Weighted Automata and Networks

Miroslav Ćirić

University of Niš, Faculty of Sciences
Departments of Computer Science and Mathematics
Niš, Serbia
miroslav.ciric@pmf.edu.rs

Seminar on Theoretical Computer Science and Discrete Mathematics
Aristotle University of Thessaloniki, Greece
October 30, 2018
I

Weighted and Multi-Valued Automata

◊ Weights and Truth values ◊
Weighted automaton model

★ **weighted automata** – classical nondeterministic automata in which the transitions carry weights

★ weights are also assigned to states – *initial* and *final* weights

★ formally $\mathcal{A} = (A, \sigma, \tau, \{\delta_x\}_{x \in X})$

$A$ – set of states with $|A| = n$

$X$ – fixed input alphabet

$\sigma, \tau : A \to K$ – *initial* and *final* vectors with entries in $K$

$\delta_x : A \times A \to K$ – *transition matrices* with entries in $K$

★ $\delta_x(a, b)$ – weight of a transition from $a$ to $b$ imposed by $x$

$\sigma(a) / \tau(a)$ – measure how much $a$ is an initial / final state
★ model certain quantitative properties
★ amount of resources needed for the execution of a transition
★ time needed for the execution
★ cost of the execution
★ probability of successful execution of a transition
★ reliability of successful execution . . .

★ operations on weights: multiplication and addition – accumulation of weights
★ structures of weights: semirings, bimonoids (distributivity is not necessary), . . .
Truth values

★ ordered algebraic structures – truth values
★ multi-valued logic – graded truth or intermediate truth
★ subtle nuances in reasoning – modeling of uncertainty
★ representation of imprecise aspects of human knowledge

★ ordering is essential – comparison of truth values
★ operations on truth values – logic conjectives
★ $[0, 1]$ (real unit interval), $\text{max}$, $\text{min}$ – Gödel structure
★ $[0, 1]$, t-norm, t-conorm – Łukasiewicz and product structure

★ linearity of the ordering is not essential
★ truth values – lattices, ordered algebraic structures
Multi-valued logics

Classical Boolean logic

Two-element Boolean algebra

Multi-valued logics with linearly ordered structures of truth values

Structures on \([0, 1]\) determined by \(t\)-norms

Multi-valued logics with general structures of truth values (not necessarily linearly ordered)

Lattices, residuated lattices, etc.
Structures of truth values (cont.)

★ *lattices*: finite infimum and supremum – *conjunction* and *disjunction* (intersection and union)
★ *complete lattices*: infinite infimum and supremum – *universal* and *existential quantifiers*

★ *infimum does not necessarily distribute over supremum* (except in distributive lattices)
★ *new operation*: *multiplication* $\otimes$ – *distributes over suprema*  
  *strong conjunction*

★ *lattice ordered monoid* 
  monoid + partial order (compatible w.r.t. multiplication $\otimes$)  
lattice w.r.t. this partial order 
  $\otimes$ *distributes over (finite) suprema*  
  $(\lor, \otimes)$-reduct – *semiring reduct*
Residuation. Residuated functions

★ How to model the implication?

★ Residuated function

★ \((P, \leq), (Q, \leq)\) – partially ordered sets, \(f : P \rightarrow Q\)

★ \(f\) is residuated if there is \(g : Q \rightarrow P\) satisfying

\[
f(x) \leq y \iff x \leq g(y)
\]

★ residuation property

★ if exists, such \(g\) is unique

★ it is called the residual of \(f\) and denoted by \(f^\#\)
Theorem on residuated functions

The following conditions for \( f : P \to Q \) are equivalent:

(i) \( f \) is residuated;

(ii) \( f \) is isotone and there is an isotone \( g : Q \to P \) such that
\[
I_P \leq f \circ g, \quad g \circ f \leq I_Q;
\]

(iii) the inverse image under \( f \) of every principal down-set of \( Q \) is a principal down-set of \( P \);

(iv) \( f \) is isotone and the set \( \{x \in P \mid f(x) \leq y\} \) has the greatest element, for every \( y \in Q \).

★ principal down-set: \( a \downarrow = \{x \in P \mid x \leq a\} \)

★ lattice-theoretical counterpart of a continuous function

★ \( f^\#(y) = \top \{x \in P \mid f(x) \leq y\} \) (\( \top H \) – greatest element of \( H \))
Residuated algebraic structures. Residuated semigroups

★ $(S, \otimes, \leq)$ – ordered semigroup
★ $\leq$ is compatible w.r.t. $\otimes$
★ for $a \in S$, functions $\lambda_a, \varrho_a : S \to S$ are defined by

$$\lambda_a(x) = a \otimes x,$$
$$\varrho_a(x) = x \otimes a$$
★ $\lambda_a$ – inner left translation w.r.t. $a$
★ $\varrho_a$ – inner right translation w.r.t. $a$
★ residuated semigroup – $\lambda_a$ and $\varrho_a$ are residuated functions
★ $a \backslash b = \lambda^\#_a(b) = \top\{x \in S \mid a \otimes x \leq b\}$ – right residual of $b$ by $a$
★ $b / a = \varrho^\#_a(b) = \top\{x \in S \mid x \otimes a \leq b\}$ – left residual of $b$ by $a$
★ residuation property

$$a \otimes b \leq c \iff b \leq a \backslash c \iff a \leq c / b$$
Quantales

★ **Quantale**

ordered semigroup \((S, \otimes, \leq)\)
complete lattice w.r.t. \(\leq\)
\(\otimes\) distributes over arbitrary suprema (finite and infinite)

