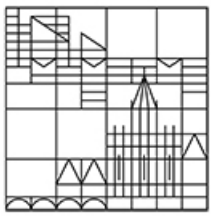
Fachbereich Informatik und
Informationswissenschaft

Universität
Konstanz



Lecture Notes

Network Dynamics

taught in Winter terms 2010, 2011, 2012, 2013

by

Sven Kosub

February 18, 2014

Version v3.25

Contents

An Internet example 1

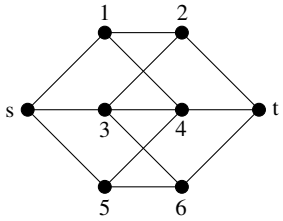
Internet routing involves the *next-hop* principle in the following way:

- A *path* is a sequence of entities which pairwise share a direct data link.
- Sending data from an entity to an entity (the next hop) over a direct data link is called *forwarding*.
- A *transmission* is the process of successively forwarding data from a source to a destination over a path with source as its first and the destination as its last element.
- The selection of a path for a transmission is called *routing*.

1.1 The routing hierarchy

A routing hierarchy reduces the number of paths through a communication network. Note that the number of paths can be exponential in the number of nodes, e.g., in a K^n network.

Example: Suppose a small portion of the Internet is



Then, there are 29 different (s,t)-paths. This represents a part of Internet graph at the router level. Now, introduce a partition of $\{s,1,2,\ldots,6,t\}$:

$$A =_{\text{def}} \{s\}, \quad B =_{\text{def}} \{1,2,3\}, \quad C =_{\text{def}} \{4,5,6\}, \quad D =_{\text{def}} \{t\}$$

If we consider the path p

$$\begin{array}{ccccccccccccccc} s & \to & 1 & \to & 4 & \to & 5 & \to & 6 & \to & 3 & \to & 2 & \to & t \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ A & & B & & C & & C & & C & & B & & B & & D \end{array}$$

then p traverses B twice. This is inappropriate, if A, B, C, D are subnets that belong to independent administrative domains (a.k.a. Autonomous Systems).

In a routing hierarchy, first, the path consisting of subnets is chosen and, second, the path through each subnet is selected.

> **Example (cont'd):** The number of paths according to the routing hierarchy is as follows:
>
> - $A \to B \to D$: 2 paths
> - $A \to C \to D$: 2 paths
> - $A \to B \to C \to D$: 6 paths
> - $A \to C \to B \to D$: 6 paths
>
> That is, the number of paths is reduced to 16.

According to the routing hierarchy, there are two types of protocols:

- Intra-Domain: RIP, OSPF within a subnet
- Inter-Domain: BGP between subnets

The Border Gateway Protocol (BGP) is the "real" Internet routing protocol.

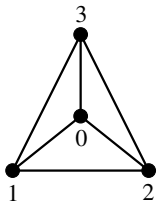
1.2 Policy routing

We consider the formal routing problem for destination node 0. Given an undirected AS graph $G = (V, E)$ where $0 \in V$, the path set P^u is defined as follows

$$P^u =_{\text{def}} \{\ p \mid p \text{ is a path in } G \text{ starting at } u \text{ and ending at } 0\ \} \cup \{\varepsilon\}$$

for each node $u \in V$. Here, ε denotes the empty path. We assume that all path sets are ranked, i.e., $(P^u, \succeq_u)$ is a total preorder (a total, reflexive, and transitive relation) for each $u \in V$. Furthermore, $p \succ_u \varepsilon$ for all $p \in P^u \setminus \{\varepsilon\}$. This assumption states that reachability is always superior to non-reachability.

> **Example:** To clarify the notions above, consider the following AS graph $G = (V, E)$ together with ranked path sets P^u is given



$$\begin{aligned}
P^0 &: (0) \succ_0 \varepsilon \\
P^1 &: (1,0) \succ_1 (1,2,0) \succ_1 (1,3,0) \succ_1 (1,3,2,0) =_1 (1,2,3,0) \succ_1 \varepsilon \\
P^2 &: (2,3,0) \succ_2 (2,1,0) \succ_2 (2,1,3,0) \succ_2 (2,3,1,0) \succ_2 (2,0) \succ_2 \varepsilon \\
P^3 &: (3,1,0) \succ_3 (3,1,2,0) \succ_3 (3,0) \succ_3 (3,2,0) \succ_3 (3,2,1,0) \succ_3 \varepsilon
\end{aligned}$$

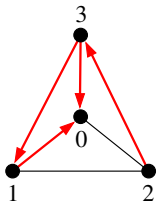
For instance, node 3 always prefers a route over node 1 over a direct route to zero or a route over node 2.

Each node in the As graph chooses one path from its path set. Then, a *configuration* is a mapping $\pi : u \mapsto \pi(u) \in P^u$. A configuration π is said to be *confluent* if and only if the union of all paths $\pi(u)$ is a tree with root 0. A configuration π is said to be *stably confluent* if and only if π is confluent and no node u can choose a better path to 0 without losing confluency.

Example (cont'd): If all nodes in the AS graph above choose their best paths according to ranked path sets, i.e.,

$$\pi(1) = (1,0), \quad \pi(2) = (2,3,0), \quad \pi(3) = (3,1,0),$$

then the configuration is not confluent:

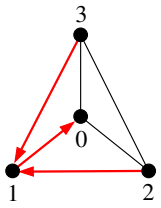


$$\begin{aligned}
P^0 &: (0) \succ_0 \varepsilon \\
P^1 &: (1,0) \succ_1 (1,2,0) \succ_1 (1,3,0) \succ_1 (1,3,2,0) =_1 (1,2,3,0) \succ_1 \varepsilon \\
P^2 &: (2,3,0) \succ_2 (2,1,0) \succ_2 (2,1,3,0) \succ_2 (2,3,1,0) \succ_2 (2,0) \succ_2 \varepsilon \\
P^3 &: (3,1,0) \succ_3 (3,1,2,0) \succ_3 (3,0) \succ_3 (3,2,0) \succ_3 (3,2,1,0) \succ_3 \varepsilon
\end{aligned}$$

If nodes 1 and 3 still choose their best paths and node 2 chooses its second best path to 0, i.e.,

$$\pi(1) = (1,0), \quad \pi(2) = (2,1,0), \quad \pi(3) = (3,1,0),$$

then the configuration is stably confluent:

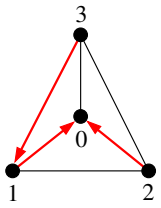


$$\begin{aligned}
P^0 &: (0) \succ_0 \varepsilon \\
P^1 &: (1,0) \succ_1 (1,2,0) \succ_1 (1,3,0) \succ_1 (1,3,2,0) =_1 (1,2,3,0) \succ_1 \varepsilon \\
P^2 &: (2,3,0) \succ_2 (2,1,0) \succ_2 (2,1,3,0) \succ_2 (2,3,1,0) \succ_2 (2,0) \succ_2 \varepsilon \\
P^3 &: (3,1,0) \succ_3 (3,1,2,0) \succ_3 (3,0) \succ_3 (3,2,0) \succ_3 (3,2,1,0) \succ_3 \varepsilon
\end{aligned}$$

Finally, if nodes 1 and 3 both choose their best paths and node 2 chooses the second-to-worst path, i.e.,

$$\pi(1) = (1,0), \quad \pi(2) = (2,0), \quad \pi(3) = (3,1,0),$$

then the resulting configuration is confluent but not stably confluent.



$$
\begin{aligned}
P^0 &: (0) \succ_0 \varepsilon \\
P^1 &: (1,0) \succ_1 (1,2,0) \succ_1 (1,3,0) \succ_1 (1,3,2,0) =_1 (1,2,3,0) \succ_1 \varepsilon \\
P^2 &: (2,3,0) \succ_2 (2,1,0) \succ_2 (2,1,3,0) \succ_2 (2,3,1,0) \succ_2 (2,0) \succ_2 \varepsilon \\
P^3 &: (3,1,0) \succ_3 (3,1,2,0) \succ_3 (3,0) \succ_3 (3,2,0) \succ_3 (3,2,1,0) \succ_3 \varepsilon
\end{aligned}
$$

1.3 Best-response dynamics

We give a formal description of a simplified BGP dynamics.

We are given a set P of paths toa (destination) node 0. For a path $p(a_0, a_1, \ldots, a_n) \in P$ such that $a_n = 0$, define

$$\Gamma(p) =_{\text{def}} \{\ (a_i, \ldots, a_n) \mid i \in \{0, 1, \ldots, n\}\ \}$$

Furthermore, we define the following path sets:

$$
\begin{aligned}
P^* &=_{\text{def}} \bigcup_{p \in P} \Gamma(p) \ \cup\ \{\varepsilon\} \\
P^a &=_{\text{def}} \{\ p \in P^* \mid p = (a, \ldots, 0)\ \} \cup \{\varepsilon\} \quad \text{for each } a \text{ lying on some path } p \in P
\end{aligned}
$$

There is an AS graph $G[P]$ induced by the path set P:

- vertex set $V = V[P]$ is defined to consist of all nodes occurring on some path $p \in P$
- edge set $E = E[P]$ is the defined to consist of all (undirected) edges occurring on some path $p \in P$, i.e., if $p = (\ldots, a, a_{i+1}, \ldots) \in P$ then $\{a_i, a_{i+1}\} \in E$.

Given a configuration $\pi : V \to P^*$, how should a node choose a new path and how does this choice affect the configuration? For $a \in V$, define

$$\beta_a(\pi) =_{\text{def}} \max\nolimits_{\succ_a} \{\ p \in P^a \mid G[\pi(V \setminus \{a\}) \cup \{p\}] \text{ is a forest } \}$$

In order to avoid tie-breaks, $(P^a, \succ_a)$ is a total order satisfying $p \succ_a \varepsilon$. We extend $\beta_.$ to map configurations to configurations as follows

$$\beta(\pi) : a \mapsto \beta_a(\pi),$$

i.e., $\beta(\pi)$ is the configuration assigning to node a path $\beta_a(\pi)$.

Proposition 1.1 *Let P be a set of paths such that $G[P]$ is connected. Let π be any configuration on $G[P]$. Then, it holds*

$$\pi = \beta(\pi) \iff \pi \text{ is stably confluent}$$

Proof: We argue for both directions individually.

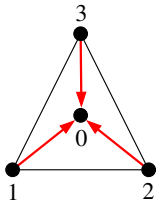
($\Leftarrow$): By definition.

($\Rightarrow$): Suppose $\pi = \beta(\pi)$. Then, $G[\pi(V)]$ is a forest. Assume $G[\pi(V)]$ is not a tree. Then, there is an $a \in V$ such that $\pi(a) = \varepsilon$ (as each path leads to the root 0). Let $a \in V$ be a node such that $\pi(a) = \varepsilon$ and distance to 0 is minimal in $G[P]$. Let $b \in V$ be the next node on a shortest path from a to 0. Then, $\pi(b) \neq \varepsilon$. Consider path $p = (a, \pi(b)) \in P^a$. Then, $p \succ_a \varepsilon$. We obtain that $\beta_a(\pi)$ is not maximal with respect to $\succ_a$, in contradiction to our assumption. Thus, $G[\pi(V)$ is a tree. Therefore, π is stably confluent.

This proves the proposition. ■

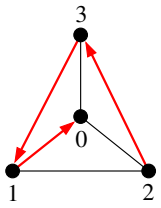
Actually, BGP uses only the next hop on the best path.

Example (cont'd): Consider the AS graph and the sets of paths from above. Suppose all nodes update their paths simultaneously. Then, we could obtain the following evolution of the routing network when starting with the inital configuration in time step $t = 0$:



$$
\begin{aligned}
P^0 &: (0) \succ_0 \varepsilon \\
P^1 &: (1,0) \succ_1 (1,2,0) \succ_1 (1,3,0) \succ_1 (1,3,2,0) =_1 (1,2,3,0) \succ_1 \varepsilon \\
P^2 &: (2,3,0) \succ_2 (2,1,0) \succ_2 (2,1,3,0) \succ_2 (2,3,1,0) \succ_2 (2,0) \succ_2 \varepsilon \\
P^3 &: (3,1,0) \succ_3 (3,1,2,0) \succ_3 (3,0) \succ_3 (3,2,0) \succ_3 (3,2,1,0) \succ_3 \varepsilon
\end{aligned}
$$

After one update step of each nodes given the network in $t = 0$, the resulting routing network is the following ($t = 1$):

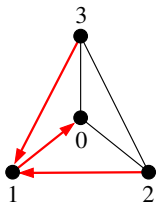


$$
\begin{aligned}
P^0 &: (0) \succ_0 \varepsilon \\
P^1 &: (1,0) \succ_1 (1,2,0) \succ_1 (1,3,0) \succ_1 (1,3,2,0) =_1 (1,2,3,0) \succ_1 \varepsilon \\
P^2 &: (2,3,0) \succ_2 (2,1,0) \succ_2 (2,1,3,0) \succ_2 (2,3,1,0) \succ_2 (2,0) \succ_2 \varepsilon \\
P^3 &: (3,1,0) \succ_3 (3,1,2,0) \succ_3 (3,0) \succ_3 (3,2,0) \succ_3 (3,2,1,0) \succ_3 \varepsilon
\end{aligned}
$$

Observe that node 2 has chosen path $(2,3,0)$ as the new best path. However, as only the edge $(2,3)$, i.e., the edge to the next hop of the best path, has been included in the routing network, and node 3 has chosen $(3,1,0)$ as the new best path, adding the edge $(3,1)$ to the routing network, the current available

path of node 2 to destination 0 is $(2, 3, 1, 0)$ which is an alternative worse than the originally chosen one.

Given the network for $t = 1$, another simultaneous update step of all nodes gives the routing network for time step $t = 2$.



$$
\begin{aligned}
P^0 &: (0) \succ_0 \varepsilon \\
P^1 &: (1,0) \succ_1 (1,2,0) \succ_1 (1,3,0) \succ_1 (1,3,2,0) =_1 (1,2,3,0) \succ_1 \varepsilon \\
P^2 &: (2,3,0) \succ_2 (2,1,0) \succ_2 (2,1,3,0) \succ_2 (2,3,1,0) \succ_2 (2,0) \succ_2 \varepsilon \\
P^3 &: (3,1,0) \succ_3 (3,1,2,0) \succ_3 (3,0) \succ_3 (3,2,0) \succ_3 (3,2,1,0) \succ_3 \varepsilon
\end{aligned}
$$

For $t > 2$, the configurations do not change anymore. We have reached an equilibrium or fixed point.

1.4 Fixed-point analysis

A fundamental question in Interdomain routing is: When does BGP converge?

If we investigate this question algorithmically then we ask if a given BGP system has a fixed point or allows for reaching a fixed point. Here, a BGP system is defined to consist of an AS graph $G = (V, E)$ and a family of ranked path-set in G to destination $0 \in V$ (in the above-mentioned sense) where we assume that not all paths to 0 need be included.

Answers to these questions are negative, in general:

- There is no guarantee of convergence (see examples on the Assignments).
- It is NP-complete to decide if a BGP system has a stably confluent state [16].
- It is PSPACE-complete to decide if a BGP system always converges into some stably confluent state [**?**].

Why, then, does BGP appear to be rather stable in everyday experience?

There are rational rules for local policies which ensure convergence, but they are not part of the protocol specification. These rules are based on contractual business relationships between Autonomous Systems which typically belong to a certain Internet Service Provider. For simplification, let us assume that there are only two types of Autonomous Systems: *customers* and *providers*. Customers buy routes from providers to get global connectivity to 0. Accordingly, providers sell routs to customers.

Given an AS graph $G = (V, E)$ we can decompose the neighborhood of a node $v \in V$:

- $\mathrm{Cust}(v)$ is the set of all customers of v (buying routes from v)

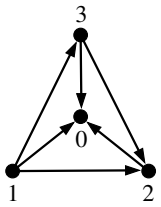
- $\mathrm{Prov}(v)$ is the set of all providers of v (selling routes to v)

The AS graph can be oriented given such a decmposition. Let $u, v \in V$ be two nodes such $\{u, v\} \in E$. Then,

- an oriented edge $u \to v$ indicates that $u \in \mathrm{Cust}(v)$ and
- an oriented edge $u \leftarrow v$ indicates that $u \in \mathrm{Prov}(v)$.

The orientation can be extended to paths. A path p in G is said to be *valley-free* if and only if its orientation pattern belongs to $\to^*\leftarrow^*$.

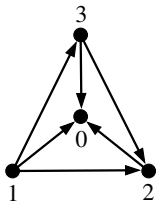
Example: We consider an AS graph with contracts among the nodes:



- The path $(1, 3, 2, 0)$ is valley-free: $\to \to \to$
- The path $(3, 1, 2, 0)$ is not valley-free: $\leftarrow \to \to$
- The path $(1, 3, 0, 2)$ is valley-free: $\to \to \leftarrow$

A path $p = (u_0, u_1, \ldots, u_m)$ (with $u_m = 0$) is called a *customer route* iff $u_1 \in \mathrm{Cust}(u_0)$, and is called a *provider route* iff $u_1 \in \mathrm{Prov}(u_0)$.

Example (cont'd): We classify routes other than ε for the AS graph above. Red routes are customer routes, black routes are provider routes.



$$
\begin{aligned}
P^1 &: (1,0) \succ_1 (1,2,0) \succ_1 (1,3,0) \succ_1 (1,3,2,0) =_1 (1,2,3,0) \\
P^2 &: (2,3,0) \succ_2 (2,1,0) \succ_2 (2,1,3,0) \succ_2 (2,3,1,0) \succ_2 (2,0) \\
P^3 &: (3,1,0) \succ_3 (3,1,2,0) \succ_3 (3,0) \succ_3 (3,2,0) \succ_3 (3,2,1,0)
\end{aligned}
$$

Given these notions we can formulate the Gao-Rexford convergence criterion [13]: A BGP system converges into a stable confluent state if the following three conditions are fulfilled:

1. All paths are valley-free.
2. The oriented AS graph is acyclic.
3. For all nodes a and all paths p, q beginning with a, if p is a customer route and q is a provider route then $p \succ_a q$.

Note that this criterion expresses only sufficient conditions for convergence.

For instance, if we eliminate all routes from the example above that are not valley-free then the Gao-Rexford criterion applies to the BGP system.

Networks 2

We foster a data-driven approach to understanding dynamical network behavior.

2.1 Network exploration and analysis

Sketch of research pipeline in empirical sciences:

1. Theory ← often omitted from the cycle
2. Hypotheses
3. Research design
4. Data collection
5. Exploration and analysis ← often data-driven studies start here
6. Interpretation and presentation

These steps are iterated (when new evidence comes in).

Step 5 will be the focus of this course. We will study

- formal and algorithmic concepts
- simulation and modelling techniques

for evaluting time-dependent network data.

2.2 Network data

2.2.1 Data

Data refers to variables for *entites* (or *units of observation*). More specifically,

- A is a set of (atomic) *items*,
- for $i \in A$, variable x_i represents values of a common *attribute* for all items in A, i.e., x is a mapping $x : A \to R : i \mapsto x_i$, or $x = (x_i)_{i \in A}$ where $x_i \in R$,

- R is the *range* of x, A is called the *domain* of x

Typically in emprical research, multiple attributes are collected in tables where the columns represent items and the rows represent attributes.

According to the range, attributes can be classified:

- *nominal* or *categorical*: there are no relationships among the elements of the range other than equality or inequality (e.g., names, types, labels)
- *ordinal*: the range satisfy certain order properties such as required for weak orders, preference relations, rankings (e.g., paths in policy routing)
- *numerical*: the range consists of number such as $\mathbb{N}$ or $\mathbb{R}_{\geq 0}$.

We assume that 0 represents a massing or neutral datum.

2.2.2 Dyadic data

Entities need not be atomic; they can be compound objects of more elementary entities. A *dyad* is a pair of items.

> **Example:** In a study we could explore relationship among married couples. The relevant data may include:
>
> - attributes of individuals: gender, income, personality
> - attributes of the couple: age difference, duration of marriage, number of children

The general assumption in (classical) dyadic data analysis is that dyads are independent.