★ \(\otimes\) is not necessarily commutative

★ **infinite distributivity** \(\Rightarrow\) existence of residuals

\[
\begin{align*}
a \backslash b &= \bigvee \{ x \in S \mid a \otimes x \leq b \} = \top \{ x \in S \mid a \otimes x \leq b \} \\
b / a &= \bigvee \{ x \in S \mid x \otimes a \leq b \} = \top \{ x \in S \mid x \otimes a \leq b \}
\end{align*}
\]

★ **unital quantale** – with a multiplicative unit \(e\)

★ \((S, \lor, \otimes, 0, e)\) – semiring (semiring reduct)

★ **integral quantale** – \(e\) is the greatest element \((e = 1)\)
Residuated lattices

★ general meaning: lattice-ordered residuated semigroup
★ not necessarily commutative (left and right residuals)
★ not necessarily complete, not necessarily bounded
★ multi-valued logic – requires commutativity and completeness

★ Residuated lattice – algebra $\mathbb{L} = (L, \lor, \land, \otimes, \to, 0, 1)$
   $(L, \lor, \land, 0, 1)$ – bounded lattice with the least element 0
   and the greatest element 1
   $(L, \otimes, 1)$ – commutative monoid with the unit 1
   $\otimes$ and $\to$ satisfy the residuation property

\[ x \otimes y \leq z \iff x \leq y \to z \]

★ Complete residuated lattice – the lattice reduct is complete
  commutative integral quantale
only one residual (left and right residuals coincide)

operation → : residuum or residual implication
models the implication

residuation property $x \otimes y \leq z \iff x \leq y \rightarrow z$

modus ponens rule
deduction theorem

biresiduum or residual equivalence:

$x \leftrightarrow y = (x \rightarrow y) \land (y \rightarrow x)$
models the equivalence

negation: $\neg x = x \rightarrow 0$
Special residuated structures

on \([0, 1]\) with \(x \land y = \min(x, y)\) and \(x \lor y = \max(x, y)\)

★ *Gödel structure*

\[ x \otimes y = \min(x, y), \quad x \rightarrow y = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise} \end{cases} \]

★ *Product structure* or *Goguen structure*

\[ x \otimes y = x \cdot y, \quad x \rightarrow y = \begin{cases} 1 & \text{if } x \leq y \\ \frac{y}{x} & \text{otherwise} \end{cases} \]

★ *Łukasiewicz structure*

\[ x \otimes y = \max(x + y - 1, 0), \quad x \rightarrow y = \min(1 - x + y, 1) \]

★ Gödel and Łukasiewicz structure on finite chains in \([0, 1]\)

★ *Heyting algebra*: \(\mathbb{L}\) with \(\otimes = \land\) (bounded *Brouwer lattice*)
Multi-valued structures and logics

Quantales
Substructural logics

Complete residuated lattices
Residuated logics

MTL-algebras
MTL-logic
Standard MTL-algebras

BL-algebras
Basic logic
Standard BL-algebras

Heyting algebras
Intuitionistic logic

Product algebras
(Goguen algebras)
Product logic
Product structure

MV-algebras
(Wajsberg algebras)
Łukasiewicz logic
Łukasiewicz structure

Gödel algebras
Gödel logic
Gödel structure

Boolean algebras
Classical logic
Boolean structure
Residuation: some historical notes

- **Dedekind** (1894) – quantales of ideals of rings
- **Schröder**, *Algebra und Logik der Relative* (Leipzig, 1895) – quantales of binary relations
- **Brouwer** (1920s) – relative pseudo complementation
- **Heyting** (1930) – Heyting algebras
- **Ward, Dilworth** (1930s) – (noncommutative) residuated lattices, arithmetical applications
- **Mulvey** (1986) – quantale of closed linear subspaces of a non-commutative $C^*$-algebra
  - applications in functional analysis, topology
  - *Gelfand, von Neumann, and Hilbert quantales*
II

Weighted Automata

◊ State Reduction ◊
Back to automata: Behaviour and Equivalence

★ behaviour of a WFA $\mathcal{A}$ or language recognized by $\mathcal{A}$

$$\llbracket \mathcal{A} \rrbracket : X^* \rightarrow S \text{ given by } \llbracket \mathcal{A} \rrbracket (u) = \sigma \cdot \delta_u \cdot \tau$$

$$\delta_u = \delta_{x_1} \cdots \delta_{x_s}, \text{ if } u = x_1 \cdots x_s, x_1, \ldots, x_s \in X$$

★ $\mathcal{A} = (A, X, \sigma^A, \tau^A, \{\delta^A_x\}_{x \in X})$ and $\mathcal{B} = (B, X, \sigma^B, \tau^B, \{\delta^B_x\}_{x \in X})$ are equivalent WFAs if $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{B} \rrbracket$

★ State Reduction Problem:

Provide efficient methods for constructing a reasonably small WFA equivalent to $\mathcal{A}$ (not necessarily minimal)

★ Equivalence Problem:

Provide efficient methods for testing whether two WFAs $\mathcal{A}$ and $\mathcal{B}$ are equivalent
State reduction

★ **Minimization of NFA** – *computationally hard* problem
★ the same goes for weighted and multi-valued automata
★ more practical – *state reduction problem*

give an *efficient construction of a reasonably small* automaton
(not necessarily minimal) *equivalent* to the given automaton

★ *How to make a state reduction?*

★ our main ideas came from algebra – *quotient algebra*
★ *congruences* – compatible equivalence relations
★ elements of the quotient algebra – *equivalence classes*
★ *rows or columns* in the corresponding Boolean matrix
Row automata

- \((S, +, \cdot, 0, e)\) – semiring with the unit \(e\)
- \(A = (A, \sigma, \tau, \{\delta_x\}_{x \in X})\) – weighted finite automaton over \(S\)
- \(\pi \in S^{A \times A}\) – given matrix

- **Our idea:** Construct an WFA whose states would be rows of \(\pi\)
- \(a\pi – a\)-row of \(\pi – \pi(a, \cdot)\), \(\pi b – b\)-column of \(\pi – \pi(\cdot, b)\)
- \(\overline{A}\) – the set of all different rows of \(\pi\)
- define \(\overline{\sigma}, \overline{\tau} : \overline{A} \rightarrow S, \overline{\delta}_x : \overline{A} \times \overline{A} \rightarrow S, x \in X\) by

\[
\overline{\sigma}(a\pi) = (\sigma \cdot \pi)(a) = \sigma \cdot (\pi a) \\
\overline{\tau}(a\pi) = (\pi \cdot \tau)(a) = (a\pi) \cdot \tau \\
\overline{\delta}_x(a\pi, b\pi) = (\pi \cdot \delta_x \cdot \pi)(a, b) = (a\pi) \cdot \delta_x \cdot (\pi b)
\]
**Question:** Are these definitions good?

**Answer:** Not necessarily.

If \( a\pi = a'\pi \) and \( b\pi = b'\pi \), it does not have to be

\[
\sigma \cdot (\pi a) = \sigma \cdot (\pi a') \quad \text{or} \quad (a\pi) \cdot \delta_x \cdot (\pi b) = (a'\pi) \cdot \delta_x \cdot (\pi b')
\]

**Question:** Under what conditions the definitions are good?

**Answer:**

- we need a partial order \( \leq \) on \( S \) (not necessarily compatible)
- a square matrix \( \pi : A \times A \to S \) is **reflexive** if \( e \leq \pi(a, a) \), for all \( a \in A \)
  
  **transitive** if \( \pi(a, b) \cdot \pi(b, c) \leq \pi(a, c) \), for all \( a, b, c \in A \)

**Quasi-order matrix** – reflexive and transitive matrix
Theorem 1

Let $\pi \in S^{A \times A}$ be a quasi-order matrix and $a, b \in A$. Then the following conditions are equivalent:

(i) $\pi(a, b) = \pi(b, a) = e$

(ii) $a\pi = b\pi$

(iii) $\pi a = \pi b$

Theorem 2

Let $\pi \in S^{A \times A}$ be a quasi-order matrix. Then $\sigma$, $\tau$ and $\delta_x$ are well-defined and $\overline{\mathcal{A}} = (\overline{A}, \overline{\sigma}, \overline{\tau}, \{\overline{\delta}_x\}_{x \in X})$ is an WFA satisfying $|\overline{\mathcal{A}}| \leq |\mathcal{A}|$.

$\overline{\mathcal{A}}$ – row automaton – isomorphic to column automaton
Notes on quasi-order matrices

★ \( S \) – semiring with unit \( e \), \( \leq \) – partial order on \( S \)
★ partial order on \( S^{A \times A} \) is defined entrywise
\[
\mu \leq \eta \iff \mu(a, b) \leq \eta(a, b), \text{ for all } a, b \in S
\]
★ if the ordering on \( S \) is compatible, then the ordering of matrices is also compatible
★ for a matrix \( \pi \) with entries in a lattice-ordered monoid
\[
\pi \text{ is reflexive} \iff \Delta \leq \pi \implies \pi \leq \pi^2
\]
\[
\pi \text{ is transitive} \iff \pi^2 \leq \pi
\]
\[
\pi \text{ is a quasi-order matrix} \implies \pi^2 = \pi
\]
★ \( \Delta(a, a) = e \) (the unit), \( \Delta(a, b) = 0 \), for \( a \neq b \) – unit matrix
★ **Question:** Under what conditions this holds for matrices with entries in a semiring?
★ \( S \) – *positively ordered semiring* – compatible partial order \( \leq \) and \( 0 \) is the least element

\[
\pi \text{ is reflexive} \implies \Delta \leq \pi \implies \pi \leq \pi^2
\]

★ to prove

\[
\pi \text{ is transitive} \implies \pi^2 \leq \pi
\]

we need something like

\[
a_1, \ldots, a_s \leq a \implies a_1 + \ldots + a_s \leq a
\]

the addition behaves somehow like supremum

★ **Question:** *In which class of semirings all of this is true?*

★ **Answer:** *Additively idempotent semirings*
Additively idempotent semirings

- $a + a = a$, for every $a \in S$ (equivalently $1 + 1 = 1$)
- positively partially ordered
- partial ordering: $a \leq b \iff a + b = b$
- supremum coincides with addition
- every quasi-order matrix $\pi$ satisfies $\pi^2 = \pi$