We say that two dyads *overlap* if and only if they share a member. This gives us the characteristic of network data:

1. Units of observation are dyads.
2. Dyads are overlapping.

That is, the essential assumption in network analysis is that dyads are dependent. We even cab define network analysis as the study of effects of overlapping dyads.

2.2.3 Time-dependent data

Attribute values may change over time. And, there are differences in how data can depend on time. In general, data time-dependent data can be classified as follows:

- *panel data* (or *longitudinal data*): we have attributes values of all items for at least two points in time, i.e., $x(1), x(2), \ldots, x(k)$ where $x(j) = (x_i(j))_{i \in A}$.
- *time-series data*: we have attribute values of a single item over time.
- *cross-sectional data*: we have attribute values of all items for one specific point in time.
- *event data*: we have attribute values for items labelled with a time stamp (e.g., log files, audit trails, live scores, etc.)

Typically, event data are transformed into panel data.

2.3 Network representations

We adopt a network view where we consider networks to be representations of a specific format. That is, we are not so much interested in *what* is represented, but *how* it is represented.

2.3.1 Whole networks

As overlapping dyads are the fundamental objects of network analysis, we need a notion to collect all possible dependencies among dyads. This is done by introducing interaction domains.

Definition 2.1 *Let A be a set of items. An* interaction domain *$\mathcal{I}$ on A is a binary, symmetric relation $\mathcal{I} \subseteq A \times A$.*

In many cases, $\mathcal{I} = A \times A$ or $\mathcal{I} = (A \times A) \setminus \{ (i, i) \mid i \in A \}$. However, when studying BGP systems, the interaction domain is the AS graph.

Definition 2.2 *Let A be a set of items. A* (whole) network *consists of a set of attributes on an interaction domain $\mathcal{I} \subseteq A \times A$ and a (possibly empty) set of attributes on A.*

For a network, items of A represent *actors*, and attribute values $x_{i,j} \neq 0$, where $(i,j) \in \mathcal{I}$, are *ties*. Notice that x_{ij} is a usual abbreviation for $x_{(i,j)}$ for any dyad $(i,j) \in \mathcal{I}$.

Definition 2.3 *Let $x : \mathcal{I} \to R$ be an attribute defined on an interaction domain $\mathcal{I} \subseteq A \times A$. The (weighted, directed)* graph $G(x) = (V, E, w)$ of network x *consists of*

- *vertex set* $V =_{\text{def}} A$,
- *edge set* $E =_{\text{def}} \{\ (i,j) \in \mathcal{I} \mid x_{ij} \neq 0\ \}$, *and*
- *edge weights* $w : E \to R : (i,j) \mapsto x_{ij}$.

If $x_{ij} = x_{ji}$ for all $(i,j) \in \mathcal{I}$, $G(x)$ can be defined correspondingly as an undirected graph.

A completion of an attribute to the full interaction domain $A \times A$ by imputing zeroes gives the *adjacency matrix* of the associated weighted graph, which is another representation of a network.

Definition 2.4 *Let $x : \mathcal{I} \to R$ be an attribute defined on an interaction domain $\mathcal{I} \subseteq A \times A$. The (binary)* relation $\to \subseteq A \times A$ of network x *is defined by*

$$(i,j) \in \to \quad \Longleftrightarrow_{\text{def}} \quad (i,j) \in \mathcal{I} \wedge x_{ij} \neq 0$$

In infix notation, this is written as $i \to j$.

2.3.2 Two-mode networks

Assume that the observational units are relations between pairs of items of different types, e.g., users and fan sites on Facebook, authors and scientific papers, or politicians and boards.

We generalize interaction domains.

Definition 2.5 *An* affiliation domain *is a relation $\mathcal{A} \subseteq A \times S$ on disjoint sets A and S.*

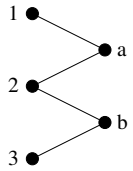
Definition 2.6 *A* two-mode network *consists of a set of attributes on an affiliation domain $\mathcal{A} \subseteq A \times S$ and a (possibly empty) set of attributes on A and S.*

All notions for networks translate to two-mode networks. Note that two-mode networks are bipartite by definition.

Definition 2.7 *Let $X \in R^{n \times m}$ be the matrix associated with a two-mode network attribute, $\|A\| = n$ and $\|S\| = m$. The networks associated with the matrices $X \cdot X^T$ and $X^T \cdot X$ are called* one-mode projections.

Note that the interaction domain of $X \cdot X^T$ is $A \times A$ and the interaction domain of $X^T \cdot X$ is $S \times S$.

Example: Consider sets $A = \{1, 2, 3\}$ and $S = \{a, b\}$. Suppose a two-mode network attribute is given by the following graph and the associated matrices X and X^T:

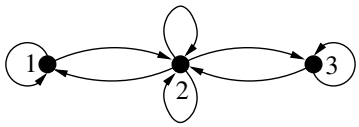


$$X = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad X^T = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Then, we calculate

$$X \cdot X^T = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

The (multi)graph of the network can be drawn as follows:



Analogously, we calculate for the other one-mode projection

$$X \cdot X^T = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

The (multi)graph of the network can be drawn as follows:

Note that each directed edge $u \to v$ respresents one walk from u to v.

2.3.3 Ego and personal networks

Definition 2.8 *An affiliation domain $\mathcal{A} \subseteq A \times S$ is said to be* egocentric *if and only if $\|\{\ s \mid (i, s) \in \mathcal{A}\ \}\| = 1$ for all $i \in A$. In other words, every element of A is affiliated with exactly one element of S.*

In an egocentric domain, elements of S are called *egos*, and elements of A are called *alteri*. An egocentric domains is uniquely decomposable into its *ego partition* (see example below).

Definition 2.9 *Given a two-mode network on an egocentric domain, each restriction of its attributes to an element of the ego partition defines an* ego network.

Definition 2.10 *Let A and S be disjoint sets, let $\mathcal{I} \subseteq A \times A$ be an interaction domain, and let $\mathcal{A} \subseteq A \times S$ be an egocentric affiliation domain. For a set of attributes defined on $\mathcal{I}$ and $\mathcal{A}$, every restriction induced by an element of the ego partition defines one* personal network.

Example: ...

2.3.4 Time-dependent networks

We consider attributes on an interaction (or affiliation) domain changing over time. The focus is on panel network-data.

Definition 2.11 *A* time-dependent network *is a set of attributes on an interaction domain $\mathcal{I} \subseteq A \times A$ and a (possibly empty) set of attributes on A, where all atributes depend on (same) time $t \in \mathbb{N}$.*

Note that we consider time-discrete networks.

2.4 Networks as dynamical systems

In this section, we want to introduce specific formal notions for studying the dynamical behavior of networks. We restrict ourselves to single-attribute networks (with a fixed interaction domain).

Let $x : \mathcal{I} \to R$ be an attribute. For the sake of convenience, we assume that x is a numerical attribute. Furthermore, we consider an infinite sequence of identical copies of x, i.e., $(x(t))_{t\in\mathbb{N}}$ or $x : \mathcal{I} \times \mathbb{N} \to R$. The attribute values are called *states*. The set of all possible sequences is called a *process*; one specific sequence is called *trajectory*. A *dynamic* F is a mechanism for selecting trajectories of a process. A dynamic makes assumptions on how the state at time step k will look like; here, depending only on the initial state z_0 and time k. We thus can express a dynamic as a sequence $(\varphi_t)_{t\in\mathbb{N}}$ where $\varphi_k : R^I \to R^I$.

We adopt notions and notations from dynamical systems. That is, the functions φ_k are iterated maps. Let $F : R^I \to R^I$ be any function. Then, inductively define

$$F^0(z) =_{\text{def}} z, \qquad F^k(z) =_{\text{def}} F(F^{k-1}(z)) \text{ for } k > 0.$$

So, $\varphi_k = F^k$.

The following summarizes the notions schematically:

$$\begin{array}{ccccccccccc}
x(0) & \to & x(1) & \to & x(2) & \to & \dots & \to & x(k) & \to & \dots \\
\downarrow & & \downarrow & & \downarrow & & & & \downarrow & & \\
z_0 & \to & z_1 & \to & z_2 & \to & \dots & \to & z_k & \to & \dots \\
\downarrow & & \downarrow & & \downarrow & & & & \downarrow & & \\
\varphi_0(z_0) & \to & \varphi_1(z_0) & \to & \varphi_2(z_0) & \to & \dots & \to & \varphi_k(z_0) & \to & \dots \\
\downarrow & & \downarrow & & \downarrow & & & & \downarrow & & \\
F^0(z_0) & \to & F^1(z_0) & \to & F^2(z_0) & \to & \dots & \to & F^k(z_0) & \to & \dots
\end{array}$$

process
trajectory
dynamic
iterated map

Notice that iterated maps describe memory-less dynamics. In this sense, they are deterministic versions of Markov chains.

2.4.1 Iterated maps

We investigate dynamics induced by iterating a map $F : D^n \to D^n$ where, more commonly, D denotes the domain of F (i.e., D corresponds to the range of an attribute) and n is the number of items or dyads.

A fundamental concept in the study of iterated maps is the orbit.

Definition 2.12 *Let $F : D^n \to D^n$ be a total mapping. Then, the* orbit of z_0 under F *is defined to be the sequence $(z_0, z_1, z_2, \dots, z_k, \dots)$ such that $z_k = F^k(z_0)$ for all $k \in \mathbb{N}$.*

An orbit is a specific trajectory.

Example: We discuss some examples of iterated maps and orbits.

- Let $D = \mathbb{N}$, $n = 1$, and $F(x) =_{\text{def}} x + 1$. The orbit of 0 is $(0, 1, 2, 3, \dots)$.
- Let $D = \mathbb{R}_{\geq 0}$, $n = 1$, and $F(x) =_{\text{def}} \frac{x}{x+1}$. The orbit of 1 is $(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots)$.
- A prominent iterated map is the Bernoulli shift: Let $D = [0, 1)$, $n = 1$, and $F : D \to D : x \mapsto 2x \bmod 1$, i.e.,

$$F(x) =_{\text{def}} \begin{cases} 2x & \text{if } 0 \leq x < \frac{1}{2} \\ 2x - 1 & \text{if } \frac{1}{2} \leq x < 1 \end{cases}$$

  Suppose $x \in [0, 1)$ is given in binary expansion as $0.a_1a_2a_2 \dots$, $a_i \in \{0, 1\}$, i.e., it holds that $x = \sum_{i=1}^{\infty} a_i 2^{-i}$. We have two cases:
  - *Case* $a_1 = 0$: That is, $x < \frac{1}{2}$. Applying F to x gives

$$\begin{aligned} F\left(\sum_{i=1}^{\infty} a_i 2^{-i}\right) = 2\sum_{i=1}^{\infty} a_i 2^{-i} &= \sum_{i=1}^{\infty} a_i 2^{-(i-1)} \\ &= \sum_{i=2}^{\infty} a_i 2^{-(i-1)} = \sum_{i=1}^{\infty} a_{i+1} 2^{-i} \end{aligned}$$

– *Case* $a_1 = 1$: That is, $x \geq \frac{1}{2}$. Analogously to the first case, applying F to x gives

$$\begin{aligned} F\left(\sum_{i=1}^{\infty} a_i 2^{-i}\right) = 2\sum_{i=1}^{\infty} a_i 2^{-i} \; - 1 \;\; &= \;\; \sum_{i=1}^{\infty} a_i 2^{-(i-1)} \; - 1 \\ &= \;\; 1 + \sum_{i=2}^{\infty} a_i 2^{-(i-1)} \; - 1 \\ &= \;\; \sum_{i=2}^{\infty} a_i 2^{-(i-1)} = \sum_{i=1}^{\infty} a_{i+1} 2^{-i} \end{aligned}$$

Hence, $F(0.a_1a_2a_3\ldots) = 0.a_2a_3a_4\ldots$. The orbit of $x_0 = 0.a_1a_2a_3\ldots$ is the sequence $(x_k)_{k\in\mathbb{N}}$ such that $x_k = 0.a_{k+1}a_{k+2}a_{k+3}\ldots$.

- Let $D = \{0,1\}, n = \|\{\ (i,j) \in \mathcal{I} \mid i < j\ \}\|$, and $\mathcal{I} = A \times A \setminus \{\ (i,i) \mid \in A\ \}$ for some set A be given. That is, we consider an interaction domain representing a complete undirected graph. So, n is the number of edges, i.e., $n = \binom{\|A\|}{2}$. Suppose edges are lexicographically enumerated. Then, $x \in D^n$ encodes an undirected graph.

  Define $F : D^n \to D^n$ to be the mapping that satisfies

$$\begin{aligned} F(x)_{\max\{k|x_k=1\}} &= 0 \qquad \text{if } x \text{ contains a cycle} \\ F(x)_{\min\{k|x_k=0\}} &= 1 \qquad \text{if } x \text{ contains no cycle} \end{aligned}$$

  with all other components of $F(x)$ unchanged compared to x. Then, examples of orbits for $n = 4$ are the following:

  ...

Proposition 2.13 *Let $F : D^n \to D^n$ be a total mapping, and let $x, y \in D^n$. Then, the orbits of x and y under F are either disjoint or there exist $k \in \mathbb{N}$ and $r \in \mathbb{Z}$ such that $F^{k'}(x) = F^{k'+r}(y)$ for all $k' \geq k$.*

Proof: Suppose the orbits of x and y are not disjoint, i.e., there are $t, t' \in \mathbb{N}$ such that $F^t(x) = F^{t'}(y)$. Define $r =_{\text{def}} t' - t$. So, $t' = t + r$. Then, by induction on $\ell \in \mathbb{N}$, we obtain that $F^{t+\ell}(x) = F^{t+r+\ell}(y)$ for all $\ell \in \mathbb{N}$:

- *Base of induction* $\ell = 0$: Then, $F^{t+0}(x) = F^t(x) = F^{t'}(y) = F^{t+r+0}(y)$.
- *Inductive step* $\ell > 0$: By the induction assumption we conclude that

$$F^{t+\ell}(x) = F(F^{t+\ell-1}(x)) = F(F^{t+r+\ell-1}(y)) = F^{t+r+\ell}(y).$$

Hence, setting $k =_{\text{def}} t$ and $k' =_{\text{def}} t + \ell$ proves the proposition. ■

2.4.2 The phase space

Given a map $F : D^n \to D^n$, all orbits under F are collected in the phase space. The fundamental problem (in statistical mechanics) is getting knowledge on the probability distribution over the phase space, i.e., to determine the visiting probablity of a certain state in an orbit.

The following concepts are essential for addressing this question.

Definition 2.14 *Let $F : D^n \to D^n$ be a total mapping.*

1. *A state $x \in D^n$ is called* fixed point of F *if and only if $F(x) = x$.*
2. *A state $x \in D^n$ is called* periodic under F *if and only if there exists a $k \in \mathbb{N}_+$ such that $F^k(x) = x$. The number $k_0 \in \mathbb{N}_+$ minimal subject to $F^{k_0}(x) = x$ is called the* periodic order *of x, and x is then called* periodic of order k_0.
3. *A state $x \in D^n$ is called* transient under F *if and only if $F^k(x) \neq x$ or all $k \in \mathbb{N}_+$, i.e., x is not periodic.*

Obviously, a *fixed point* is a periodic state of order 1.

Example: We give examples for each part of the definition.

- For the Bernoulli shift, 0 is the only fixed point, $\frac{2}{3}$ is an example of a periodic state of order 2, and $\frac{1}{\sqrt{2}}$ is an example of a transient state.
- Consider again the map $F : \{0,1\}^n \to \{0,1\}^n$ from above for $n = 4$. Then, there is no fixed point of F, ... are periodic of order 2, and ... is transient.

The following proposition explains why recurring states are referred to as "periodic."

Proposition 2.15 *Let $F : D^n \to D^n$ be a total function. Let $x \in D^n$ be a periodic state of order k_0, and let $k \in \mathbb{N}$. Then, the following holds:*

$$F^k(x) = x \iff k_0 \text{ divides } k$$

Proof: We prove both directions individually.

($\Leftarrow$) Observe that $x = F^{k_0}(x) = F^{k_0}(F^{k_0}(x))$. An easy inductive argument shows that $x = F^{c \cdot k_0}(x)$ for all $c \in \mathbb{N}$. Hence, if k_0 divides k, i.e., $k = c \cdot k_0$ for some $c \in \mathbb{N}$, then $F^k(x) = x$.

($\Rightarrow$) Case $k = 0$ is trivial. Now, suppose $k \geq k_0 > 0$. Then, $k = c \cdot k_0 + r$ for uniquely determined $c \in \mathbb{N}_+$ and $r \in \{0, 1, \ldots, k_0 - 1\}$. Thus,

$$x = F^k(x) = F^{c \cdot k_0 + r}(x) = F^r(F^{c \cdot k_0}(x)) = F^r(x)$$

Since k_0 is the smallest positive number with this property, it follows that $r = 0$. Hence, $k = c \cdot k_0$. So, k_0 divides k.

This proves the proposition. ■

Definition 2.16 *Let $F : D^n \to D^n$ be a total mapping. Let x be a periodic state of order k. Then, the set $\{x, F(x), F^2(x), \ldots, F^{k-1}(x)\}$ is called a* limit cycle *(of length k) of F.*

Limit cycles (of length k) are also called *attractors* (of length k). The limit cycle corresponding to a fixed point is also called *singleton attractor*.

Any easy consequence of Proposition 2.13 is that limit cycles are either disjoint or identical.

Corollary 2.17 *Let $F : D^n \to D^n$ be a map. If $\{x_1, \ldots, x_\ell\}$ and $\{y_1, \ldots, y_r\}$ are two limit cycles of F such that $\{x_1, \ldots, x_\ell\} \cap \{y_1, \ldots, y_r\} \neq \emptyset$ then $\{x_1, \ldots, x_\ell\} = \{y_1, \ldots, y_r\}$.*

For finite domains, orbits have a simple structure.

Proposition 2.18 *Let $F : D^n \to D^n$ be a map over a finite domain D. Let $(x_i)_{i \in \mathbb{N}}$ be the orbit of $x_0 \in D^n$ under F. Then, there are $k_0 \in \mathbb{N}$ and $\ell_0 \in \mathbb{N}_+$ such that*

(a) $\{x_0, \ldots, x_{k_0 - 1}\}$ is the set of k_0 transient states of the orbit of x_0 under F and

(b) $\{x_{k_0}, \ldots, x_{k_0 + \ell_0 - 1}\}$ is a limit cycle of length ℓ_0 of F.

Proof: Let $(x_i)_{i \in \mathbb{N}}$ be the orbit of $x_0 \in D^n$ under F, i.e., $x_i = F^i(x_0)$. D is finite, so is D^n. Thus, there are $k \geq 0$ and $\ell > 0$ such that $F^k(x_0) = x_k = x_{k+\ell})F^{k+\ell}(x_0)$. Define parameters k_0 and ℓ_0 as follows (in this order):

$$\begin{aligned} k_0 &=_{\text{def}} \min \{\ k \mid F^k(x_0) = F^{k+r}(x_0) \text{ for some } r > 0\ \} \\ \ell_0 &=_{\text{def}} \min \{\ r \mid F^k(x_0) = F^{k_0 + r}(x_0)\ \} \end{aligned}$$

Then, for all $r > 0$, it holds that

$$F^{k_0 + r}(x_0) = F^r\left(F^{k_0}(x_0)\right) = F^r\left(F^{k_0 + \ell_0}(x_0)\right) = F^{k_0 + \ell_0 + r}(x_0).$$

Hence, x_i is periodic of order ℓ_0 if $i \geq k_0$, which is tatement (b), and x_i is transient if $i < k_0$, which is statement (a). This proves the Proposition. ■

The orbit under a iterated, finite-domain map can be visualized by the following transition diagramm:

...

In principle, iterated maps can be studied graph-theoretically. A map $F : D^n \to D^n$ over a finite domain D can be associated with the directed graph $\Gamma(F) = (V, E)$, called *state graph of* F, where

$$V =_{\text{def}} D^n, \qquad E =_{\text{def}} \{\ (x, F(x)) \mid x \in D^n\ \}.$$

Note that $\Gamma(F)$ might have loops.

According to Proposition 2.13, Corollary **??**, and Proposition **??**, the state graph of F can be uniquely decomposed into

- disjoint cycles $C_1, \ldots, C_k$ (representing limit cycles) and
- disjoint (directed) trees $T_1, \ldots, T_r$ (representing transient states) each of which is incident with exactly one cycle $C_1, \ldots C_k$

  Example: We consider the map

  $$F : \{0,1\}^3 \to \{0,1\}^3 : (x_1, x_2, x_3) \mapsto (x_2 \oplus x_3, 1 \oplus x_1 \oplus x_3, x_! \oplus x_2)$$

  where $\oplus$ denotes XOR or, equivalently, addition modulo 2. In order to determine the state graph of F, we first represent F as a truth table:

(x_1, x_2, x_3)	$F(x_1, x_2, x_3)$
000	010
001	100
010	111
011	001
100	001
101	111
110	100
111	010

  From this, we easily obtain the state graph of F:

  ...

A cycle together with all its incident trees is called *basin of attraction*.

More formally, let $F : D^n \to D^n$ be a map over a finite domain D. A set $E \subseteq D^n$ is called *invariant set* if and only if for alle $k \in \mathbb{N}$, $F^k(E) \subseteq E$. Each invariant set contsains a limit

cycle. So, D^n can be uniquely decomposed into r invariant sets where r is the number of limit cycles of F. A basin of attraction is one component of this decomposition.

It is clear that transient states have visiting probability zero. The following proposition gives the precise visiting probability of a periodic state in terms of the structure of its corresponding basin of attraction.

Proposition 2.19 *Let $F : D^n \to D^n$ be a map over a finite domain D, $\|D\| = m$. Let $z \in D^n$ be periodic, and let $E \subseteq D^n$ be the basin of attraction of (the limit cycle of) z. Suppose E consists of s transient and r perdiodic states. Then, the visiting probability of z in a random orbit is*

$$\left(1 + \frac{s}{r}\right) \cdot m^{-n}.$$

Proof: Let $x \in D^n$ be an arbitrary state. Consider the orbit $(x_i)_{i\in\mathbb{N}}$ such that $x = x_0$ and $F^k(x_0) = x_k$ for all $k > 0$. Suppose $x_0, \ldots, x_{k_0-1}$ are all transient states and $x_{k_0}, \ldots, x_{k_0+r-1}$ are all periodic states (of order r). Let $z \in D^n$ be a state in the orbit $(x_i)_{i\in\mathbb{N}}$. Define

$$P_z =_{\text{def}} \mathbf{P}\left[z \text{ is visited in } (x_i)_{i\in\mathbb{N}}\right].$$

Then, P_z is given by a frequency sequence of the initial segments of the orbit:

$$P_z = \lim_{N\to\infty} \frac{\|\{\ i \mid i \in \{0, 1, \ldots, N-1\} \text{ and } x_i = z\ \}\|}{N}$$

To calculate P_z, we have two cases.

- Suppose z is transient. Thus, $\|\{\ i \mid i \in \{0, 1, \ldots, N-1\} \text{ and } x_i = t\ \}\| = 1$ for $N \geq k_0$. Hence,

  $$P_z = \lim_{N\to\infty} \frac{1}{N} = 0.$$

- Suppose z is perdiodic. Thus, for $N \geq k_0 + 1$,

  $$\left\lfloor \frac{N - k_0 - 1}{r} \right\rfloor \leq \|\{\ i \mid i \in \{0, 1, \ldots, N-1\} \text{ and } x_i = t\ \}\| \leq \left\lfloor \frac{N - k_0 - 1}{r} \right\rfloor + 1.$$

  Hence, we obtain

  $$\begin{aligned} P_z &\leq \lim_{N\to\infty} \frac{\frac{N-k_0-1}{r} + 1}{N} = \lim_{N\to\infty} \frac{N - k_0 - 1 + r - 1}{rN} = \frac{1}{r} \\ P_z &\geq \lim_{N\to\infty} \frac{1 + \frac{N-k_0-1}{r} - 1}{N} = \lim_{N\to\infty} \frac{N - k_0 - 1 - r}{rN} = \frac{1}{r} \end{aligned}$$

  Consequently, $P_z = \frac{1}{r}$.

Now, consider any periodic $z \in D^n$ following the specification given in the proposition. Then, z lies on $s + r$ orbits. So, the visiting probability of z is

$$\mathbf{P}[z \text{ is visited in some orbit}] = \frac{s+r}{m^n} \cdot \frac{1}{r} = \left(1 + \frac{s}{r}\right) \cdot m^{-n}.$$

This proves the proposition. ■

Example (cont'd): Consider the map $F : \{0,1\}^3 \to \{0,1\}^3$ from above. The visiting probabilities for the periodic states $010, 111, 001, 100$ are all $\frac{1}{4}$.

2.4.3 Series, levels, and plots

Even for finite domains D, the state graph $\Gamma(F)$ for an iterated map $F : D^n \to D^n$ can be too large to construct explicitly. Since $\|V\| = \|E\| = \|D\|^n$, it has size exponential in the number n. If we consider boolean network attributes then the size $2^{O(n^2)}$. This forces us to reduce the dimensionality of the phase space. In this subsection, we look at three methods to achieve this reduction.

(Multivariate) Time series

Let $F : D^n \to D^n$ be a map. Let $\tau : D^n \to \mathbb{R}$ be any function. Then, the *time series* $(\tau_i)_{i \in \mathbb{N}}$ associated with an orbit $(x_i)_{i \in N}$ is given by

$$\tau_i = \tau(x_i) = \tau(F^i(x_0)).$$

The following summarizes the construction of a time series schematically:

$$\begin{array}{ccccccccccc} x_0 & \xrightarrow{F} & x_1 & \xrightarrow{F} & x_2 & \xrightarrow{F} & \dots & \xrightarrow{F} & x_k & \xrightarrow{F} & \dots \\ \tau \downarrow & & \tau \downarrow & & \tau \downarrow & & & & \tau \downarrow & & \\ \tau_0 & \to & \tau_1 & \to & \tau_2 & \to & \dots & \to & \tau_k & \to & \dots \end{array}$$

orbit

time series

If $\tau : D^n \to \mathbb{R}^m$ has m-dimensional function values then the sequence $(\tau_i)_{i \in \mathbb{N}}$ is called *multivariate* (or *multidimensional*) *time series*.

Example: A typical example of derived time series from an underlying dynamics is the evolution of the voting distribution among voters over some time period. (We will study this in the opinion dyamics section.)

For a more technical example, consider the following state graph:

...

Several time series can be derived from the orbit of this state graph.

- Define $\tau_1 : \{0,1\}^4 \to \mathbb{R} : (x_1, x_2, x_3, x_4) \mapsto |(x_1, x_2, x_3, x_4)|_1$ (where $|x|_a$ denotes the number of a's in a tuple $x \in \{0,1\}^n$)

- For a bivariate times series, define

  $$\tau_2 : \{0,1\}^4 \to \mathbb{R}^2 : (x_1, x_2, x_3, x_4) \mapsto (|(x_1, x_2, x_3, x_4)|_0, |(x_1, x_2, x_3, x_4)|_1)$$

- Define $\tau_3 : \{0,1\}^4 \to \mathbb{R} : (x_1, x_2, x_3, x_4) \mapsto x_1 \oplus x_2 \oplus x_3 \oplus x_4$ (i.e., τ_3 is a parity function)
- Define $\tau_3 : \{0,1\}^4 \to \mathbb{R} : (x_1, x_2, x_3, x_4) \mapsto x_2$ (i.e., τ_4 is a projection function)

Levels

Let $F : D^n \to D^n$ be a map. A subset $L \subseteq \{1, \ldots, n\}$ of size $\|L\| = m$ is called *level of* f if and only if $L = \emptyset$ or there is a map $G : D^m \to D^m$ such that for all $x \in D^n$,

$$\pi(F(x)) = G(\pi(x)),$$

where $\pi : D^n \to D^m$ is the projection that maps $(x_1, \ldots, x_n)$ to those m components indexed by the set L, i.e., $\pi(x_1, \ldots, x_n) = (x_{i_1}, \ldots, x_{i_m})$ and $L = \{i_1, \ldots, i_m\}$.

Note that by an inductive argument, it follows that:

$$\pi(F^k(x)) = G^k(\pi(x)) \quad \text{for all } k \in \mathbb{N}$$

Indeed, case $k = 0$ is trivial, and for the case $k > 0$, we obtain by using the inductive assumption

$$\pi(F^k(x)) = \pi(F(F^{k-1}(x)) = G(\pi(F^{k-1}(x))) = G(G^{k-1}(\pi(x))) = G^k(\pi(x)).$$

So, the iterated map G induces a subdynamic on the elements of the level L.

This can be summarized schematically as follows:

$$\begin{array}{ccccccccccccc}
x_0 & \xrightarrow{F} & x_1 & \xrightarrow{F} & x_2 & \xrightarrow{F} & \ldots & \xrightarrow{F} & x_k & \xrightarrow{F} & \ldots & & \text{orbit} \\
\pi\downarrow & & \pi\downarrow & & \pi\downarrow & & & & \pi\downarrow & & & & \\
y_0 & \xrightarrow{G} & y_1 & \xrightarrow{G} & y_2 & \xrightarrow{G} & \ldots & \xrightarrow{G} & y_k & \xrightarrow{G} & \ldots & & \text{orbit}
\end{array}$$

Example: We determine all levels for the state graph $\Gamma(F)$ above:

- $\{2,3,4\}$ is a level of F with the map $G : \{0,1\}^3 \to \{0,1\}^3$
  ...
- $\{3,4\}$ is a level of F (and of G) with the map $G' : \{0,1\}^2 \to \{0,1\}^2$
  ...

There are no other levels of F. As an example, consider the set $\{1,3,4\}$. The sequence $(\pi(F^i(..,..,..)))_{i \in \mathbb{N}}$

The DERRIDA *plot*

This is a method for identifying turbulent behavior in the phase space.

Consider a map $F : \{0,1\}^n \to \{0,1\}^n$. Let $d_H(x,y)$ denote the HAMMING distance between $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$, i.e.,

$$d_H(x,y) =_{\text{def}} \|\{\ i \mid x_i \neq y_i\ \}\|.$$

The DERRIDA relation $\mathcal{D}(F)$ consists of the following multiset

$$\mathcal{D}(F) =_{\text{def}} \{\ (h_1,h_2) \quad \mid \quad \begin{array}{l}\text{there are states } x,y \text{ such that } d_H(x,y) = h_1 \\ \text{and } d_H(F(x),F(y)) = h_2\ \}\end{array}$$

The multiplicity of the pairs (h_1, h_2) is given by the number of pairs (x, y) that realize the values specified by h_1 and h_2.

Now, we can plot the relation $\mathcal{D}(F)$ as a diagram (with an appropriate representation of multiplicities).

> **Example:** We consider the following three maps for $n = 4$: Then, plot might look like as follows
>
> The curves are interpolations of resulting distances.

The intuition behind the DERRIDA plot is the following: The more pairs above the diagonal, the more chaos.

Using this intuition we can introduce a numerical measure for chaos. The measure is based on linear regression. Suppose $(x_1, y_1), \ldots, (x_N, y_N)$ are pairs in $\mathcal{D}(F)$ with multiple occurrences corresponding up to their multiplicities. We try to find a linear function βx which minimizes the distances to all pairs in the list, i.e., we want to find the right slope β. According to the methode of least squares we define

$$L(\beta) =_{\text{def}} \sum_{i=1}^{N} (y_i - \beta x_i)^2.$$

By taking derivatives, we obtain

$$L'(\beta) = \sum_{i=1}^{N} 2(y_i - \beta x_i)(-x_i), \qquad L''(\beta) = \sum_{i=1}^{N} 2x_i^2 > 0$$

Therefore, any zero of L' minimizes L. Hence, the optimale slope is

$$\beta = \frac{\sum_{i=1}^{N} y_i x_i}{\sum_{i=1}^{N} x_i^2}$$

The DERRIDA coefficient $\mathrm{Dc}(F)$ of a map is a scaled version of β:

$$\mathrm{Dc}(F) =_{\mathrm{def}} \log_2 \beta$$

It is obvious that the DERRIDA coefficient ranges between $-\infty$ and ∞. The coefficient can be used to make distinction between different behaviors of iterated maps. The interpretations for F are as follows:

$$\begin{aligned} \mathrm{Dc}(F) \gg 0 &: F \text{ shows } \textit{chaotic} \text{ behavior} \\ \mathrm{Dc}(F) \approx 0 &: F \text{ shows } \textit{critical} \text{ behavior} \\ \mathrm{Dc}(F) \ll 0 &: F \text{ shows } \textit{frozen} \text{ behavior} \end{aligned}$$

The classification into these three types of behaviors becomes clear by looking at the following diagrams:

2.4.4 Local maps

An iterated map $F : D^n \to D^n$ describes a global behavior:

$$\begin{aligned} x_1 \leftarrow F(x_1, x_2, \ldots, x_n)_1 &= f_1(x_1, x_2, \ldots, x_n) \\ x_2 \leftarrow F(x_1, x_2, \ldots, x_n)_2 &= f_2(x_1, x_2, \ldots, x_n) \\ \vdots & \\ x_n \leftarrow F(x_1, x_2, \ldots, x_n)_n &= f_n(x_1, x_2, \ldots, x_n) \end{aligned}$$

global map local maps

In many cases, only local descriptions are available or even observable.

Let $f : D^n \to D$ be any function. A variable x_j (or an index j) is *fictive in* f if and only if

$$f(z_1, \ldots, z_{j-1}, z_j, z_{j+1}, \ldots z_n) = f(z_1, \ldots, z_{j-1}, z'_j, z_{j+1}, \ldots z_n)$$

for all $z_1, \ldots, z_{j-1}, z_{j+1}, \ldots z_n \in D$, $z_j, z'_j \in D$. Given a collection of functions (local maps) $f_1, \ldots, f_n : D^n \to D$, we say that x_i *depends on* x_j *in* f_i iff x_j is not fictive in f_i.

The *interdependence graph* of an iterated maps $F : \{0,1\}^n \to \{0,1\}^n$ (considered as the collection of its n local maps) is defined to consist of

- vertex set $V =_{\mathrm{def}} \{1, \ldots, n\}$ and
- edge set $E = \{\ (i,j) \mid x_i \text{ depends on } x_j \text{ in } F\ \}$

We also uses undirected versions without loops.

Example: Let $F : \{0,1\}^3 \to \{0,1\}^3$ be given by the following truth table.

(x_1, x_2, x_3)	$F(x_1, x_2, x_3)$
000	100
001	101
010	011
011	011
100	100
101	101
110	101
111	101

Then, x_1 depends on x_1, x_2, x_2 depends on x_1, x_2, and x_3 depends on x_2, x_3. So, the interpendence graph of F is as follows:

We briefly discuss the connection between the interdependence graph of an iterated network map and its underlying interaction domain. Suppose F is a map on an attribute x on a symmetrical interaction domain $\mathcal{I} \subseteq A \times A$.

1. Suppose x is an attribute on items, i.e., $x : A \to R$. Then, x_i depends on all x_j such that $\{i, j\} \in \mathcal{I}$. So, the (undirected) interdependence graph of F is a subgraph of $\mathcal{I}$.

2. Suppose x is an attribute on dyads, i.e., $x : \mathcal{I} \to R$. Then, x_{ij} depends on all x_e such that $e \in \mathcal{I}$ and $e \cap \{i, j\} \neq \emptyset$. So, the interdependence graph of F is a subgraph of $L(\mathcal{I})$, where $L(G)$ is the *line graph* of an undirected graph $G = (V, E)$, i.e., $L(G) = (V', E')$ such that

$$V' =_{\text{def}} E, \qquad E' =_{\text{def}} \{\ (e, f) \mid e, f \in E \text{ and } e \cap f = \emptyset\ \}$$

So far, we have considered a given iterated network map F decomposed into a collection of local network maps. We now turn our point of view and consider iterated network maps composed by a given collection of local network maps $f_1, \ldots, f_n$.

Suppose $x : \mathcal{I} \to R$ is a network attribute. Assume that $\mathcal{I}$ is enumerated $1, \ldots, n$. Let $M = \{\ f_i \mid i \in \mathcal{I}\ \}$ be a set of *local transitions* $f_i : D^{\deg_i + 1} \to D$ where $\deg_i$ denotes the in-degree of i in the interdependence graph. Additionally, suppose that we are given a mapping $\alpha : \{1, \ldots T\} \to \mathcal{P}(\mathcal{I})$ which is called *schedule*; the parameter $T > 0$ is any natural number.

For each $i \in \mathcal{I}$ and for each subset $U \subseteq \mathcal{I}$, *activity function* $\varphi_i[U]$ is defined for configuration $\vec{z} = (z_1, \ldots, z_n) \in D^n$ by

$$\varphi_i[U](\vec{z}) =_{\text{def}} \begin{cases} f_i(z_{i_1}, \ldots, z_{i_{\deg_i + 1}}) & \text{if } i \in U \\ z_i & \text{if } i \notin U \end{cases}$$

where $\{i_1, i_2, \ldots, i_{\deg_i+1}\}$ is the set of neighbors in the interdependence graph.

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

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

Finally, the *global network map* $\mathbf{F}_{(M,\alpha)} : D^n \to D^n$ *induced by* (M, α) is defined by

$$\mathbf{F}_{(M,\alpha)} =_{\text{def}} \prod_{k=1}^{T} \mathbf{F}_M[\alpha(k)],$$

i.e., $\mathbf{F}_{(M,\alpha)}$ is defined by the composition of global transition functions 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

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

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

Example: Suppose $\mathcal{I} = L(\mathcal{I}) = K^3$. Let $M = \{f_1, f_2, f_3\}$ consist of local transitions $f_i\{0,1\}^3 \to \{0,1\}$ such that for each $i \in \{1,2,3\}, z_1, z_2, z_3 \in \{0,1\}$

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

Let $U_1 = \{1,2\}$ and $U_2 = \{1,2,3\}$ be subsets of $\mathcal{I}$. Then we obtain the following activity functions:

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

The global transition function looks as follows:

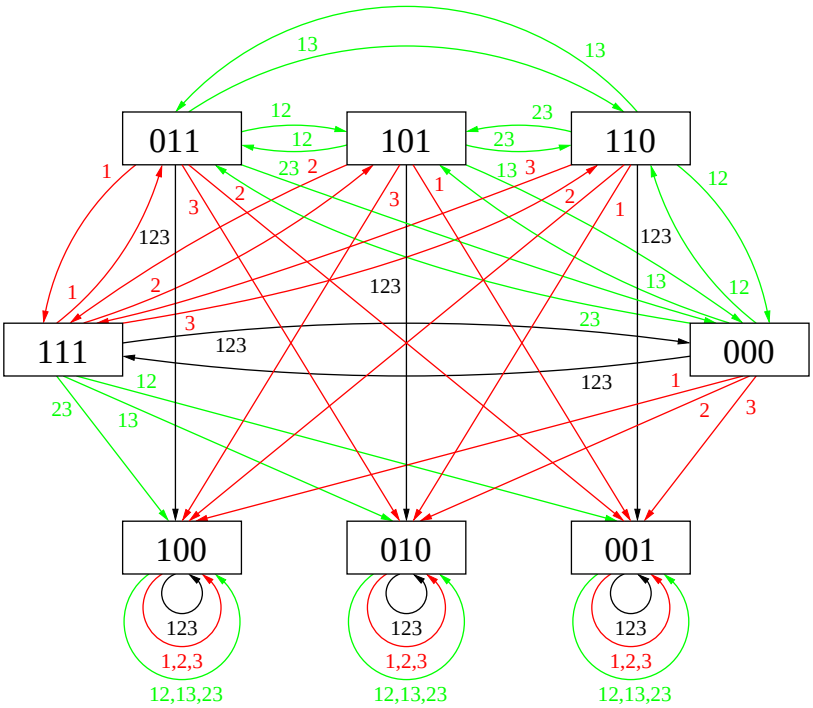
- $\mathbf{F}_M[U_1](z_1, z_2, z_3) = (f_1(z_1, z_2, z_3), f_2(z_1, z_2, z_3), z_3)$; concrete function values are, e.g.,

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

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

$$\mathbf{F}_M[U_2](1,1,1) = (0,0,0)$$
$$\mathbf{F}_M[U_2](0,0,0) = (1,1,1)$$

The following figure visualizes the global transition functions completely:



For the update schedules $\alpha_1 : \{1\} \to \mathcal{P}(\{1,2,3\})$ and $\alpha_2 : \{1,2,3\} \to \mathcal{P}(\{1,2,3\})$ given by

$$\alpha_1 : \{1\} \mapsto \{1,2,3\}, \qquad \alpha_2 : \begin{array}{l} \{1\} \mapsto \{2\} \\ \{2\} \mapsto \{1\} \\ \{3\} \mapsto \{3\} \end{array}$$

the induced network maps are as follows:

(x_1, x_2, x_3)	$F(x_1, x_2, x_3)$
000	111
001	001
010	010
011	100
100	100
101	010
110	001
111	000

(x_1, x_2, x_3)	$F(x_1, x_2, x_3)$
000	010
001	001
010	010
011	001
100	100
101	010
110	100
111	001

Simulation 3

3.1 Agent-based modelling

3.2 The agency problem

3.2.1 Push or pull?

3.2.2 Dyads or actors?

3.3 Sequential dynamical systems*

Theory loves synchronous schedules, computers love sequential schedules.