**Question**: Why $\pi^2 = \pi$ is so important?

- behaviour of the row automaton $\overline{A}$ (the general case)

$$\llbracket \overline{A} \rrbracket(\varepsilon) = \sigma \cdot \pi^2 \cdot \tau$$
$$\llbracket \overline{A} \rrbracket(x_1 x_2 \cdots x_k) = \sigma \cdot \pi^2 \cdot \delta_{x_1} \cdot \pi^2 \cdot \delta_{x_2} \cdot \pi^2 \cdots \pi^2 \cdot \delta_{x_k} \cdot \pi^2 \cdot \tau$$

- with $\pi^2 = \pi$ we avoid squares
Additively idempotent semirings – importance

★ the basic concept of *idempotent analysis*

the usual arithmetic operations (+, ·) are replaced by a new set of basic operations – semiring operations (max, +), (min, +), etc.

some problems that are non-linear in the traditional analysis turn out to be linear over a suitable semiring

*tropical mathematics, tropical geometry ... (tropical semiring)*

★ *algebraic path problems* (generalization of the shortest path problem in graphs)

★ *optimization problems* (including *dynamic programming*)

★ *discrete-event systems*

★ *automata and formal language theory*
★ **Question:** Is $\mathcal{A}$ equivalent to $\overline{\mathcal{A}}$?  
**Answer:** Not necessarily.

★ **Question:** Under what conditions they are equivalent?

$\llbracket \mathcal{A} \rrbracket (\varepsilon) = \sigma \cdot \tau$

$\llbracket \mathcal{A} \rrbracket (x_1 x_2 \cdots x_k) = \sigma \cdot \delta_{x_1} \cdot \delta_{x_2} \cdots \delta_{x_k} \cdot \tau$

$\llbracket \overline{\mathcal{A}} \rrbracket (\varepsilon) = \sigma \cdot \pi \cdot \tau$

$\llbracket \overline{\mathcal{A}} \rrbracket (x_1 x_2 \cdots x_k) = \sigma \cdot \pi \cdot \delta_{x_1} \cdot \pi \cdot \delta_{x_2} \cdot \pi \cdots \pi \cdot \delta_{x_k} \cdot \pi \cdot \tau$

★ $\pi$ has to be a solution of the general system

$\sigma \cdot \tau = \sigma \cdot \pi \cdot \tau$

$\sigma \cdot \delta_{x_1} \cdot \delta_{x_2} \cdots \delta_{x_k} \cdot \tau = \sigma \cdot \pi \cdot \delta_{x_1} \cdot \pi \cdot \delta_{x_2} \cdot \pi \cdots \pi \cdot \delta_{x_k} \cdot \pi \cdot \tau$
Notes on the general system

★ it may consist of infinitely many equations
★ can not be solved efficiently
★ we have to find as possible greater solutions (greater solutions provide better reductions)
★ in the general case, there is no the greatest solution

★ instances of the general system
★ systems whose any solution is a solution to the general system
★ we need instances with finitely many equations or inequations which have the greatest solution and can be solved efficiently
III

Weakly linear systems

- The General Results
Weakly linear systems (The general form)

★ \( I = I_1 \cup I_2 \cup I_3 \cup I_4 \) – nonempty set

★ \( A, B \) – nonempty sets

★ \( \{\alpha_i\}_{i \in I} \in S^{A \times A}, \{\beta_i\}_{i \in I} \in S^{B \times B} \) – given families of matrices

★ \( \mu \) – unknown taking values in \( S^{A \times B} \)

★ \textit{weakly linear system}

\[
\begin{align*}
\alpha_i \cdot \mu & \leq \mu \cdot \beta_i, \quad i \in I_1, \\
\mu \cdot \beta_i & \leq \alpha_i \cdot \mu, \quad i \in I_2, \\
\mu^\top \cdot \alpha_i & \leq \beta_i \cdot \mu^\top, \quad i \in I_3, \\
\beta_i \cdot \mu^\top & \leq \mu^\top \cdot \alpha_i, \quad i \in I_4,
\end{align*}
\]

★ \textit{homogeneous system} – \( A = B \) and \( \alpha_i = \beta_i \), for all \( i \in I \)

★ otherwise – \textit{heterogeneous system}
The greatest solutions

**Theorem 1**

For an arbitrary \( \gamma_0 \in S^{A \times B} \) the exists the greatest solution of the WLS which is less than or equal to \( \gamma_0 \).

★ in the case of an homogeneous WLS we have

**Theorem 2**

Let \( \gamma_0 \) be a quasi-order matrix, and \( \gamma \) the greatest solution of the WLS such that \( \gamma \leq \gamma_0 \). Then \( \gamma \) is also a quasi-order matrix.

★ How to compute the greatest solutions?
The function $\phi$

\[
\begin{align*}
\alpha_i \cdot \mu & \leq \mu \cdot \beta_i, & i \in I_1, \\
\mu \cdot \beta_i & \leq \alpha_i \cdot \mu, & i \in I_2, \\
\mu^\top \cdot \alpha_i & \leq \beta_i \cdot \mu^\top, & i \in I_3, \\
\beta_i \cdot \mu^\top & \leq \mu^\top \cdot \alpha_i, & i \in I_4.
\end{align*}
\]

**Definition**

A function $\phi : S^{A \times B} \rightarrow S^{A \times B}$ is defined as follows

\[
\phi(\gamma) = \left( \bigwedge_{i \in I_1} \alpha_i \backslash (\gamma \cdot \beta_i) \right) \land \left( \bigvee_{i \in I_2} (\alpha_i \cdot \gamma) / \beta_i \right) \\
\land \left( \bigwedge_{i \in I_3} [(\beta_i \cdot \gamma^\top) / \alpha_i]^\top \right) \land \left( \bigvee_{i \in I_4} [\beta_i \backslash (\gamma^\top \cdot \alpha_i)]^\top \right),
\]

The function $\phi$ (cont.)