A sequential dynamical system results from a dynamical system on n items or dyads by replacing each update step with a sequence of n update steps in which only one item or dyad is allowed to update its state. Of course, the coice of the schedule matters. For instance, each item or dyads should appear in the schedule.

3.3.1 Permutation schedules

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.

3.3.2 Functional equivalence

Definition 3.1 *Let L be a set of local transitions on X, $\|X\| = n$. Let $\pi, \pi' : X \to X$ be permutations 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 3.2 *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. ∎

3.3.3 The 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 $U = U(X, E)$:

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

Proposition 3.3 *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 $\vec{z} \in D^n$

$$
\begin{aligned}
&\mathbf{F}_{(L,\pi)}(\vec{z}, n) \\
&= \left(\prod_{j=1}^{n} \mathbf{F}_L[\pi(j)] \right)(\vec{z}) \\
&= \left(\prod_{j=1}^{k-1} \mathbf{F}_L[\pi(j)] \cdot \mathbf{F}_L[\pi(k)] \cdot \mathbf{F}_L[\pi(k+1)] \cdot \prod_{j=k+2}^{n} \mathbf{F}_L[\pi(j)] \right)(\vec{z}) \\
&= \left(\prod_{j=1}^{k-1} \mathbf{F}_L[\pi(j)] \cdot \mathbf{F}_L[\pi(k+1)] \cdot \mathbf{F}_L[\pi(k)] \cdot \prod_{j=k+2}^{n} \mathbf{F}_L[\pi(j)] \right)(\vec{z}) \\
&\qquad\qquad \text{(by Proposition 3.2)} \\
&= \left(\prod_{j=1}^{k-1} \mathbf{F}_L[\pi_{d-1}(j)] \cdot \mathbf{F}_L[\pi_{d-1}(k)] \cdot \mathbf{F}_L[\pi_{d-1}(k+1)] \cdot \prod_{j=k+2}^{n} \mathbf{F}_L[\pi_{d-1}(j)] \right)(\vec{z}) \\
&= \left(\prod_{j=1}^{n} \mathbf{F}_L[\pi_{d-1}(j)] \right)(\vec{z}) \\
&= \mathbf{F}_{(L,\pi_{d-1})}(\vec{z}, n) \\
&= \mathbf{F}_{(L,\pi')}(\vec{z}, n) \qquad \text{(by induction assumption)}
\end{aligned}
$$

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

$$f_G \; : \; S_X / \sim_U \; \rightarrow \; \text{Acyc}(G),$$

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

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

$$i \leq_\pi j \Longleftrightarrow_{\text{def}} \pi(i) \leq \pi(j).$$

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

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

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 $f_G : S_X/\sim_U \to \text{Acyc}(G)$ by $f_G([\pi]_U) =_{\text{def}} \tilde{f}_G(\pi)$. 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:

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

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

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

and for $h \in H$

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

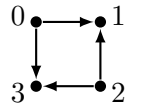
where $\text{rank}(i_j) = h$ and $i_j < i_k$ for $j < k$. Furthermore, consider

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

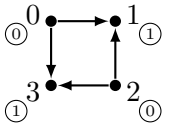
with $t = \max H$. Then, clearly, f_G maps $\big[\big(\text{rnk}^{-1}(0), \ldots, \text{rnk}^{-1}(t)\big)\big]_U$ to the given orientation. Thus, f_G is surjective. Hence, f_G is bijective. ■

Example: Consider Circ_4

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



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



Proposition 3.5 *For any undirected graph $G = (X,E)$, $\|X\| = n$, and any set L of local transition functions with interdepence structure E,*

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

and the bound is sharp.

Proof: Using Proposition 3.3 and Proposition 3.4, 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.

3.3.4 Acyclic orientations and the chromatic polynomial

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

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 3.6 *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 3.7 *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 3.8 *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 3.8 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 3.9 *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 3.10 (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 3.9 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 3.11 $\|\text{Acyc}(G)\| = (-1)^n \; P_G(-1)$.

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

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

- *Induction 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 3.11, 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 3.12 *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.*

3.4 Ensemble approaches

Models 4

4.1 Potentials

Models based on maximizing a potential function are typical for rational actors (aka agents). They belong to a class of mechanisms where actors choose their decisions on changing attribute values depending on the decisions of other actors in such a way they maximize their benefits, utilities, or preferences. In the following, we use game theory to analyze such models.

4.1.1 Games with utility functions

Definition 4.1 *A* game with utilities Γ *is a triple* $(A, (S_1, \ldots, S_m), (u_1, \ldots, u_m))$*, where*

1. $A = \{1, \ldots, m\}$ *is a finite, non-empty set of* agents*,*
2. S_i *is a non-empty set of* strategies *of agent* $i \in A$*, and*
3. $u_i : S_1 \times \cdots \times S_m \to \mathbb{R}$ *is a utility function for agent* i*.*

According to the definiton above, we introduce some notations:

- $S =_{\text{def}} \bigtimes_{k=1}^{m} S_k$ denotes the set of all strategy profiles of all agents; $S_{-i} =_{\text{def}} \bigtimes_{\substack{k=1 \\ k \neq i}}^{m} S_k$ denotes the set of all strategy profiles of all agents except agent i.
- For a strategy profile $s = (s_1, \ldots, s_m) \in S$, let s_{-i} denote the $(m-1)$-tuple consisting of strategies of all agents except agent i, i.e., $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_m)$.
- So, $s = (s_i, s_{-i})$ and $S = S_i \times S_{-i}$, by convention.
- We use $u = (u_1, \ldots, u_m) : S \to \mathbb{R}^m$ to denote the vector utility function, and we use $u_i(s) = u_i(s_1, \ldots, s_m) = u_i(s_i, s_{-i})$ to denote agent's i utility of a strategy profile

We consider a game Γ as a *one-shot non-cooperative game*. Each agent u chooses a strategy $s_i \in S_i$ independently of other agents and without knowing the choices of the other agents. The result is a strategy profile $s = (s_1, \ldots, s_m)$. Each agent i evaluates strategy profile s according to the utility function u_i.

A notion central to game theory is the Nash equilibrium.

Definition 4.2 *Let $\Gamma = (A, S, u)$ be a game with utilities, involving m agents. A strategy profile $s^* = (s_1^*, \ldots, s_m^*)$ is called* Nash equilibrium *if and only if $u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$ for all $s_i \in S_i$ and all $i \in A$.*

Intuitively, in a Nash equilibrium, no agent has an incentive to deviate from the chosen strategy.

Example: We exemplify the notions for three standard games.

- *Battle of sexes*: Male M and Female F want to spend time together, i.e., $A = \{M, F\}$. Alternatives are cinema (c) or football (f). So, the sets of strategies for both are $S_M = S_F = \{c, f\}$. The set of strategy profiles is

$$S = S_F \times S_M = \{\ (c,c), (c,f), (f,c), (f,f)\ \}$$

  where the first component of a pair denotes Female's strategy and the second component is Male's strategy. Now, on the one hand-side, Male prefers football over cinema but together is better than alone. So, M's preference can be described by the following utility function:

$$u_M : \begin{array}{rcl} (f,f) & \mapsto & 3 \\ (c,c) & \mapsto & 2 \\ (c,f) & \mapsto & 1 \\ (f,c) & \mapsto & 0 \end{array}$$

  On the other hand-side, Female prefers cinema over football but together is better than alone. So, F's utilities could be as follows:

$$u_F : \begin{array}{rcl} (c,c) & \mapsto & 3 \\ (f,f) & \mapsto & 2 \\ (c,f) & \mapsto & 1 \\ (f,c) & \mapsto & 0 \end{array}$$

  Combined, both utility functions can be modelled as a payoff (bi-)matrix:

$$\begin{array}{cc} & \begin{array}{cc} M & \\ f & c \end{array} \\ \begin{array}{c} F \\ \begin{array}{c} f \\ c \end{array} \end{array} & \left(\begin{array}{cc} (2,3) & (0,0) \\ (1,1) & (3,2) \end{array} \right) \end{array}$$

  Since all information on the game is contained in this representation, we will also identify such a matrix with a 2-person game.

Which strategy profiles are Nash equilibria? We examine all strategy profiles individually:

- (c, c) is a Nash equilibrium, since

$$u_F(c, c) = 3 > 0 = u_F(f, c)$$
$$u_M(c, c) = 2 > 1 = u_M(c, f)$$

- (c, f) is not a Nash equilibrium, since

$$u_F(c, f) = 1 < 2 = u_F(f, f)$$

- (f, c) is not a Nash equilibrium, since

$$u_M(f, c) = 0 < 3 = u_M(f, f)$$

- (f, f) is a Nash equilibrium, since

$$u_F(f, f) = 2 > 1 = u_F(c, f)$$
$$u_M(f, f) = 3 > 0 = u_M(f, c)$$

Now, suppose Female is more decisive: she excludes football an option. Thus, F's modified utility function leads to the following (bimatrix) game

$$\begin{pmatrix} (1,3) & (0,0) \\ (2,1) & (3,2) \end{pmatrix}$$

Then, the only Nash equilibrium is (c, c).

- *Prisoner's dilemma*: Bonnie and Clyde have been captivated and charged with bank robbery. However, the prosecutor is only able to prove illegal possession of firearms to them; without confessions, the sentence will then be 3 years in prison. If one of them makes a confession then the confessor will be sentenced to one year and the non-confessor will be sentenced to 9 years in prison. If both confess then they will be sentenced to 7 years in prison, respectively.

  A game-based formulation of this decision scenario is given by the following game with utilities:

$$\begin{array}{c} \\ s_{11} \\ s_{12} \end{array} \begin{array}{c} \begin{array}{cc} s_{21} & s_{22} \end{array} \\ \begin{pmatrix} (2,3) & (0,0) \\ (1,1) & (3,2) \end{pmatrix} \end{array}$$

  where s_{i1} stands for strategy "confession" and s_{i2} stands for "no confession."

  Which strategy profiles are Nash equilibria?

  - (s_{11}, s_{21}) is a Nash equilibrium, since

$$u_1(s_{11}, s_{21}) = -7 > -9 = u_1(s_{12}, s_{21})$$
$$u_2(s_{11}, s_{21}) = -7 > -9 = u_2(s_{11}, s_{22})$$

– (s_{11}, s_{22}) is not a Nash equilibrium, since

$$u_2(s_{11}, s_{22}) = -9 < -7 = u_2(s_{11}, s_{21})$$

– (s_{12}, s_{21}) is not a Nash equilibrium, since

$$u_1(s_{12}, s_{21}) = -9 < -7 = u_1(s_{11}, s_{21})$$

– (s_{12}, s_{22}) is not a Nash equilibrium, since

$$u_1(s_{12}, s_{22}) = -3 < -1 = u_1(s_{11}, s_{22})$$

Why is this game a dilemma? Because (s_{12}, s_{21}) would be a better strategy profile for both. But it is no equilibrium; each agent could be better off when changing the strategy. The reason for that is the lack of communication and coordination.

- *Rock-paper-scissor*: The scenario consists of two players each of them chooses one of the three gestures "rock", "paper", or "scissor" as a strategy. The rules of winning the game are as follows:
  – rock defeats scissor
  – scissor defeats paper
  – paper defeats rock

  The loser of a game pays a unit to the winner. We can express this a game with utilities by the following bimatrix game:

$$\begin{array}{r} \\ \text{rock} \\ \text{paper} \\ \text{scissor} \end{array} \begin{array}{c} \begin{array}{ccc} \text{rock} & \text{paper} & \text{scissor} \end{array} \\ \begin{pmatrix} (0,0) & (-1,1) & (1,-1) \\ (1,-1) & (0,0) & (-1,1) \\ (-1,1) & (1,-1) & (0,0) \end{pmatrix} \end{array}$$

  Obviously, there is no Nash equilibrium for this game in pure strategies.

An alternative characterization of Nash equilibria can be given by best-response dynamics.

Definition 4.3 *Let $\Gamma = (A, S, u)$ be a game with utilities.*

1. *The* best response *(map) $\beta_i : S_{-i} \to \mathcal{P}(S_i)$* for agent *$i \in A$ is defined by*

$$\beta_i(s_{-i}) =_{\text{def}} \left\{ s_i \in S_i \;\middle|\; u_i(s_i, s_{-i}) = \max_{s_i' \in S_i} u_i(s_i', s_{-i}) \right\}$$

2. *The* best response *$\beta : S \to \bigtimes_{i=1}^{m} \mathcal{P}(S_i)$ is defined by*

$$\beta(s) =_{\text{def}} (\beta_1(s_{-1}), \ldots, \beta_n(s_{-n})).$$

Theorem 4.4 *Let $\Gamma = (A, S, u)$ be a game with utilities. For all $s^* \in S$, it holds*

$$s^* \text{ is a Nash equilibrium} \iff s^* \in \beta(s^*).$$

Proof: Let $s^* \in S$ be a strategy profile. Then, the following chain of equivalences holds:

$$\begin{aligned}
&s^* \text{ is a Nash equilibrium} \\
&\iff u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \text{ for all } i \in A,\ s_i \in S_i && \text{(by Definition 4.2)} \\
&\iff u_i(s_i^*, s_{-i}^*) = \max_{s_i \in S_i} u_i(s_i, s_{-i}^*) \text{ for all } i \in A,\ s_i \in S_i \\
&\iff s_i^* \in \beta_i(s_{-i}^*) \text{ for all } i \in A && \text{(by Definition 4.3.1)} \\
&\iff s^* \in \beta(s^*) && \text{(by Definition 4.3.2)}
\end{aligned}$$

This proves the theorem. ■

Example: Consider the following payoff matrix for a two-person game with identical utility function

$$\begin{array}{c} \\ s_{11} \\ s_{12} \end{array} \begin{array}{c} \begin{array}{cc} s_{21} & s_{22} \end{array} \\ \begin{pmatrix} 1 & 3 \\ 1 & 2 \end{pmatrix} \end{array}$$

The best responses for the agents are

$$\begin{aligned}
\beta_1(s_{21}) &= \{s_{11}, s_{12}\}, & \beta_1(s_{22}) &= \{s_{11}\} \\
\beta_2(s_{11}) &= \{s_{22}\}, & \beta_2(s_{12}) &= \{s_{22}\}
\end{aligned}$$

So, the best response is:

$$\begin{aligned}
\beta(s_{11}, s_{21}) &= (\{s_{11}, s_{12}\}, \{s_{22}\}) \\
\beta(s_{11}, s_{22}) &= (\{s_{11}\}, \{s_{22}\}) \\
\beta(s_{12}, s_{21}) &= (\{s_{11}, s_{12}\}, \{s_{22}\}) \\
\beta(s_{12}, s_{22}) &= (\{s_{11}\}, \{s_{22}\})
\end{aligned}$$

By Theorem 4.4, (s_{11}, s_{22}) is a unique Nash equilibrium.

4.1.2 Potential games

An important class of games with equilibrium guarantee is the class of potential games.

Definition 4.5 *Let $\Gamma = (A, S, u)$ be a game with utilities, and let $P : S \to \mathbb{R}$ be any function.*

1. *P is said to be an* ordinal potential function *for Γ if and only if for all $i \in A$, $s_{-i} \in S_{-i}$, $s_i, \bar{s}_i \in S_i$,*

$$u_i(s_i, s_{-i}) - u_i(\bar{s}_i, s_{-i}) > 0 \iff P(s_i, s_{-i}) - P(\bar{s}_i, s_{-i}) > 0.$$

   Γ is said to be an ordinal potential game *if and only if there is an ordinal potential function for Γ.*

2. *P is said to be a* potential function *for Γ if and only if for all $i \in A$, $s_{-i} \in S_{-i}$, $s_i, \bar{s}_i \in S_i$,*

$$u_i(s_i, s_{-i}) - u_i(\bar{s}_i, s_{-i}) = P(s_i, s_{-i}) - P(\bar{s}_i, s_{-i}).$$

   Γ is said to be a potential game *if and only if there is a potential function for Γ.*

Example: We discuss the notions for two games.

- Consider the following bimatrix game:

$$\Gamma = \begin{pmatrix} (0,3) & (1,2) \\ (3,1) & (2,0) \end{pmatrix}$$

  According to Definition 4.5, it suffices to consider the following differences:

$$\begin{aligned} u_1(s_{11}, s_{21}) - u_1(s_{12}, s_{21}) &= -3 \\ u_1(s_{11}, s_{22}) - u_1(s_{12}, s_{22}) &= -1 \\ u_2(s_{11}, s_{21}) - u_2(s_{11}, s_{21}) &= 1 \\ u_2(s_{12}, s_{21}) - u_2(s_{12}, s_{22}) &= 1 \end{aligned}$$

  Then, Γ is an ordinal potential game. An ordinal potential function P is represented by the matrix

$$\Gamma = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix},$$

  since

$$\begin{aligned} P(s_{11}, s_{21}) - P(s_{12}, s_{21}) &= -1 < 0 \\ P(s_{11}, s_{22}) - P(s_{12}, s_{22}) &= -1 < 0 \\ P(s_{11}, s_{21}) - P(s_{11}, s_{21}) &= 1 > 0 \\ P(s_{12}, s_{21}) - P(s_{12}, s_{22}) &= 1 > 0 \end{aligned}$$

  However, Γ is not a potential game. (An explanation will be given later.)

- Recall that the *prisoner's dilemma* can be represent by the following bimatrix game:

$$\Gamma = \begin{pmatrix} (-7,-7) & (-1,-9) \\ (-9,-1) & (-3,-3) \end{pmatrix}$$

Γ is a potential game, where the potential function P is given by

$$\Gamma = \begin{pmatrix} 4 & 2 \\ 2 & 0 \end{pmatrix}.$$

Proposition 4.6 *Let $\Gamma = (A, S, u)$ be a game with utilities and an ordinal potential function P, and let $s^* \in S$. Then, s^* is a Nash equilibrium if and only if for all $i \in A$ and $s_i \in S_i$, it holds that*

$$P(s^*) \geq P(s_i, s^*_{-i}).$$

Proof: Immediate from Definition 4.5. ■

Corollary 4.7 *Each finite ordinal potential game has a Nash equilibrium.*

Proof: For a finite ordinal potential game Γ, it's ordinal potential funciton P has a maximum. Let $s^* \in S$ be such that $P(s^*)$ is maximum. Then, s^* is a Nash equilibrium by Proposition 4.6. ■

4.1.3 A structural characterization of potential games

How can we decide whether a given game with utilites is, in fact, a potential game? To answe this question, we give a characterization based on the structure of utility functions. It is helpful to introduce some additional notions.

Let $\Gamma = (A, S, u)$ be a game with utilites.

A sequence $p = (s^0, s^1, \ldots, s^N)$ is a *path* in Γ if and only if for all $k \geq 1$, there is an $i \in A$ such that $s^k = (s_i, s^{k-1}_{-i})$ for some $s_i \in S_i$ with $s_i \neq s^{k-1}_i$. The agent $i \in A$ is then called the *deviator* for k. A path $p = (s^0, s^1, \ldots, s^N)$ is said to be *closed* iff $s^0 = s^N$. A path $p = (s^0, s^1, \ldots, s^N)$ is said to be *simple* iff $s^j \neq s^k$ for all $0 \leq j < k \leq N-1$.

Furthermore, for a finite path $p = (s^0, s, \ldots, s^N)$ in Γ, define

$$I(\Gamma, p) =_{\text{def}} \sum_{k=1}^{N} \left(u_{i_k}(s^k) - u_{i_k}(s^{k-1}) \right),$$

where i_k is the deviator for k.

Theorem 4.8 *Let $\Gamma = (A, S, u)$ be a game with utilities. The following statements are equivalent:*

1. *Γ is a potential game.*
2. *$I(\Gamma, p) = 0$ for each finite, closed path p in Γ.*
3. *$I(\Gamma, p) = 0$ for each finite, simple, closed path p in Γ.*
4. *$I(\Gamma, p) = 0$ for each finite, simple, closed path p in Γ of length 4.*

Proof: We show the following implications:

- (1) $\Rightarrow$ (2): Let P be a potential function for $\Gamma = (A, S, u)$. Let $p = (s^0, s^1, \ldots, s^N)$ be a closed path. Then, we conclude

$$\begin{aligned} I(\Gamma, p) &= \sum_{k=1}^{N} \Big(u_{i_k}(s^k) - u_{i_k}(s^{k-1}) \Big) \\ &= \sum_{k=1}^{N} \Big(P(s^k) - P(s^{k-1}) \Big) \qquad \text{(since } P \text{ is a potential function for } \Gamma\text{)} \\ &= P(s^N) - P(s^0) \\ &= 0 \qquad \text{(since } s^N = s^0\text{)} \end{aligned}$$

- (2) $\Rightarrow$ (1): Fix an arbitrary strategy profile $z \in S$. For $s \in S$, let $p(s) = (s^0, \ldots, s^N)$ denote an arbitrary path from $s^0 = z$ to $s^N = s$. We define

$$P(s) =_{\text{def}} I(\Gamma, p(s)).$$

  Note that there is always a path from z to a strategy profile s. We have to show that the following two statements are true:

  1. P is well-defined, i.e., the definition of P is independent of the choice of the path $p(s)$.
  2. P is a potential function for Γ

  This can be seen as follows:

  1. Let $q(s) = (s^0, \ldots t^M)$ be another path such that $t^0 = z$ and $t^M = s$. Then, the concatenated path $\gamma = (s^0, \ldots, s^N, t^{M-1}, \ldots, t^0)$ is a closed path in Γ. By our assumption, it holds that $I(\Gamma, \gamma) = 0$. We conclude

$$\begin{aligned} I(\Gamma, p(s)) &= -I\left(\Gamma, (s^N, t^{M-1}, \ldots, t^0)\right) \\ &= I\left(\Gamma, (t^0, \ldots, t^{M-1}, s^N)\right) \\ &= I(\Gamma, q(s)) \end{aligned}$$

2. For $i \in A$, let $s_i, s_i' \in S_i$ be two strategies, let $s_{-i} \in S_{-i}$. Again by our assumption, we obtain

$$\begin{aligned} 0 &= I\left(\Gamma, \left((s_i, s_{-i}), \ldots, z, \ldots, (s_i', s_{-i}), (s_i, s_{-i})\right)\right) \\ &= I\left(\Gamma, ((s_i, s_{-i}), \ldots, z)\right) + I\left(\Gamma, \left(z, \ldots, (s_i', s_{-i}), (s_i, s_{-i})\right)\right) \\ &= -I\left(\Gamma, (z, \ldots, (s_i, s_{-i}))\right) + I\left(\Gamma, \left(z, \ldots, (s_i', s_{-i})\right)\right) + \\ &\qquad + u_i(s_i, s_{-i}) - u_i(s_i', s_{-i}) \end{aligned}$$

Consequently,

$$\begin{aligned} u_i(s_i, s_{-i}) - u_i(s_i', s_{-i}) &= I(\Gamma, (z, \ldots, (s_i, s_{-i}))) - I(\Gamma, (z, \ldots, (s_i', s_{-i}))) \\ &= P(s_i, s_{-i}) - P(s_i', s_{-i}) \end{aligned}$$

Hence, P is a potential function.

- (2) $\Rightarrow$ (3): Trivial.

- (3) $\Rightarrow$ (4): Trivial.

- (4) $\Rightarrow$ (2): Suppose $I(\Gamma, p) = 0$ for all simple, closed paths p of length 4 in $\Gamma = (A, S, u)$. We show that $I(\Gamma, p) = 0$ for all closed paths p of length N in Γ by induction on N:

  - *base of induction* $N \leq 4$: Cases $N \in \{1, 2, 3\}$ are trivial (in particular, there are no closed paths of odd lengths); for $N = 4$, the statement holds by the assumption.

  - *inductive step* $N > 4$: Let $p = (s^0, s^1, \ldots, s^N)$ be a closed path with $N \geq 5$. Let $(i_1, \ldots, i_N)$ be the sequence of deviators for each step, i.e., $s^j = (s_{i_j}, s^{j-1}_{-i_j})$ such that $s_{i_j} \neq s^{j-1}_{i_j}$. Without loss of generality, assume $i_1 = 1$. Since $s^N = s^0$, there is $2 \leq j \leq N$ such that $i_j = 1$ and $s^j_{i_j} = s^0_1$. Choose j to be minimal subject to this condition, i.e., there is no $2 \leq k < j$ satisfying $i_k = 1$ and $s^k_{i_k} = s^0_1$.

    First, suppose $j = 2$. That is, $s^2 = s^0$. Consider the path $q =_{\text{def}} (s^2, \ldots, s^N)$ of length $N - 1$. Then,

$$\begin{aligned} I(\Gamma, p) &= I(\Gamma, q) + u_1(s^2) - u_1(s^1) + u_1(s^1) - u_1(s^0) \\ &= u_1(s^2) - u_1(s^0) && \text{(by inductive assumption)} \\ &= 0 && \text{(since } s^2 = s^0\text{)} \end{aligned}$$

Now, suppose $j \geq 3$, i.e, $j \in \{3, \ldots, N\}$. Then, we have two subcases:

1. Subcase $i_{j-1} = i_j$. Consider path $q =_{\text{def}} (s^0, \ldots, s^{j-2}, s^j, \ldots, s^N)$. It holds

$$\begin{aligned}
&I(\Gamma, q) \\
&= I\left(\Gamma, (s^0, \ldots, s^{j-2})\right) + u_{i_j}(s^j) - u_{i_j}(s^{j-2}) + I\left(\Gamma, (s^j, \ldots, s^N)\right) \\
&= I\left(\Gamma, (s^0, \ldots, s^{j-2})\right) + u_{i_j}(s^j) - u_{i_j}(s^{j-1}) + \\
&\qquad + u_{i_j}(s^{j-1}) - u_{i_j}(s^{j-2}) + I\left(\Gamma, (s^j, \ldots, s^N)\right) \\
&= I\left(\Gamma, (s^0, \ldots, s^{j-2})\right) + u_{i_j}(s^j) - u_{i_j}(s^{j-1}) + \\
&\qquad + u_{i_{j-1}}(s^{j-1}) - u_{i_{j-1}}(s^{j-2}) + I\left(\Gamma, (s^j, \ldots, s^N)\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\text{since } i_{j-1} = i_j) \\
&= I\left(\Gamma, (s^0, \ldots, s^{j-2}, s^{j-1}, s^j, \ldots, s^N)\right) \\
&= I(\Gamma, p)
\end{aligned}$$

   Thus, by the inductive assumption, $I(\Gamma, p) = I(\Gamma, q) = 0$.

2. Subcase $i_{j-1} \neq i_j$. That is, we have the following scenario: ... Define a path $q_j = \text{def}(s^0, \ldots, s^{j-2}, t^{j-1}, s^j, \ldots, s^N)$ where $t^{j-1} = (s_{i_j}, s^{j-2}_{-i_j})$, i.e., the deviator in $(j-1)$-st step is 1. Now, path $r =_{\text{def}} (s^{j-2}, t^{j-1}, s^j, s^{j-1}, s^{j-2})$ is simple (since $i_j \neq i_{j-1}$), closed, and has length 4. Hence, $I(\Gamma, r) = 0$. That is,

$$I\left(\Gamma, (s^{j-2}, s^{j-1}, s^j)\right) = I\left(\Gamma, (s^{j-2}, t^{j-1}, s^j)\right).$$

   Therefore, $I(\Gamma, p) = I(\Gamma, q_j)$.
   Recursively repeated, we obtain a sequence of paths $q_j, q_{j-1}, \ldots, q_3$ such that $I(\Gamma, p) = I(\Gamma, q_k)$ for all $k \in \{3, \ldots, j\}$ and the deviator in q_k's step $k-1$ is 1. The path q_3 corresponds to the case $j = 2$ above. Thus,

$$I(\Gamma, p) = I(\Gamma, q_3) = 0.$$

This proves the theorem. ∎

4.1.4 A dynamical characterization of potential games

We want to characterize potential games from a dynamical perspective.

Let $\Gamma = (A, S, u)$ be a game with utilities. Let $(s^t)_{t \in I}$ be any finite or infinite sequence of strategy profiles, i.e., $I = \mathbb{N}$ or $I = \{0, 1, \ldots, n\}$ for some $n \in \mathbb{N}$. Then, the sequence $(s^t)_{t \in I}$ is called an *improvement path* if and only if for all $t \in I, t > 0$, there is an $i \in A$ such that $s^t \neq s^{t-1}$, $(s^t)_{-i} = (s^{t-1})_{-i}$, and $u_i(s^t) > u_i(s^{t-1})$. The intuition behind this definition is that each deviator choose a better alternative. Γ is said to have the *Finite Improvement Property* (FIP) if and only if every improvement path is finite.

To establish our characterization, some technical limitations on games are required: A game $\Gamma = (A, S, u)$ is called *degenerate* iff there exist $i \in A$, $s_i, s_i' \in S_i$, $s_i \neq s_i'$, and $s_{-i} \in S_{-i}$ such that $u_i(s_i, s_{-i}) = u_i(s_i', s_{-i})$; otherwise, Γ is called *nondegenerate*.

Theorem 4.9 *Let Γ be a finite, nondegenerate game with utilities. Then, Γ has the FIP if and only Γ is an ordinal potential game.*

Proof:

($\Leftarrow$): Let $\Gamma = (A, S, u)$ be a finite game with ordinal potential function P, i.e., for all $i \in A$, $s_i, s_i' \in S_i$, $s_{-i} \in S_{-i}$,

$$u_i(s_i', s_{-i}) \geq u_i(s_i, s_{-i}) \Longleftrightarrow P(s_i, s_{-i}) \geq P(s_i', s_{-i}).$$

Let $\gamma = (s^0, s^1, s^2, \dots)$ be an improvement path, and let $(i_1, i_2, \dots)$ be the sequence of γ's deviators. Then, for all $t \in I, t > 0$, it holds that $u_{i_t}(s^t) > u_{i_t}(s^{t-1})$. Hence, $P(s^0) < P(s^1) < P(s^2) < \dots$. As S is a finite set, $\gamma = (s^0, s^1, s^2, \dots)$ is a finite sequence, i.e., $\|I\| < \infty$.

($\Rightarrow$): Let $\Gamma = (A, S, u)$ have the FIP. Define a binary relation $>$ on S:

$$s > s' \Longleftrightarrow_{\text{def}} s \neq s' \text{ and there is an improvement path from } s \text{ to } s'$$

Since Γ has the FIP, $>$ is a strict order relation on S, i.e., $>$ is irreflexive and transitive. Any finite strict order can be represented by a function: A set $Z \subseteq S$ is *represented* iff there is a mapping $Q : Z \to \mathbb{R}$ such that for all $s, s' \in Z$, $s > s'$ implies $Q(s) > Q(S')$. Let Z^* be a maximal, represented subset of S.

We show $Z^* = S$. To the contrary, assume there is an $x \in S$, $x \notin Z^*$. Then, there are three (possibly overlapping) cases:

1. There is no $z \in Z*$ such that $z > x$. Define an extension $Q' : Z^* \cup \{x\} \to \mathbb{R}$ by:

$$Q'(z) = \begin{cases} Q(z) & \text{if } z \in Z^* \\ \max\{\ Q(z) \mid z \in Z^*\ \} + 1 & \text{if } z = x \end{cases}$$

   Q' represents $Z^* \cup \{x\}$ and, thus, contradicts to the maximality of Z^*.

2. There is no $z \in Z*$ such that $z < x$. Dually to the first case, define an extension $Q' : Z^* \cup \{x\} \to \mathbb{R}$ by:

$$Q'(z) = \begin{cases} Q(z) & \text{if } z \in Z^* \\ \min\{\ Q(z) \mid z \in Z^*\ \} - 1 & \text{if } z = x \end{cases}$$

   Q' represents $Z^* \cup \{x\}$ and, thus, contradicts to the maximality of Z^*.

3. For some $z, z' \in Z*$, it holds that $z > x > z'$. In this case, define an extension $Q' : Z^* \cup \{x\} \to \mathbb{R}$ by:

$$Q'(z) = \begin{cases} Q(z) & \text{if } z \in Z^* \\ \frac{1}{2}(\max\{\ Q(z) \mid z < x\ \} + \min\{\ Q(z) \mid z > x\ \}) & \text{if } z = x \end{cases}$$

Q' represents $Z^* \cup \{x\}$ and, thus, contradicts to the maximality of Z^*.

Therefore, $Z^* = S$.

Let Q represent S. Then, Q is an ordinal potential function: Suppose $s_i, s_i' \in S_i$, $s_{-i} \in S_{-i}$. Then, $u_i(s_i, s_{-i}) \neq u_i(s_i', s_{-i})$ since Γ is nondegenerate. So, without loss of generality, $u_i(s_i, s_{-i}) > u_i(s_i', s_{-i})$. Thus, $(s_i, s_{-i}) > (s_i', s_{-i})$. (Note there is an improvement path of length one.) Hence, $Q(s_i, s_{-i}) > Q(s_i', s_{-i})$.

This proves the theorem. ■

4.1.5 Congestion games

Congestion games have been introduced in economics by Robert W. Rosenthal in 1973. A scenario related to computer science is as follows. Suppose we are given the following interaction domain $\mathcal{I} \subseteq A \times A$, $A = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}\}$:

...

Here, $\mathtt{A}, \mathtt{B}, \mathtt{C}$, and $\mathtt{D}$ are routers. Router $\mathtt{A}$ wishes to select to route to $\mathtt{C}$, and router $\mathtt{B}$ wishes to select a route to $\mathtt{D}$. If both routers use the same link then the congestion, or latency, increases according to cost function c_i. The routers aim at minimizing their costs.

We can analyze this scenario as a game with utilities

$$\Gamma =_{\text{def}} (\{\mathtt{A}, \mathtt{B}\}, \{\{1,2\},\{3,4\}\} \times \{\{1,3\},\{2,4\}\}, u)$$

where the utility function $u = (u_1, u_2)$ is given by the following bimatrix:

$$\begin{array}{c} \\ \{1,2\} \\ \{3,4\} \end{array} \begin{array}{c} \begin{array}{cc} \{1,3\} & \{2,4\} \end{array} \\ \begin{pmatrix} (c_1(2)+c_2(1), c_1(2)+c_3(1)) & (c_1(1)+c_2(2), c_2(2)+c_4(1)) \\ (c_3(2)+c_4(1), c_1(1)+c_3(2)) & (c_3(1)+c_4(2), c_2(1)+c_4(2)) \end{pmatrix} \end{array}$$

There are two simple, closed paths of length 4 in the game Γ. So, let p be the one in counter-clockwise direction starting with the upper left strategy profile corner, and let q be the one in clockwise direction also starting with the upper left strategy profile corner.

It holds that

$$\begin{aligned} I(\Gamma,p) &= \underbrace{c_3(2)+c_4(1)-c_1(2)-c_2(1)}_{\texttt{A deviates}} + \overbrace{c_2(1)+c_4(2)-c_1(1)-c_3(2)}^{\texttt{B deviates}} + \\ &\quad + \underbrace{c_1(1)+c_2(2)-c_3(1)-c_4(2)}_{\texttt{A deviates}} + \overbrace{c_1(2)+c_3(1)-c_2(2)-c_4(1)}^{\texttt{B deviates}} \\ &= 0 \end{aligned}$$

Since $I(\Gamma,q) = -I(\Gamma,p) = 0$, we obtain from Theorem 4.8 that Γ is a potential game.

Definition 4.10 *A* congestion model *is a tuple* $(A, F, (S_i)_{i\in A}, (w_f)_{f\in F})$ *such that*

1. $A = \{1,\dots,n\}$ *is a non-empty, finite set of* agents *(routers),*
2. F *is a non-empty, finite set of* facilities *(links),*
3. $S_i \subseteq \mathcal{P}(F)$ *is a non-empty set of strategies (routes) for each agent* $i \in A$*, and*
4. $w_f : \{1,\dots,n\} \to \mathbb{R}$ *is a* cost *(wealth, latency)* function *for each facility* $f \in F$*; if* k *agents choose* f *then the cost for each agent is* $w_f(k)$*.*

Definition 4.11 *Let* $(A, F, (S_i)_{i\in A}, (w_f)_{f\in F})$ *be a congestion model. Then,* $\Gamma = (A, (S_i)_{i\in A}, u)$ *is called* congestion game *if and only if for all* $i \in A$*,* $s = (s_i, s_{-i}) \in S$*,*

$$u_i(s) = \sum_{f\in S_i} w_f(\sigma_f(s)),$$

where $\sigma_f(s) = \|\{i \in A | f \in s_i\}\|$*.*

Without proof we state the following theorem which shows that potential games and congestion games are essentially the same class of finite games.

Theorem 4.12 *1. Each congestion game is a potential game.*

2. Each potential game is isomorphic to a congestion game.

The proof of the first statement relies on the Rosenthal *potential*:

$$P(s) =_{\text{def}} \sum_{f\in\bigcup_{i\in A} s_i} \sum_{k=1}^{\sigma_f(s)} w_f(k)$$

4.2 Thresholds

Threshold models are a widely used class of models for behavioral attributes, that is, attributes on items. For instance, they have been used to model diffusion processes (contagion) of innovation, riots, rumors and diseases, strikes, voting (see, e.g, [**?**]), and furthermore in the context of neural networks (e.g., Hopfield networks).

Example: The following scenario was discussed by Granovetter in [**?**] to advocate the usage of thrshold models in social sciences: Imagine there are 100 people milling around in a square–a potential riot situation. Now, assume that there are two slightly different threshold distribution among individuals:

1. There is one individual with threshold 0 (the instigator), one individual with threshold 1, and so on, and one individual with threshold 99. In a "domino" effect, the instigator breaks a window; this activates the person with threshold 1, and so on; finally, all 100 people have joined.
2. There is one individual with threshold 0, no individual with threshold 1, two individuals with threshold 2, one individual with threshold 3, and so on. That is, the crowds are essentially identical. Of course, the riots end with one rioter.

However, newspapers will likely react very differently:

1. "A crowd of radicals engaged in riotous behavior."
2. "A demented troublemaker broke a window while a group of solid citiziens looked on."