Theorem 3

- $\phi$ is an isotone function on the complete lattice $S^{A \times B}$
- the considered WLS is equivalent to the inequation
  \[ \mu \leq \phi(\mu) \]

- $\gamma \leq \phi(\gamma)$ – $\gamma$ is a post-fixed point of $\phi$
- $\gamma = \phi(\gamma)$ – $\gamma$ is a fixed point of $\phi$

- solving the WLS $\equiv$ computing post-fixed points of $\phi$
Knaster-Tarski Fixed Point Theorem

Let $L$ be a complete lattice, $\phi : L \rightarrow L$ an isotone function, and $a_0 \in L$

- there exists the greatest post-fixed point $a$ of $\phi$ satisfying $a \leq a_0$
- $a$ is also the greatest fixed point of $\phi$ satisfying $a \leq a_0$

- Knaster-Tarski theorem provides the existence of the greatest solution of WLS which is less or equal to a given $\gamma_0 \in S^{A \times B}$
- it does not provide a way to compute this solution

- **Problem:** How to compute the greatest solutions?
Kleene Fixed Point Theorem

Let $L$ be a complete lattice, $\phi : L \rightarrow L$ an isotone function, and $a_0 \in L$

Define Kleene’s descending chain $\{a_k\}_{k \in \mathbb{N}}$ of $\phi$ by

$$a_1 = a_0, \quad a_{k+1} = a_k \land \phi(a_k)$$

★ If the chain stabilizes at some $a_k$ (i.e. $a_{k+1} = a_k$), then $a_k$ is the greatest fixed point of $\phi$ less than or equal to $a_0$

★ if $\phi$ is Scott-continuous (i.e., it preserves lower-directed infima), then the greatest fixed point of $\phi$ less than or equal to $a_0$ is

$$a = \bigwedge_{k \in \mathbb{N}} a_k$$
The greatest solutions of WLS

★ for $\gamma_0 \in S^{A \times B}$ we consider Kleene’s descending chain $\{\gamma_k\}_{k \in \mathbb{N}}$ given by

$$\gamma_1 = \gamma_0, \quad \gamma_{k+1} = \gamma_k \land \phi(\gamma_k)$$

★ if the subalgebra of $S$ generated by all entries of matrices $\alpha_i$, $\beta_i$ and $\gamma_0$ satisfies DCC, the chain stabilizes at some $\gamma_k$

★ then $\gamma_k$ is the greatest solution of the WLS

★ special cases: $S$ satisfies DCC or is locally finite

★ special case: $S$ is the max-plus quantale

– $\phi$ is Scott-continuous (i.e., $\omega$-continuous)

– the greatest solution which is less than or equal to $\gamma_0$ is

$$\gamma = \bigwedge_{k \in \mathbb{N}} \gamma_k$$

Miroslav Ćirić

Weighted Automata and Networks
Max-plus quantale

- \( \mathbb{R}_\infty = \mathbb{R} \cup \{ -\infty, +\infty \} \)
- usual ordering
- multiplication

\[
a \otimes b = \begin{cases} 
    a + b & \text{if } a, b \in \mathbb{R}, \\
    -\infty & \text{if } a = -\infty \text{ or } b = -\infty, \\
    +\infty & \text{if } a = +\infty, b \neq -\infty \text{ or } a \neq -\infty, b = +\infty,
\end{cases}
\]

- commutative unital quantale
- residuation

\[
a \rightarrow b = \begin{cases} 
    b - a & \text{if } a, b \in \mathbb{R}, \\
    -\infty & \text{if } b = -\infty, a \neq -\infty, \\
    +\infty & \text{if } b = +\infty \text{ or } a = b = -\infty.
\end{cases}
\]
Related quantales

★ $Q_\infty = Q \cup \{-\infty, +\infty\}$, $\mathbb{Z}_\infty = \mathbb{Z} \cup \{-\infty, +\infty\}$

★ subquantales of $\mathbb{R}_\infty$

★ $\mathbb{R}_{\geq 0} \cup \{+\infty\}$, with multiplication and residuation

\[
a \otimes b = \begin{cases} 
a + b & \text{if } a, b \in \mathbb{R}_{\geq 0}, \\
+\infty & \text{if } a = +\infty \text{ or } b = +\infty, \end{cases}
\]

\[
a \rightarrow b = \begin{cases} 
b - a & \text{if } a, b \in \mathbb{R}_{\geq 0} \text{ and } a \leq b, \\
0 & \text{if } b \in \mathbb{R}_{\geq 0} \text{ and } a > b, \\
+\infty & \text{if } b = +\infty. \end{cases}
\]

★ subquantales $\mathbb{Q}_{\geq 0} \cup \{+\infty\}$ and $\mathbb{Z}_{\geq 0} \cup \{+\infty\}$,
\[ a \otimes b = \begin{cases} 
  a \cdot b & \text{if } a, b \in \mathbb{R}_{\geq 0}, \\
  0 & \text{if } a = 0 \text{ or } b = 0, \\
  +\infty & \text{if } a = +\infty, \ b \neq 0, \text{ or } b = +\infty, \ a \neq 0
\end{cases} \]

\[ a \rightarrow b = \begin{cases} 
  b/a & \text{if } a \in \mathbb{R}_{> 0}, \ b \in \mathbb{R}_{\geq 0}, \\
  0 & \text{if } a = +\infty, \ b \in \mathbb{R}_{\geq 0}, \\
  +\infty & \text{if } a = 0 \text{ or } b = +\infty.
\end{cases} \]