It is hazardous to infer individual dispositions from aggregate outcomes.

An n-ary function $f : \{0,1\}^n \to \{0,1\}$ is a *threshold function* if and only if there are weights $w_1, \ldots, w_n \in \mathbb{R}_{\geq 0}$ and a threshold $\vartheta \in \mathbb{R}_{\geq 0}$ such that for all $z_1, \ldots, z_n \in \{0,1\}$,

$$f(z_1, \ldots, z_n) = 1 \iff \sum_{i=1}^{n} w_i \cdot z_i \geq \vartheta_i$$

Example: We discuss several functions.

- The standard case of a threshold function is the n-ary *majority function* which is specified via weights $w_1 = \cdots = w_n = 1$ and threshold $\vartheta = \lceil \frac{n-1}{2} \rceil$.
- Which binary functions are threshold functions?
  - AND is a threshold function: $w_1 = w_2 = 1$, $\vartheta = 2$:

x_1	x_2	$x_1 \wedge x_2$
0	0	0
0	1	0
1	0	0
1	1	1

– OR is a threshold function: $w_1 = w_2 = 1$, $\vartheta = 1$:

x_1	x_2	$x_1 \vee x_2$
0	0	0
0	1	1
1	0	1
1	1	1

– XOR is not a threshold function:

x_1	x_2	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

- The ternary function $f : \{0,1\}^3 \to \{0,1\} : (x_1, x_2, x_3) \mapsto (x_1 \wedge x_2) \vee x_3$ is a threshold function, e.g., via $w_1 = 2, w_2 = 2, w_3 = 5$, and $\vartheta = 4$:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

  Observe that f is the disjunction of two threshold functions.

- In light of the preceding example, is $(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$ then a threshold function as well? No, it is not. Assume it is. Then, there are weights w_1, w_2, w_3, w_4 such that

  $$(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \equiv 1 \iff w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + w_4 \cdot x_4 \geq 1.$$

  So, the following inequalities are true:

$w_1 + w_2 \geq 1$	since $(1,1,0,0)$ is a satisfying assignment
$w_3 + w_4 \geq 1$	since $(0,0,1,1)$ is a satisfying assignment
$w_2 + w_3 \geq 1$	since $(0,1,1,0)$ is not a satisfying assignment
$w_1 + w_4 \geq 1$	since $(1,0,0,1)$ is not a satisfying assignment

  Consequently, $2 \leq w_1 + w_2 + w_3 + w_4 < 2$. A contradiction.

We want to find fixed points in networks over threshold functions. In fact, we prove that fixed points always exist in networks where local transitions belong to a larger class of functions.

A function $f : \{0,1\}^n \to \{0,1\}^k$ is said to be *monotone* if and only if for all $x, y \in \{0,1\}^n$, $x \leq y$ implies $f(x) \leq f(y)$. The less-than-or-equal relation is defined to be the vector-ordering.

Proposition 4.13 *Each threshold function is monotone.*

Proof: Suppose $x = (x_1, \ldots, x_n) \leq (y_1, \ldots, y_n) = y$, i.e., $x_i \leq y_i$ for all $i \in \{1, \ldots, n\}$, and $f(x) = 1$. Since the weights $w_1, \ldots, w_n$ are non-negative, we obtain

$$\vartheta \leq \sum_{i=1}^{n} w_i \cdot x_i \leq \sum_{i=1}^{n} w_i \cdot y_i.$$

Thus, $f(y) = 1$. Therefore, f is a monotone function. This proves the proposition. ■

Note that in the examples above, the last function is monotone, though not a threshold function.

Proposition 4.14 *Let $M = \{f_1, \ldots, f_m\}$ be a set of local transitions on an item set A such that all f_i are threshold functions. Then, $\mathbf{F}_M[A]$ is monotone (with respect to the vector-ordering).*

Proof: Each component of $\mathbf{F}_M[A]$ is a threshold function. That is, each component of $\mathbf{F}_M[A]$ is monotone. Hence, $\mathbf{F}_M[A]$ is monotone with respect to the vector-ordering. This shows the proposition. ■

Theorem 4.15 *Let $M = \{f_1, \ldots, f_n\}$ be a set of local transition functions on an item set A. Let α be any schedule on A. Then, the global network map (M, α) has a fixed point.*

Proof: We give an algorithm for finding a fixed point of (M, α) that, essentially, simulates the global network map along a specific orbit:

```
[1] x := (0, 0, ..., 0)
[2] repeat
[3]     y := x
[4]     x := F_(M,α)(y)
[5] until x = y
```

If the procedure terminates then a fixed point is found. To show the termination property, observe that we construct an ascending chain

$$x^{(0)} \leq x^{(1)} \leq x^{(2)} \leq x^{(3)} \leq \cdots \leq x^{(k)} \leq \ldots,$$

where $x^{(0)} =_{\text{def}} (0, 0, \ldots, 0)$ and $x^{(k)}$ is the configuration assigned to x in the fourth line of the algorithm above. Indeed, this is easily seen by induction:

- *base of induction* $n = 1$: It holds that

  $$x^{(0)} \leq \mathbf{F}_{(M,\alpha)}(x^{(0)}) = x^{(1)}$$

  since $x^{(0)}$ is a bottom element in the poset $(\{0,1\}^n, \leq)$ and since $\mathbf{F}_{(M,\alpha)}$ is monotone by Proposition 4.14.

- *inductive step* $n > 1$: Using the monotonicity of $\mathbf{F}_{(M,\alpha)}$, we obtain

  $$\begin{aligned} x^{(n)} &= \mathbf{F}_{(M,\alpha)}(x^{(n-1)}) && \text{(line (4) of the algorithm)} \\ &\leq \mathbf{F}_{(M,\alpha)}(x^{(n)}) && \text{(by inductive assumption and Proposition 4.14)} \\ &= x^{(n+1)} && \text{(line (4) of the algorithm)} \end{aligned}$$

Since $\{0,1\}^n$ is a finite set, the chain si finite. Thus, the procedure terminates. Therefore, there a fixed point for (M, α). This proves the theorem. ■

Note that, in the proof above, there is a dual procedure starting at $(1, 1, \ldots, 1)$.

4.3 Opinion Dynamics

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.

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 4.1 describes a situation of an interaction pair (i, j).

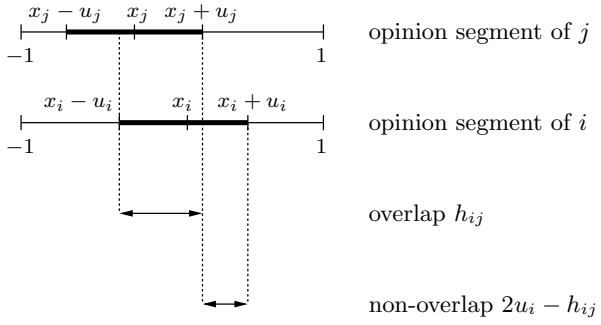


Figure 4.1: The Relative Agreement model

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, $0 < \mu < 1$.

Proposition 4.16 *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 4.17 *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 4.16, 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. ∎

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 [**?**]:

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 4.2, 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.

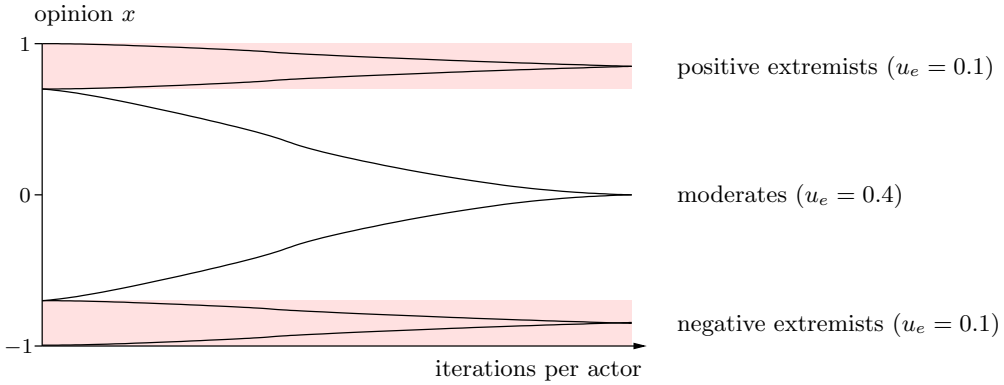


Figure 4.2: Scheme of central convergence.

In Figure 4.3, 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.

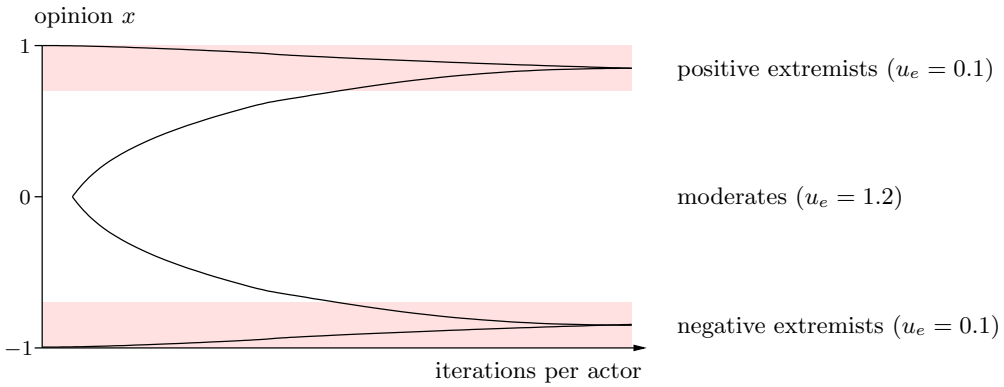


Figure 4.3: Scheme of bipolarization.

In Figure 4.4, 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.

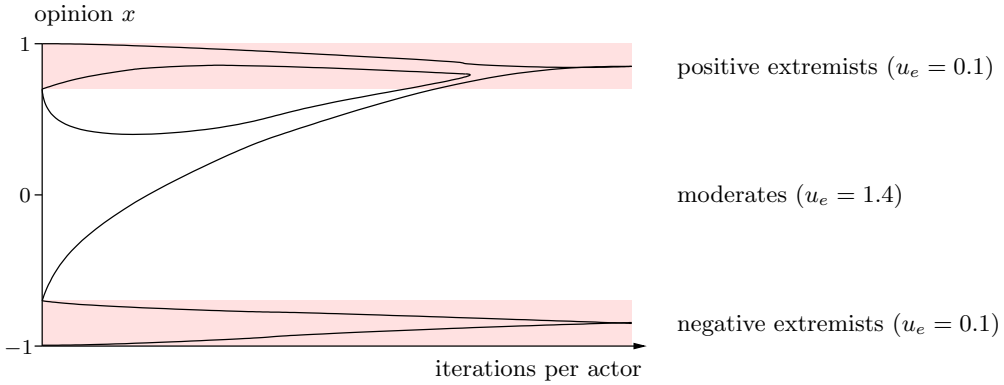


Figure 4.4: Scheme of single polarization.

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., [14, 6]) 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., [20]).

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 [39, 38] 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., [20, 19]). 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 [38]. 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.

The Border Gateway Protocol B

This chapter is taken from [23].

B.1 Background

The Internet is the ultimate communication network at present. Recent estimations from March 2007 [28] show that a total of 1,114,274,426 Internet users worldwide are connected which corresponds to an Internet penetration rate of 16,9 % of the world population. The basic functionality of the Internet is sending data from source entities to destination entities over the data links of the network. The technical characteristics of the data links impose at least two kinds of restrictions on the network:

- only a limited number of entities can share the same data link and
- only a limited amount of data can be sent over the data link in a certain time.

In view of the large variety of Internet infrastructure these restrictions are significant. Consequently, to maintain the functionality the Internet operation mode has been based on the next-hop principle and data fragmentation.

The next-hop principle is involved in the Internet routing problem. The routing problem in the Internet can be characterized by the following terms:

- A *path* is a sequence of entities which pairwise share a direct data link.
- Sending data from an entity to an entity—the next hop—over a direct data link is called *forwarding*.
- A *transmission* is the process of successively forwarding data from a source to a destination over a path with the source as its first and the destination as its last element.
- The selection of a path for a transmission is called *routing*.

To act as intermediaries the entities need routing information. A *datagram* is a self-contained packet consisting of data and routing information. In communication networks of interest all data are transmitted as datagrams. The next-hop principle provides a successful solution to the first restriction above.

A solution to the second restriction is fragmenting and reassembling datagrams when necessary for a transmission through small-capacity networks.

Functionalities of communication networks are specified by protocols, i.e., sets of allowed messages (datagrams) and descriptions of the orders in which these messages have to be communicated. As the Internet aggregates an increasing number of functionalities, many different protocols interact in a non-trivial way. The current Internet is characterized by the TCP/IP protocol suite, a family of about 500 protocols named after the two most important protocols in it: the Transmission Control Protocol (TCP) [33] and the Internet Protocol (IP) [32]. The protocols in the TCP/IP protocol suite can be divided into five layers[1]: the physical layer, the datalink layer, the network layer, the transport layer, and the application layer. These layers are hierarchically ordered with the physical layer at the bottom and the application layer at the top. Typically, the functionality of a protocol in a layer above the physical one relies on the functionality of the protocols in the subjacent layers.

The protocols in the network layer are responsible for routing. They handle three major tasks: addressing physical objects, providing fragmentation support of datagram forwarding from a source address towards the destination address, and disseminating and selecting routes from source addresses to destination addresses. The first two tasks are managed by the IP protocol. The third task is under control of routing protocols. The Border Gateway Protocol (BGP or BGP-4) [34] is the *de facto* standard routing protocol for interdomain routing, i.e., it rules how different networks learn which routes they can use to communicate.

This chapter gives a systematic survey on BGP and its functionality. The exposition is based on [40, 24].

B.2 Terminology

In this section we gather the relevant terminology of the IP protocol necessary to see how BGP is involved in Internet routing. In the forthcoming we identify IP with IP version 4 (IPv4). Though newer versions of IP, known as IPng or IPv6 [7], are designed and implemented to overcome some shortcomings of IPv4, mainly the limited number of addresses, IPv4 is still the most used version of IP. Also, the fundamental principles do not differ.

Basically, the Internet Protocol (IP) implements two functions [32]: addressing and fragmentation. Fragmentation is less relevant for routing. So we give a short introduction to addressing and the related forwarding method.

B.2.1 Physical networks

The elementary physical entities in the Internet are computing devices running an implementation of IP and the physical datalink connections between them. We should note

[1]Note that the TCP/IP layer structure is different to those of the OSI model (Open System Interconnection Basic Reference Model) which provides an abstract seven-layer model for networking (see, e.g., [36]).

that data links typically connect many devices. In this respect they should be graph-theoretically modeled by hyperedges rather than by edges. Technically, connecting many devices via a data link is realized by gadgets such as switches, bridges, and hubs. As long as these gadgets do not follow the IP protocol, they are not considered a visible part of the Internet.

A *physical subnet* is a set of devices connected by a single data link. *Physical networks* are inductively defined as follows:

- Each physical subnet is a physical network.
- The (finite) union of physical networks is a physical network.
- Nothing else is a physical network.

Thus, a physical network can be decomposed in a finite number of physical subnets. If two physical subnets have at least one device in common, they are *connected*. A physical network is connected if and only if for each pair (N, N') of physical subnets there are physical subnets $I_0, I_1, \ldots, I_r$ such that $N = I_0$, $N' = I_r$, and for all $j \in \{1, \ldots, r\}$, I_{j-1} and I_j are connected. The Internet in a physical sense is the physical network maximal subject to set inclusion. Note that the physical size of the Internet is time-depending.

The devices of a physical network are divided into hosts and routers: a *host* is a computing device having only one datalink connection to the physical network. A *router* has at least two datalink connections to the physical network. The boundary of a computing device and a data link is called an *interface*.

B.2.2 Logical networks

The elementary logical entities in the Internet are binary words. Addressing refers to mapping physical entities to logical entities. However, addressing is more than that: an address should indicate where to find the physical entity.

An *IP address* (or simply, *address*) is a word over $\{0, 1\}$ of length 32 (or of length 128 in IPv6). Each interface of a physical network has a unique address.

For a word $p \in \{0, 1\}^*$ such that $|p| < 32$, the set $p \cdot \{0, 1\}^{32-|p|}$ is called a *logical network* (or simply, *network*). The word p is called an *IP prefix* (or simply, *prefix*). Typically, physical networks are embedded into logical networks, i.e., the set of addresses of the interfaces of a physical network is a subset of a logical network. The other way round, a logical network with prefix p can contain an embedded physical network having at most $2^{32-|p|}$ interfaces. Networks are partially ordered according to the dual prefix relation, i.e., if a word p is a prefix of a word p', then the network with prefix p is a *supernetwork* of the network with prefix p' and the network with prefix p' is a *subnetwork* of the network with prefix p.

It is common to denote IP addresses in dot-decimal notation, i.e., the address is separated by dots into four blocks of eight bits and each block is written in decimal.

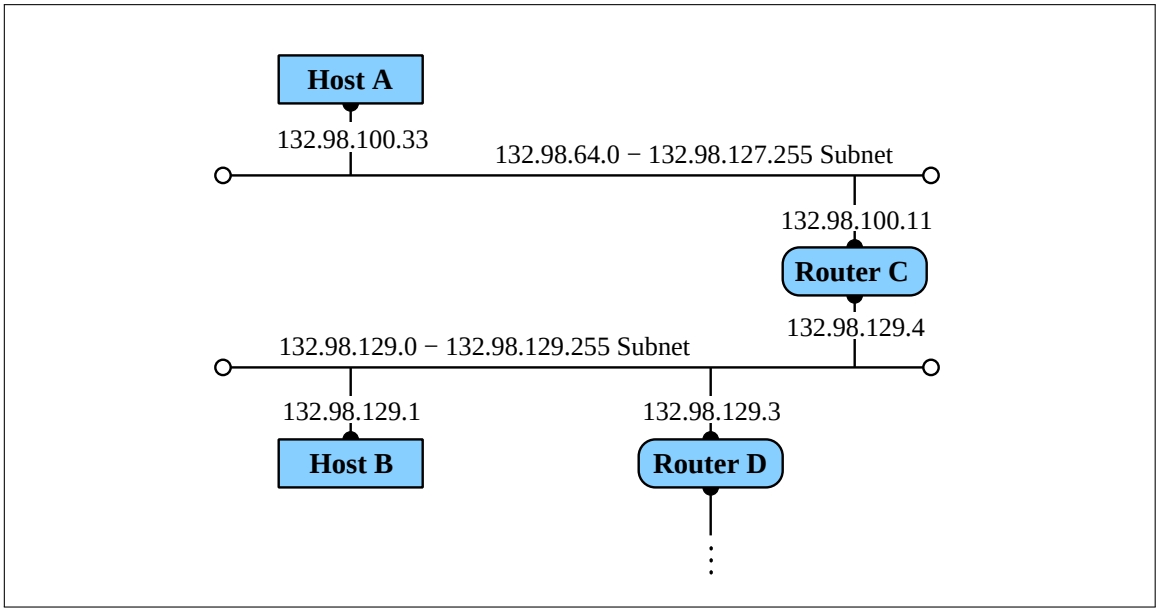


Figure B.1: A small network example

Example: In dot-decimal notation the IP address 10000100011001000110011000100001 corresponds to 132.98.100.33.

IP addresses are considered to consist of two parts: a network part and a host part. A network part of size b consists of the first b bits of the IP address and is used to identify the network. It is clear that the network part is just the prefix of the network. The remaining $32 - b$ bits of the IP address are used to identify a certain host or router within the network. The first address in the network, i.e., the address with an all-zero host part, is the network address and cannot be used for addressing a single interface. The last address in the network, i.e., the address with an all-one host part, is the network broadcast address and cannot be used either. Network addresses are denoted by usual addresses.

Following the Classless Inter-Domain Routing (CIDR) address architecture [35, 11], there are two ways to specify uniquely the network part of an address:

- *enhanced prefix format*: Simply attach the length of the network part to the address.
- *netmask format*: A netmask consists of a block of 1's, corresponding to the network part, followed by a block of 0's, corresponding to the host part. Netmasks are also denoted in dot-decimal notation.

In the following we identify prefixes and network addresses. The address 0.0.0.0/0 is the network address of the Internet.

Example: Suppose the address 132.98.100.33 is assigned to an interface in a network with an 18-bit identifier. Then, the network address is denoted by 132.98.64.0/18 in

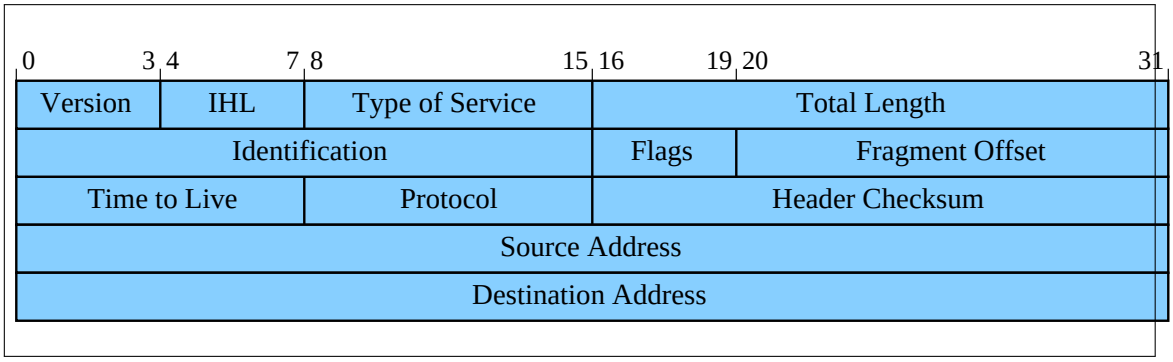


Figure B.2: The format of the IP header

the enhanced prefix format or is uniquely described by the netmask 255.255.192.0 for IP address 132.98.100.33 (taking a bit-wise AND). A small network exemplifying the situation is shown in Figure B.1. The upper subnetwork has the network address 132.98.64.0/18 and the broadcast address 132.98.127.255. The lower subnetwork has the network address 132.98.129.0/24 and the broadcast address 132.98.129.255. All interfaces have addresses in these ranges.

B.2.3 Datagrams and forwarding

A datagram is a self-contained piece of code consisting of user data and routing address information. An *IP datagram* is a datagram starting with a header of 160 bits divided into 12 fields followed by the datagram body containing the user data. The format definition of the header is given in the Figure B.2. The most interesting fields from the viewpoint of routing are the following:

- Destination Address containing the essential routing information
- Total Length containing the length of the entire IP datagram, including the header, in bytes

It follows that an IP datagram has a maximal size of 65536 bytes (a size not exceeded in IPv6 as well). A detailed description of all fields can be found in [32]. As no datagram type other than IP datagrams occurs in our setting we usually speak of datagrams when meanining IP datagrams.

Forwarding refers to transmitting datagrams from device to device over a common datalink connection in order to bring the data nearer to the destination. Forwarding according to IP works as follows: on the reception of a datagram (which has not reached its destination address), a computing device, typically a router, proceeds with three steps:

1. Take the address written in the Destination Address field of the datagram.

Destination	Host A	Host B	Router C	Router D
132.98.64.0/18	directly connected	132.98.129.4	directly connected	132.98.129.4
132.98.129.0/24	132.98.100.11	directly connected	directly connected	directly connected
0.0.0.0/0	132.98.100.11	132.98.129.3	132.98.129.3	...

Table B.1: Forwarding Information Bases for hosts and routers in Figure B.1.

2. Find an associated interface—the *next hop*—for the address in the Forwarding Information Basis.

3. Transmit the datagram over a datalink connection to this interface.

As an additional step the datagram can be fragmented into smaller pieces (which too have the IP datagram format) if the original datagram is too large to traverse a data link. In IPv4 each router on a transmission path is allowed to do this. In IPv6 this is reserved for hosts.

The Forwarding Information Basis (FIB) of a computing device simply consists of a list of network addresses and associated IP addresses (interfaces) in the logical (physical) network. Given an FIB and given a destination address the correct interface is found by solving the *longest prefix matching* problem, i.e., find the network address in the FIB which is the longest IP prefix of the destination address. Several algorithms and data structures have been designed to solve this problem efficiently (see, e.g., [41, 21]).

Example: Table B.1 shows the content of the Forwarding Information Bases of all devices of our example network in Figure B.1. Suppose we want to send data from host B to the address 132.98.100.33. That is, the longest prefix that matches is 132.98.64.0/18 and the associated interface has address 132.98.129.4 which belongs to Router C. Note that the prefix 0.0.0.0/0 always matches. This is the default prefix which has to be contained in each forwarding table. For instance, if we want to send data from host B to the address 212.126.34.12, then the default prefix is the only prefix that matches the address. In this case the address 212.126.34.12 lies outside the network under consideration and we do not know where to find a specific network containing the address. We thus send packets to somewhere in the Internet where routers have more information on the address.

B.3 Autonomous Systems

The introduction of Autonomous Systems reduces the complexity of routing, simply by reducing the number of possible paths through the Internet. In this section we define Autonomous Systems and discuss related concepts.

B.3.1 Definition

According to [17], an *Autonomous System* (AS) is a connected group of one or more IP prefixes run by one or more network operators which has a single and clearly defined routing policy. Each AS has a unique natural number for identification. The current version of BGP limits the size of these numbers to 16 bits. More precisely, the set of possible AS numbers is $\{1, 2, \ldots, 65535\}$. In practice, however, the AS numbers in the range 64512–65535 are reserved for private use without global visibility [17].

Recall that an IP prefix is a range of IP addresses a physical network is embedded into. In a connected group of IP prefixes the underlying physical networks are connected. An interface belongs to an AS if its IP address belongs to the AS. A computing device belongs to an AS if one of its interfaces belongs to the AS. A router belonging to some AS is called *visible* if it shares a data link with a computing device belonging to another AS. The other routers are called invisible or *internal*. A visible router is also called a *gateway router*.

The *connectivity graph* (at the AS level) or simply *AS graph* is an undirected graph having the set of assigned AS numbers as its vertex set. The edge set is defined as follows: there is an edge between ASes u and v if and only if a router belonging to AS u and a router belonging to AS v share a common datalink connection. The AS graph is an abstract view on the underlying physical Internet level. Actually, it can be considered as a minor of the graph based on the router level.

Example: Figure B.3 shows a small Internet example at the AS level. There is an edge drawn between two ASes if there is a data link connecting routers from both ASes. For instance, AS 2 is connected to AS 1, AS 3, AS 4, and AS 5. As indicated by the detailed view into AS 2, there are two routers (A and B) connected with AS 1, router D shares one data link with AS 4 and another with AS 5, and router E is connected with AS 3. The routers A, B, D, and E are visible. The router E is an internal router. Note that AS 2 embeds a connected physical network.

B.3.2 Interrelationships

An AS is associated with an organization or an administrative domain. Within this organization a network operator is an institution responsible for managing the network at both the physical and the logical level (see, e.g., [40, Chapter 8] for a list of responsibilities). A typical organization owning an AS is an Internet Service Provider (ISP). An ISP sells transmission paths (with differentiated quality of service) to its customers. Thus, a primary operating goal of the network operator within an ISP is maintaining global reachability for its customers, i.e., ideally providing a path in both directions to every reachable prefix.

An ISP can be a customer of another ISP. A customer ISP uses its provider ISP for *transit*, i.e., for transmissions along paths through the network of the provider ISP to destination prefixes possibly outside any AS owned by the provider ISP. An AS inherits the business

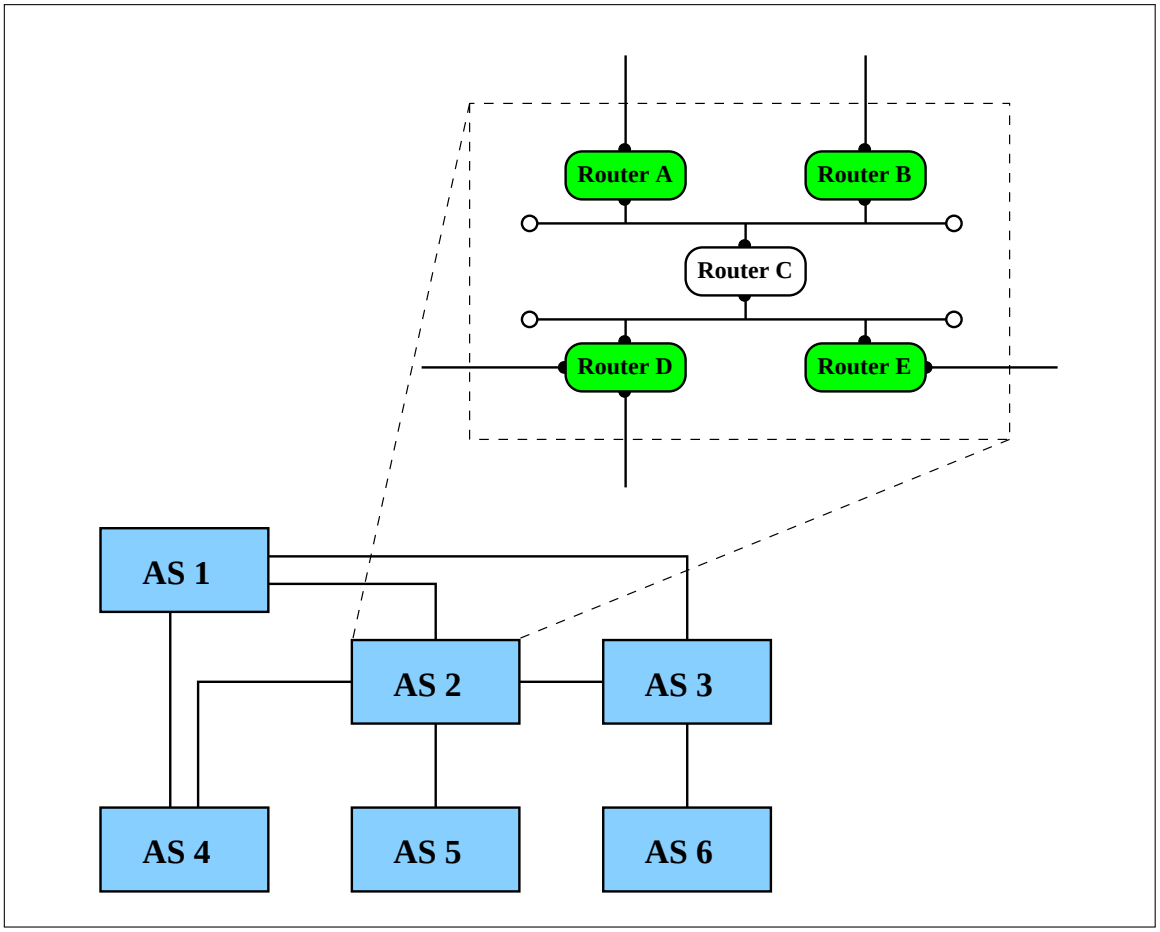


Figure B.3: A small Internet example at the AS level

relationships of its owning ISP. There are three fundamental types of interrelationships between an AS i and an AS j:

- *customer-to-provider*: the AS i is a customer of the AS j if the ISP owning AS i buys transmission paths from the ISP owning AS j,
- *peer-to-peer*: the ASes i and j provide special paths to their customers without the owning ISPs having a customer-to-provider relationship in either direction,
- *sibling-to-sibling*: the ASes i and j belong to the same ISP.

More peculiar types of relationships appear in the real world (see, e.g., [12]). We restrict ourselves to the three mentioned types.

Example: Figure B.4 shows a different view on our Internet example. First, we are only interested in the relation between the different ASes. An arrow from an AS to another AS indicate a customer-to-provider relationship. For instance, AS 4 is a customer of both

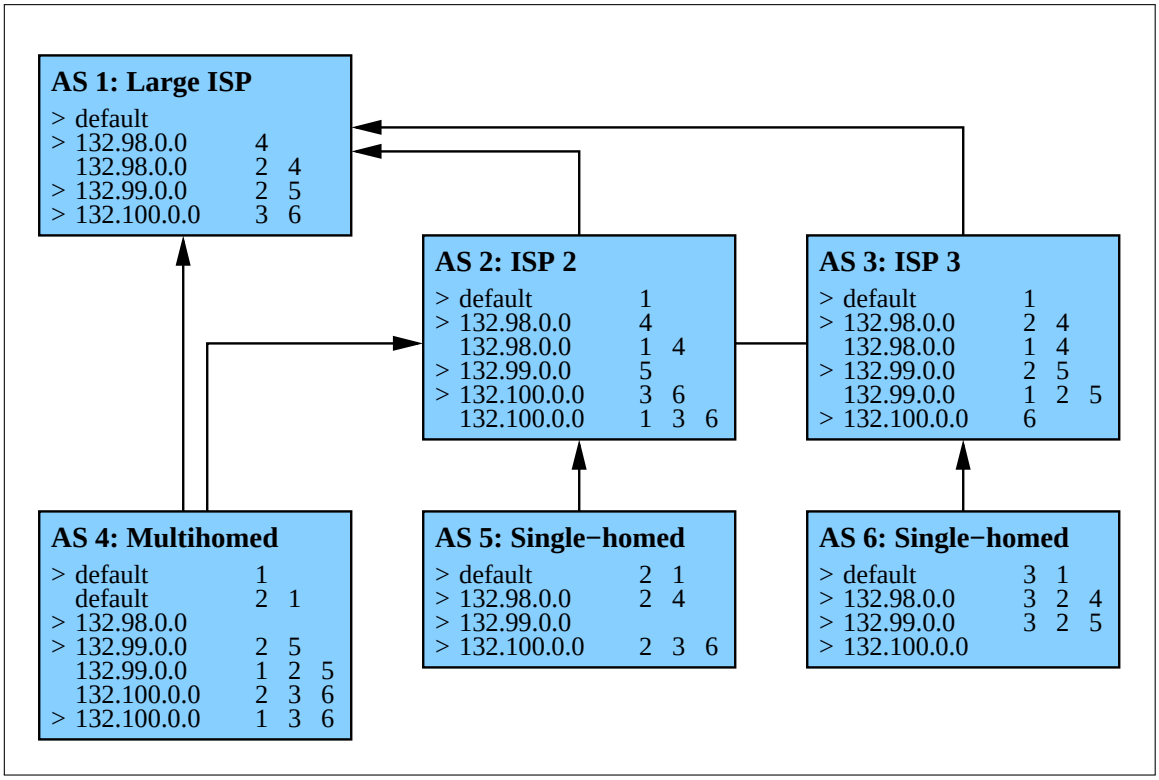


Figure B.4: A small Internet example with local Routing Information Bases

AS 1 and AS 2. AS 1 has no provider. The large ISP behind AS 1 is a so-called *tier-1* provider. All other ASes have to pay at least one AS for transit. The ISPs 2 and 3 have a peer-to-peer relationship.

B.3.3 Routing policies

A *routing policy* is a fixed and formalized set of rules outlining the distribution of routing information between an AS and the neighbored ASes. As routing policies are implemented in gateway routers, a routing policy can be viewed as the union of all implemented sets of rules of gateway routers of an AS. In case of a policy change the gateway routers need to synchronize their set of rules to implement a single routing policy of the AS. Reasonably, the routing policies of an AS are in accordance with existing customer-to-provider and peer-to-peer relationships.

Example: Inside the boxes of each AS in Figure B.4 it is shown which path is considered best by the AS to reach a certain IP prefix. As an example let us look at AS 4. The IP prefix 132.98.0.0/16 is located in AS4 itself and is thus reachable over a local path. In contrast, to reach the prefix 132.100.0.0/16 which is owned by AS 6 the chosen path goes through AS 1. Thus, the routers within AS 4 forward datagrams with destination prefix 132.100.0.0/16 towards some gateway routers sharing a data link with a gateway

router of AS 1. There the datagrams leave AS 4 and enter AS 1. An analysis of the local Routing Information Base of AS 4 leads to the following (incomplete) rule for all prefixes: a shorter path is preferred over a longer path and paths through AS 1 are preferred among equal-length paths. As another example, AS 3 prefers the path through AS 2 over the path through AS 1 to reach the IP prefix 132.98.0.0/16 in AS 4. This is reasonable as AS 3 and AS 2 have a peer-to-peer relationship whereas AS 3 is a customer of AS 1. Transmissions through AS 2 are thus typically cheaper than those through AS 1.

B.3.4 Routing hierarchy

Example B.3.3 demonstrates that two mechanisms are needed to select an address-to-address path:

- one for deciding which sequence of ASes leads to the destination address
- one for deciding which sequence of routers traverses an AS

The mechanisms are hierarchically ordered: a local path can only be selected after the decision has been made which pair of gateway routers is to be connected for an AS traversal. Any routing protocol used within an AS for selecting local paths is called an *intra-AS routing protocol* or *interior gateway protocol* (IGP). There are two classical types of such protocols (see, e.g., [24]):

- A *link-state protocol* implements a link-state algorithm. A link-state algorithm consists of two phases: in the first phase, the cost of each connection in the network is broadcast until all devices know the complete cost matrix. In the second phase, at each device an appropriate single-source shortest path algorithm is executed on the same cost matrix. Due to their communication overhead link-state algorithms do not scale well with an increasing number of devices. Examples of link-state protocols are the Open Shortest Path First (OSPF) protocol [29, 30] or the Intermediate-System-to-Intermediate-System (IS-IS) protocol [31, 4].
- A *distance-vector protocol* implements a distance-vector algorithm. A distance-vector algorithm works in an iterative, asynchronous, and distributed fashion and converges always. In essence, each device communicates to its neighbors its currently known distances to all destination devices in the network. On reception of such a distance vector a source device computes a new distance to each destination in the network as the minimum of the cost of a connection to some neighbor plus the neighbor's distance to the destination (distance-relaxation rule). If a distance to some destination has changed, then the new distance vector is communicated. Distance-vector protocols tend to congest routers having a high betweenness centrality, i.e., those routers that lie on many shortest paths. They do not respect administrative autonomy which makes them only suitable for usage within a single AS. Examples of such protocols are the Routing Information Protocol (RIP) [18, 25] or the Enhanced Interior Gateway Routing Protocol (EIGRP) [3].

An *inter-AS routing protocol* or *exterior gateway protocol* (EGP) is a protocol responsible for routing at the AS level. BGP is the *de facto* standard inter-AS routing protocol. Note that BGP can be also applied as an intra-AS routing protocol. Sometimes BGP used within an AS is denoted by iBGP and BGP used between ASes is denoted by eBGP.

B.4 Protocol outline

We turn to the specification of BGP as proposed in [34]. What is currently understood as BGP refers to BGP-4. The precursor versions of the protocol are completely obsolete. In contrast to the link-state and distance-vector protocols mentioned in Subsection B.3.4, BGP can be characterized as a *path-vector protocol*.

A computing device capable of performing BGP communications is called a *BGP speaker*. The primary function of a BGP speaker is exchanging routing information with neighbored BGP speakers. The information consists of network reachability information and a list of ASes that the information has traversed. BGP uses the TCP protocol for establishing reliable connections.

In the following we describe only those parts of the BGP protocol that are relevant for the routing problem, thus omitting details concerning the communication process itself. In particular, we omit all error handling features. Moreover, we assume that all necessary BGP communications can be safely processed.

B.4.1 Operating mode

The general operating mode of BGP is the following: Two BGP speakers form a TCP connection between one another. They exchange messages to open and confirm the connection parameters. A connection between BGP speakers of different ASes is called an *external link*. A connection between BGP speakers within the same AS is called *internal link*. Initially, the complete routing information is exchanged between the BGP speakers. Subsequently, incremental updates are sent whenever routing information of one of the BGP speaker has changed.