\* max-min quantale or fuzzy algebra – \( \mathbb{R} \cup \{-\infty, +\infty\} \), with

\[ a \otimes b = a \wedge b, \quad a \rightarrow b = \begin{cases} 
  b & \text{if } a > b, \\
  +\infty & \text{if } a \leq b.
\end{cases} \]
★ **Max-plus semiring** – carrier $\mathbb{R}_{\text{max}} = \mathbb{R} \cup \{-\infty\}$

★ **Open problem:** How to avoid $+\infty$ as an entry in matrices $\gamma_k$?

★ if $\tau(a) = -\infty$, for some $a \in A$, then

$$\gamma_0(a, a) = (\tau/\tau)(a, a) = \tau(a) \rightarrow \tau(a) = -\infty \rightarrow -\infty = +\infty$$

★ To replace $+\infty$ on the diagonal of $\gamma_0$ by an enough big real number?

★ the final result should be a quasi-order matrix

★ the new starting matrix should also be quasi-order matrix
IV

Weighted Automata

◊ Back to State Reduction ◊
Right and left invariant matrices

★ \( \mathcal{A} = (A, \sigma, \tau, \{\delta_x\}_{x \in X}) \) – weighted automaton over a quantale \( S \)

★ right invariant matrices – solutions of

\[
\mu \cdot \delta_x \leq \delta_x \cdot \mu, \quad x \in X,
\]
\[
\mu \cdot \tau \leq \tau
\]

★ left invariant matrices – solutions of

\[
\delta_x \cdot \mu \leq \mu \cdot \delta_x, \quad x \in X,
\]
\[
\sigma \cdot \mu \leq \sigma
\]

★ \( \mu \cdot \tau \leq \tau \iff \mu \leq \tau/\tau \) and \( \sigma \cdot \mu \leq \sigma \iff \mu \leq \sigma\setminus\sigma \)

★ \( \tau/\tau, \sigma\setminus\sigma \in S^{A \times A} \) are quasi-order matrices given by

\[
(\tau/\tau)(a, b) = \tau(a)/\tau(b), \quad (\sigma\setminus\sigma)(a, b) = \sigma(a)\setminus\sigma(b)
\]
Computation of the greatest invariant matrices

★ How to compute the greatest right and left invariant matrices?

★ they can be computed as the greatest solutions of the corresponding WLS that are less than or equal to $\tau/\tau$ or $\sigma/\sigma$

★ Why we need right and left invariant matrices?

★ right and left invariant matrices are solutions of the general system

★ right and left invariant matrices provide state reductions that may be efficiently realised
★ $\mathcal{A}$ – WFA, $\pi$ – the greatest left invariant q-o.m. on $\mathcal{A}$ when $\mathcal{A}$ is reduced by means of $\pi$, the resulting row automaton $\mathcal{A}/\pi$ can not be reduced by means of right invariant q-o.m. however, it could be reduced by means of left invariant q-o.m.

★ Alternate reductions

we alternately make a series of reductions by means of the greatest right and left invariant q-o.m., or vice versa

this procedure will be interrupted when we get an automaton that can not be reduced by means of alternate reductions
V

Weighted Automata

◊ Equivalence ◊

◊ Simulation and Bisimulation ◊
Equivalence of WFAs

★ **Equivalence Problem:** Provide efficient methods for testing whether two WFAs $A$ and $B$ are equivalent

★ Equivalence Problem is computationally hard

★ equivalence of WFAs can not be expressed through matrices, as some kind of relationship between states

★ **simulation:** $A \times B$-matrix which provides that $B$ simulates $A$

★ **bisimulation:** $A \times B$-matrix which, together with its transpose, provides that $B$ and $A$ simulate each other

★ existence of a bisimulation implies equivalence of $A$ and $B$

bisimulations provide *approximations of equivalence*

★ **simulations** and **bisimulations** – defined as solutions of particular systems of matrix inequations
Simulations

★ $S$ – unital quantale

★ WFAs $\mathcal{A} = (A, X, \sigma^A, \tau^A, \{\delta^A_x\}_{x \in X}), \quad \mathcal{B} = (B, X, \sigma^B, \tau^B, \{\delta^B_x\}_{x \in X})$

★ forward simulations – solutions of
\[
\begin{align*}
\mu^\top \cdot \delta^A_x & \leq \delta^B_x \cdot \mu^\top, \quad x \in X, \\
\mu^\top \cdot \tau^A & \leq \tau^B \
\end{align*}
\]
(equivalently $\mu \leq \gamma_0 = (\tau^B / \tau^A)^\top$)
\[
\begin{align*}
\sigma^A & \leq \sigma^B \cdot \mu^\top 
\end{align*}
\]