A *route* is defined to be a tuple consisting of an IP prefix and attributes of a path to the AS containing the IP prefix. Routes are advertised between a pair of BGP speakers in update messages. The IP prefix is contained in the Network Layer Reachability Information (NLRI) field, and the path together with a set of attributes is reported in the same update message. Routes are stored in Routing Information Bases (RIBs). Conceptually, there are three types of RIBs:

- An *Adj-RIB-In* contains information on routes that have been learned from inbound update messages.
- A *Loc-RIBs* contains information on routes that have been selected from the Adj-RIBs-In by applying local policies.

Marker	Length	Type	Message contents
16 bytes	2 bytes	1 byte	0-4077 bytes

Table B.2: BGP message format.

UR Length	Withdrawn Routes	Total PA Length	Path Attributes	NLRI
2 bytes	Variable	2 bytes	Variable	Variable

Table B.3: BGP update message format.

- An *Adj-RIBs-Out* contains information on routes that have been selected for advertisement to BGP speakers in an outbound update message.

A BGP speaker has an Adj-RIB-In and an Adj-RIB-Out for each possible connection but only one Loc-RIB. Note that RIBs need not be implemented separately.

BGP also provides mechanisms to inform another BGP speaker that a previously advertised route is no longer available, e.g., the IP prefix can be included in the Withdrawn Routes field of an update message or a replacement route to the same IP prefix can be advertised.

B.4.2 Message formats

All BGP information is exchanged in form of messages. A BGP message has up to 4096 bytes consisting of a fixed-size header of 19 bytes.

Table B.2 shows the structure of a BGP message. The Marker field usually contains all one's and is intended to check whether communication works correctly. Any zero occuring in the Marker field induces an error handling procedure. The Length field contains the length of BGP message in bytes. The Type field contains a code for the type of the message. Originally, there are four types of messages: open, update, keepalive, and notification messages. For our purposes, only update messages are relevant.

In Table B.3 the structure of an update message is shown. The fields of an update message are used as follows:

- The Unfeasible Routes (UR) Length field contains the length of the Withdrawn Routes field in bytes; zero means that the field is not present in this message.
- The Withdrawn Routes field is optional and contains a variable-length list of IP prefixes encoded by the enhanced prefix format.

Attribute Flags	Attribute Code	Type	Attribute Length	Attribute Value
1 byte	1 byte		1-2 bytes	Variable

Table B.4: Path attribute format.

- The Total Path Attribute (PA) Length field contains the length of the Path Attribute field in bytes; zero means that the Path Attribute field and the NLRI field are not present in this message.
- The Path Attribute field is optional and contains a variable-length list of path attributes which are explained in more detail in the next subsection.
- The Network Layer Reachability Information (NLRI) field is optional and contains a variable-length list of IP prefixes encoded by the enhanced prefix format.

Note that an update message can advertise only one route but can simulteanously withdraw many routes. An advertised route thus contains information on how to reach each prefix present in the NLRI field of an update message.

B.4.3 Path attributes

Path attributes are contained in the Path Attribute field of an update message.

The structure of a path attribute is shown in Table B.4. The meaning of the fields is as follows:

- The Attribute Flags field contains information on how to process the attribute. For instance, if Bit 0 is set to zero and Bit 1 is set to one, then the attribute is well-known mandatory, i.e., it must be included in every update message. If Bit 3 is set to zero, then the Attribute Length field is one byte, otherwise it is two bytes. Though flags are important in practice, they are not relevant in our setting.
- The Attribute Type Code field contains information on the meaning of the attribute.
- The Attribute Length field contains the length of the Attribute Value field in bytes.
- The Attribute Value field contains data according to the Attribute Type Code field.

Important available types of attributes as described in the Attribute Type Code field are the following:

- ORIGIN is a well-known mandatory attribute that defines the origin of the path information.

- AS_PATH is a well-known mandatory attribute that is composed of a sequence of AS path segments. A path segment can be one of the two types AS_SET, i.e., an unordered set of ASes a route in the update message has traversed, and AS_SEQUENCE, i.e., an ordered set of ASes a route in the update message has traversed. The AS_SET type of path segment is involved in route aggregation.

- NEXT_HOP is a well-known mandatory attribute that defines the IP address of a gateway router that should be used for datagram forwarding towards the destination listed in the NLRI field of the update message.

- MULT_EXIT_DISC is an attribute that can be used to discriminate among multiple external links of an AS.

- LOCAL_PREF is an attribute that is used to inform the BGP speakers with the same AS of the degree of preference of the originating BGP speaker for an advertised route.

- COMMUNITIES is an attribute for aiding policy management. A community is a set of destination prefixes sharing a common property [37].

The attribute type most relevant for route dissemination is the AS_PATH type. At an internal link the AS_PATH attribute is processed as follows:

- When a BGP speaker originates a route then the originating speaker includes an empty AS_PATH attribute in all update message sent to BGP speakers in its own AS.

- When a BGP speaker propagates a route learned from another speaker then the AS_PATH attribute associated with the route is not modified.

At an external link the AS_PATH attribute is processed as follows:

- When a BGP speaker originates a route then the originating speaker includes its own AS number as the only entry in the AS_PATH attribute in all update messages sent over the external link.

- When a BGP speaker propagates a route learned from another speaker then the AS_PATH attribute associated with the route is modified depending on the type of the path segment:

  - If the first path segment of the AS_PATH attribute is of type AS_SEQUENCE, the BGP speaker prepends its own AS number to the sequence.
  - If the first path segment of the AS_PATH attribute is of type AS_SET, the BGP speaker prepends a new path segment of type AS_SEQUENCE to the AS_PATH attribute, including its own AS number in that segment.

B.4.4 Route propagation

The update message format is designed to support the process of propagating routes among BGP speakers. When a BGP speaker receives an update message from a neighboring BGP speaker containing withdrawn routes, it removes all routes from the corresponding Adj-RIB-In which have an NLRI listed in the Withdrawn Routes field of the message. When a BGP speaker receives a new route in an update message from a neighboring BGP speaker, it executes the following procedure:

1. Check the *inbound filters* defined for the connection. If the route does not pass all filters, then the procedure stops.
2. Insert the new route in the corresponding Adj-RIB-In.
3. Execute the *route-selection algorithm* on all routes present in any Adj-RIB-In which have the same NLRI as the new route. If the new route is not selected as the best route, then the procedure stops.
4. Include the new best route in the Loc-RIB and include the next-hop address for NLRI in the Forwarding Information Base. The old best route to NLRI is removed from Loc-RIB and the old next-hop address is removed from the Forwarding Information Base.
5. Revoke the old best route in update messages to those BGP speakers that had received the old best route.
6. Check the *outbound filters* for each link. If the new best route passes the filter for the link, then include the route in the corresponding Adj-RIB-Out. Note that for an internal link there is typically no link.
7. Send update messages advertising the new best route to all neighboring BGP speakers for which the Adj-RIB-Out has changed. Recall how the AS_PATH attributes are processed for each link.

The procedure follows an idealtypical scheme. In practice there is space for implementation diversity. This does not only concern filters and the route-selection algorithm. For instance, route flap damping leads to a different route-revocation procedure in the fifth step (see, e.g., [40, chapter 10]). The technique is a best practice which is not covered by the BGP protocol [42, 26]. Examples which are covered by the protocol are route aggregation and handling overlapping routes [34]. Both techniques lead to minor changes in the propagation process.

B.4.5 Route-selection algorithms and filters

Local routing policies influence the route propagation process

- by inbound filters,

- by the route-selection algorithm, and
- by outbound filters.

The route-selection algorithm is the central tool for implementing a routing policy. It provides a mechanism for selecting the best route from the set of available routes from different neighbors. A standard implementation of the route-selection algorithm works as follows: compute for each route under consideration an integer local preference according to the local Policy Information Base and choose the route with highest local preference. If the highest local preference is taken by more than one route, then iteratively apply a set of tie-breaking rules until a single route is selected. The following set of criteria is recommended for application in exactly this order [34]:

1. highest local preference
2. lowest number of AS path segments
3. lowest MULT_EXIT_DISC entry (if all routes have such an entry)
4. lowest cost of a path through the same AS to the next-hop address
5. lowest IP address of a BGP speaker advertising over an external link
6. lowest IP address of a BGP speaker advertising over an internal link

In contrast to the prescribed BGP message format which allows no deviation practical route-selection algorithms have additional steps throughout the selection process. The scheme above ignores such vendor-dependent extra steps. Furthermore, there is no requirement that all BGP speakers within the same AS agree on how to select the best route.

Filters are algorithms for discarding undesirable routes. Routes not discarded can be transformed in the process. Inbound filters implement the import policies of an AS. They are primarily intended to exclude routing loops, i.e., if a BGP speakers detect its own AS number in the received AS path, then the route is discarded. Furthermore, inbound filters are used to influence the route-selection algorithm, e.g., by assigning weights to routes or by prepending the own AS number one or more times to the beginning of the AS paths in order to make the path longer. Outbound filters implement the export policies of an AS.

The rules for manipulating routes and for setting local preferences are specified by *filter lists* or *route maps* in Policy Information Bases. In general, there is no specific formalization for expressing these rules. Typically, they are described by if-then-constructions: if the head of the rule matches a route, then a specified action is executed on this route.

Example: A unifying approach to define import and export policies is the Routing Policy Specification Language (RPSL)[2, 27] intended to collect routing policies in a Routing Registry. For instance, the routing policy of AS4 in Figure B.4 could look like the following in RPSL format:

```
aut-num:    AS4
```

```
import:     from AS1
            action pref=2; med=10
            accept ANY
import:     from AS2
            action pref=2; med=20
            accept ANY
export:     to AS1
            announce NOT AS2
export:     to AS2
            announce NOT AS1
```

In this fragment, `aut-num` is the object that describes the AS by its number. The `import` clauses specify that the routers in AS 4 import all routes advertised by AS 1 routers and AS 2 routers. Note that the `pref` object differs from the local preference value in the route-selection algorithm: in the RPSL semantics a lower `pref` value is preferred over a larger one. In our case routes received from AS 1 and from AS 2 have equal values. The difference in the `med` values (which correspond to MULT_EXIT_DISC field) implies that shorter AS paths are preferred. The `export` clauses specify that AS 4 exports no route to AS 1 received from AS 2 and vice versa.

B.5 The Selective Export Rule

Using various kinds of simple path-vector protocols, in several studies it has been shown that certain configurations can force the BGP route propagation process to not converge (see, e.g., [16, 15, 8, 9, 10, 5]). Things are worse: in general the problem of deciding whether an explictly given configuration is bad in this sense is NP-complete, even for very extensional representations [15]. However, all these configurations are based on rather bizarre export-policy patterns, e.g., "always prefer a path of length two over other paths" [16]. In practice, routing policies are much more reasonable.

In the last section of this chapter we describe a prominent example of a policy pattern—the Selective Export Rule [1, 22, 12]. It is an extension of the most basic rule for policy routing "send routes only to paying customers" [40, p. 11] and applies to all business relationships among ASes.

In the following we consider an AS v in the AS graph. According to the business relationships we divide the set $N(v)$ of neighbors of v into the following sets:

- $\mathrm{Cust}(v)$ is the set of all customers of v.
- $\mathrm{Prov}(v)$ is the set of all providers of v.
- $\mathrm{Peer}(v)$ is the set of all peering partners of v, i.e., if $u \in \mathrm{Peer}(v)$ then u and v have a peer-to-peer relationship.

AS v exports to	**Provider**	**Customer**	**Peer**	**Sibling**
Own routes	Yes	Yes	Yes	Yes
Customer routes	Yes	Yes	Yes	Yes
Provider routes	No	Yes	No	Yes
Peer routes	No	Yes	No	Yes

Table B.5: The Selective Export Rule.

- $\mathrm{Sibl}(v)$ is the set of all siblings of v, i.e., if $u \in \mathrm{Sibl}(v)$ then u and v have a sibling-to-sibling relationship. We let $\mathrm{Sibl}(v)$ contain v as well.

Some of the sets may be empty.

Let $R(v)$ denote the set of all AS paths contained in the Loc-RIB of the AS v. Assumed that there are no misconfigurations of BGP, all AS paths in $R(v)$ are loopless and not including v. Here, we say that an AS path is *loopless* whenever between two sibling ASes on the path, no non-sibling AS is passed. Based on the neighborhood classification we further divide $R(v)$ into four categories. A loopless AS path $(u_1, \ldots, u_r) \in R(v)$ is

$$\begin{aligned}
\text{a } \textit{customer route} \text{ of } v &\iff_{\text{def}} \text{ leftmost } u_i \notin \mathrm{Sibl}(v) \text{ lies in } \mathrm{Cust}(v),\\
\text{a } \textit{provider route} \text{ of } v &\iff_{\text{def}} \text{ leftmost } u_i \notin \mathrm{Sibl}(v) \text{ lies in } \mathrm{Prov}(v),\\
\text{a } \textit{peer route} \text{ of } v &\iff_{\text{def}} \text{ leftmost } u_i \notin \mathrm{Sibl}(v) \text{ lies in } \mathrm{Peer}(v),\\
\text{an } \textit{own route} \text{ of } v &\iff_{\text{def}} \text{ for all } 1 \le i \le r,\ u_i \in \mathrm{Sibl}(v).
\end{aligned}$$

The Selective Export Rule summarized in Table B.5 provides a scheme of which route is allowed to be exported to a certain neighbor depending on the type of the route and the class of the neighbor.

Example: It is easily seen that all AS paths shown in the Loc-RIBs of our small Internet example in Figure B.4 result from export policies which respect the Selective Export Rule. In particular, consider AS 4 together with the RPSL fragment from Example B.4.5. Every route AS 4 receives from AS 1 is a provider route and should not be exported to the provider AS 2 according to Table B.5. Analogously each route received from AS 2 should not be exported to AS 1. The both `export` clauses of the RPSL fragment do exactly this.

Based on the Selective Export Rule some guidelines can be identified guaranteeing convergence of the BGP route propagation process. The following guideline has been proposed in [13]:

> Suppose an AS v possess two routes r_1 and r_2 containing the AS paths $(u_1, \ldots, u_k)$ and $(u'_1, \ldots, u'_\ell)$, respectively. If $u_1 \in \mathrm{Cust}(v)$ and $u'_1 \in \mathrm{Peer}(v) \cup \mathrm{Prov}(v)$ then the LOCAL_PREF of r_1 is higher than LOCAL_PREF of r_2

If all ASes respect the Selective Export Rule for their export policies and this guideline for the route-selection algorithms then it can be proven that the route propagation process always stops [13]. This guideline can be relaxed (at the cost of certain structural assumptions to the business relations in the AS graph) to equally ensure convergence. However, a general description of all policy patterns with convergence guarantee is not yet available.

Bibliography

[1] C. Alaettinoğlu. Scalable router configuration for the Internet. In *Proceedings of the 5th International Conference on Computer Communications and Networks (ICCCN'96)*. IEEE Computer Society Press, Washington, D.C., 1996.

[2] C. Alaettinoğlu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra. Routing Policy Specification Language (RPSL). RFC 2622, The Internet Society, 1999.

[3] R. Albrightson, J. J. Garcia-Luna-Aceves, and J. Boyle. EIGRP - a fast routing protocol based on distance vectors. In *Proceedings of the NetWorld/Interop Engineer Conference (NetWorld/Interop'94)*, 1994.

[4] R. W. Callon. Use of OSI IS-IS for routing in TCP/IP and dual environments. RFC 1195, The Internet Society, 1990.

[5] C. Chau, R. Gibbens, and T. G. Griffin. Towards a unified theory of policy-based routing. In *Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'06)*. IEEE Computer Society Press, Washington, D.C., 2006.

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

[7] S. E. Deering and R. M. Hinden. Internet Protocol, version 6 (IPv6) specification. RFC 2460, The Internet Society, 1998.

[8] N. Feamster, R. Johari, and H. Balakrishnan. Implications of autonomy for the expressiveness of policy routing. In *Proceedings of the ACM SIGCOMM 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'05)*, pages 25–36. ACM Press, New York, NY, 2005.

[9] J. Feigenbaum, D. R. Karger, V. S. Mirrokni, and R. Sami. Subjective-cost policy routing. In *Proceedings of the 1st International Workshop on Internet and Network Economics (WINE'05)*, volume 3828 of *Lecture Notes in Computer Science*, pages 174–183. Springer-Verlag, Berlin, 2005.

[10] J. Feigenbaum, D. R. Karger, V. S. Mirrokni, and R. Sami. Mechanism design for policy routing. *Distributed Computing*, 18(4):293–305, 2006.

[11] V. Fuller, T. Li, J. J. Y. Yu, and K. Varadhan. Classless Inter-Domain Routing (CIDR): An address assignment and aggregation strategy. RFC 1519, The Internet Society, 1993.

[12] L. Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, 2001.

[13] L. Gao and J. Rexford. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking*, 9(6):681–692, 2001.

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

[15] T. G. Griffin, F. B. Shepherd, and G. T. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10(2):232–243, 2002.

[16] T. G. Griffin and G. T. Wilfong. An analysis of BGP convergence properties. *ACM SIGCOMM Computer Communication Review*, 29(4):277–288, 1999.

[17] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an autonomous system (AS). RFC 1930, The Internet Society, 1996.

[18] C. Hedrick. Routing Information Protocol. RFC 1058, The Internet Society, 1988.

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

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

[21] B. F. Hummel. Automata-based IP packet classification, 2006. Diplomarbeit, Fachbereich Informatik, Technische Universität München, München.

[22] G. Huston. Interconnection, peering and settlements—Part II. *The Internet Protocol Journal*, 2(2):2–23, 1999.

[23] S. Kosub. Computational analysis of complex systems: Discrete foundations, algorithms, and the internet. Habilitation, Fakultät für Informatik, Technische Universität München, 2007.

[24] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley Longman, Amsterdam, 3rd edition, 2004.

[25] G. S. Malkin. RIP version 2 - carrying additional information. RFC 1723, The Internet Society, 1994.

[26] Z. M. Mao, R. Govindan, G. Varghese, and R. H. Katz. Route flap damping exacerbates Internet routing convergence. In *Proceedings of the ACM SIGCOMM 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'02)*, pages 221–233. ACM Press, New York, NY, 2002.

[27] D. Meyer, J. Schmitz, C. Orange, M. Prior, and C. Alaettinoğlu. Using RPSL in practice. RFC 2650, The Internet Society, 1999.

[28] Miniwatts Marketing Group. Internet world stats: Usage and population statistics. `http://www.internetworldstats.com`.

[29] J. Moy. The OSPF specification. RFC 1131, The Internet Society, 1989.

[30] J. Moy. OSPF version 2. RFC 2178, The Internet Society, 1997.

[31] D. Oran, editor. OSI IS-IS intra-domain routing protocol. RFC 1142, The Internet Society, 1990.

[32] J. B. Postel, editor. Internet Protocol – DARPA Internet program protocol specification. RFC 791, Information Sciences Institute, University of Southern California, Marina del Rey, CA, 1981.

[33] J. B. Postel, editor. Transmission Control Protocol – DARPA Internet program protocol specification. RFC 793, Information Sciences Institute, University of Southern California, Marina del Rey, CA, 1981.

[34] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771, The Internet Society, 1995.

[35] Y. Rekhter and T. Li, editors. An architecture for IP address allocation with CIDR. RFC 1518, The Internet Society, 1993.

[36] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, Upper Saddle River, NJ, 4th edition, 2002.

[37] P. Traina, R. Chandra, and T. Li. BGP community attribute. RFC 1997, The Internet Society, 1996.

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

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

[40] I. van Beijnum. *BGP*. O'Reilly & Associates, Sebastopol, CA, 2002.

[41] G. Varghese. *Network Algorithmics: An Interdisciplinary Approach to Designing Fast Networked Devices*. Morgan Kaufmann Publishers, San Francisco, CA, 2005.

[42] C. Villamizar, R. Chandra, and R. Govindan. BGP route flap damping. RFC 2439, The Internet Society, 1998.