★ backward simulations – solutions of
\[
\begin{align*}
\delta^A_x \cdot \mu & \leq \mu \cdot \delta^B_x, \quad x \in X, \\
\sigma^A \cdot \mu & \leq \sigma^B \
\end{align*}
\]
(equivalently $\mu \leq \gamma_0 = \sigma^A \setminus \sigma^B$)
\[
\begin{align*}
\tau^A & \leq \mu \cdot \tau^B 
\end{align*}
\]
Forward and backward simulations
	nondeterministic automata

\[ \sigma^A \]

\[ \tau^A \]

\[ \sigma^B \]

\[ \tau^B \]
Forward and backward simulations

nondeterministic automata

\[ \sigma^A \rightarrow Q \rightarrow \sigma^B \]

\[ \tau^A \rightarrow \rightarrow \tau^B \]

\[ A \rightarrow Q \rightarrow B \]
Forward and backward simulations

nondeterministic automata

\[ a_0 \rightarrow \sigma^A \]

\[ a_k \rightarrow x_k \rightarrow a_{k+1} \rightarrow x_{k+1} \rightarrow \cdots \rightarrow a_n \rightarrow x_n \rightarrow \tau^A \]

\[ \sigma^B \rightarrow Q \rightarrow \text{forward simulation} \rightarrow \tau^B \]

arbitrary successful run
Forward and backward simulations

nondeterministic automata

\[ \sigma^A \]

\[ \sigma^B \]

\[ \tau^A \]

\[ \tau^B \]

\[ a_0 \]

\[ a_k \]

\[ a_{k+1} \]

\[ a_n \]

\[ x_1 \]

\[ x_k \]

\[ x_{k+1} \]

\[ x_{k+2} \]

\[ x_n \]

\[ x_1 \]

\[ x_k \]

\[ x_{k+1} \]

\[ x_{k+2} \]

\[ x_n \]

We have to build an arbitrary successful run which simulates the original run.

\[ \emptyset \]

forward simulation

\[ A \]

\[ B \]
Forward and backward simulations

nondeterministic automata

\[ a_0 \xrightarrow{x_1} a_k \xrightarrow{x_k} a_{k+1} \xrightarrow{x_{k+1}} a_n \]

\[ b_0 \xrightarrow{x_1} b_k \xrightarrow{x_k} b_{k+1} \xrightarrow{x_{k+1}} b_n \]

\[ \tau^A \rightarrow a_0 \rightarrow a_k \rightarrow a_{k+1} \rightarrow a_n \]

\[ \tau^B \rightarrow b_0 \rightarrow b_k \rightarrow b_{k+1} \rightarrow b_n \]

\[ Q \xrightarrow{\sigma^A} \rightarrow a_0 \rightarrow a_k \rightarrow a_{k+1} \rightarrow a_n \]

\[ Q \xrightarrow{\sigma^B} \rightarrow b_0 \rightarrow b_k \rightarrow b_{k+1} \rightarrow b_n \]

Arbitrary successful run

we have to build a successful run which simulates the original run
Forward and backward simulations

nondeterministic automata

we have to build a successful run which simulates the original run
Forward and backward simulations

nondeterministic automata

\[ \sigma^A \leq \sigma^B \cdot Q^\top \]

we have to build a successful run which simulates the original run
Forward and backward simulations

\[ \sigma^A \leq \sigma^B \cdot Q^T \]

nondeterministic automata

we have to build a successful run which simulates the original run

arbitrary successful run
Forward and backward simulations

nondeterministic automata

\[ \sigma^A \leq \sigma^B \cdot \mathcal{Q}^\top \]
Forward and backward simulations

nondeterministic automata

A

\[ \sigma^A \]

\[ a_0 \]

\[ \cdots \]

\[ a_k \]

\[ a_{k+1} \]

\[ \cdots \]

\[ a_n \]

\[ \tau^A \]

B

\[ \sigma^B \]

\[ b_0 \]

\[ \cdots \]

\[ b_k \]

\[ b_{k+1} \]

\[ \cdots \]

\[ b_n \]

\[ \tau^B \]

\[ \delta^A_{x_{k+1}} \]

\[ \mathcal{Q} \]

forward simulation

we have to build a successful run which simulates the original run

\[ (1) \quad \sigma^A \leq \sigma^B \cdot \mathcal{Q}^\top \]
Forward and backward simulations

nondeterministic automata

\( \sigma^A \leq \sigma^B \cdot \varrho^T \)
Forward and backward simulations

nondeterministic automata

\[
\begin{align*}
\sigma^A &\leq \sigma^B \cdot \mathcal{Q}^\top
\end{align*}
\]
Forward and backward simulations

nondeterministic automata

\[ \sigma^A \leq \sigma^B \cdot \mathcal{Q}^\top \]
Forward and backward simulations

nondeterministic automata

\[ \sigma^A \leq \sigma^B \cdot \varphi^\top \]

\[ \varphi^\top \cdot \delta^A_x \leq \delta^B_x \cdot \varphi^\top \]

we have to build a successful run which simulates the original run

арbitrary successful run

(1) \( \sigma^A \leq \sigma^B \cdot \varphi^\top \)

(2) \( \varphi^\top \cdot \delta^A_x \leq \delta^B_x \cdot \varphi^\top \)
Forward and backward simulations

nondeterministic automata

\[ A \stackrel{\sigma^A}{\longrightarrow} a_0 \stackrel{x_1}{\longrightarrow} \cdots \stackrel{x_k}{\longrightarrow} a_k \stackrel{x_{k+1}}{\longrightarrow} a_{k+1} \stackrel{x_{k+2}}{\longrightarrow} \cdots \stackrel{x_n}{\longrightarrow} a_n \]

\[ B \stackrel{\sigma^B}{\longrightarrow} b_0 \stackrel{x_1}{\longrightarrow} \cdots \stackrel{x_k}{\longrightarrow} b_k \stackrel{x_{k+1}}{\longrightarrow} b_{k+1} \stackrel{x_{k+2}}{\longrightarrow} \cdots \stackrel{x_n}{\longrightarrow} b_n \]

\[ Q \text{ forward simulation} \]

\[ \sigma^A \preceq \sigma^B \cdot Q^\top \quad (1) \quad Q^\top \cdot \delta^A_x \preceq \delta^B_x \cdot Q^\top \quad (2) \]

we have to build a successful run which simulates the original run

\[ a_{k+1} \]

\[ b_{k+1} \]
Forward and backward simulations

nondeterministic automata

We have to build a successful run which simulates the original run.

(1) \( \sigma^A \leq \sigma^B \cdot \varrho^T \)

(2) \( \varrho^T \cdot \delta^A_x \leq \delta^B_x \cdot \varrho^T \)
Forward and backward simulations

nondeterministic automata

\( \sigma^A \leq \sigma^B \cdot \varrho^\top \)  \( \varrho^\top \cdot \delta^A_x \leq \delta^B_x \cdot \varrho^\top \)  \( \varrho^\top \cdot \tau^A \leq \tau^B \)

we have to build a successful run which simulates the original run
**Forward and backward simulations**

**nondeterministic automata**

For any successful run \( \sigma^A \) in automaton \( A \) and \( \sigma^B \) in automaton \( B \), we have to build a successful run which simulates the original run.

**forward simulation** – the sequence \( b_0, \ldots, b_n \) is built starting from \( b_0 \) and ending with \( b_n \).

**backward simulation** – the sequence \( b_0, \ldots, b_n \) is built starting from \( b_n \) and ending with \( b_0 \).
### Bisimulations

<table>
<thead>
<tr>
<th>type of bisimulations</th>
<th>$\mathcal{Q}$</th>
<th>$\mathcal{Q}^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>forward</td>
<td>forward</td>
<td>forward</td>
</tr>
<tr>
<td>backward</td>
<td>backward</td>
<td>backward</td>
</tr>
<tr>
<td>forward-backward</td>
<td>forward</td>
<td>backward</td>
</tr>
<tr>
<td>backward-forward</td>
<td>backward</td>
<td>forward</td>
</tr>
</tbody>
</table>

★ there are also **two-way simulations**

$B$ simulates $A$ and $A$ simulates $B$, but two simulations are independent
Tests of existence

★ **Forward bisimulation**

\[
\begin{align*}
\mu^\top \cdot \delta^A_x &\leq \delta^B_x \cdot \mu^\top, & \mu \cdot \delta^B_x &\leq \delta^A_x \cdot \mu, & x \in X \\
\mu^\top \cdot \tau^A &\leq \tau^B, & \mu \cdot \tau^B &\leq \tau^A \\
\sigma^A &\leq \sigma^B \cdot \mu^\top, & \sigma^B &\leq \sigma^A \cdot \mu
\end{align*}
\]

★ first row – WLS with \( I_1 = I_4 = \emptyset, \ I = I_2 \cup I_3 \)

★ second row – equivalent to \( \mu \leq \gamma_0 = (\tau^B / \tau^A)^\top \land (\tau^A / \tau^B) \)

**Theorem (Test of existence for forward bisimulations)**

Let \( \gamma \) be the greatest solution of the above WLS such that \( \gamma \leq \gamma_0 \).

★ If \( \sigma^A \leq \sigma^B \cdot \gamma^\top \) and \( \sigma^B \leq \sigma^A \cdot \gamma \), then \( \gamma \) is the greatest forward bisimulation between \( \mathcal{A} \) and \( \mathcal{B} \).

★ If \( \gamma \) does not satisfy this condition, then there is no any forward bisimulation between \( \mathcal{A} \) and \( \mathcal{B} \).
Back to additively idempotent semirings

★ $S$ – additively idempotent semiring
★ $\preceq$ – natural partial ordering on $S$ and its extension to matrices
★ we can define all types of simulations and bisimulations for WFAs over $S$

★ Problem: How to test the existence and compute the greatest ones?

★ in the general case, there is no residuation for matrices over $S$
★ for $\alpha \in S^{A \times B}$ and $\gamma \in S^{A \times C}$ inequation $\alpha \cdot \mu \preceq \gamma$ may not have the greatest solution in $S^{B \times C}$ (similarly for $\mu \cdot \beta \preceq \gamma$)

★ Problem: Can this inequation have the greatest solution in some $M \subseteq S^{B \times C}$?
Relative residuals

- $\alpha \in S^{A \times B}$, $\beta \in S^{B \times C}$, $\gamma \in S^{A \times C}$
- $M \subseteq S^{B \times C}$, $N \subseteq S^{A \times B}$

- **relative right residual** of $\gamma$ by $\alpha$ w.r.t. $M$ – greatest solution of $\alpha \cdot \mu \leq \gamma$ in $M$, if it exists

- **relative left residual** of $\gamma$ by $\beta$ w.r.t. $N$ – greatest solution of $\mu \cdot \beta \leq \gamma$ in $N$, if it exists

- when $(M, +, 0)$ and $(N, +, 0)$ are finite submonoids of $(S^{B \times C}, +, 0)$ and $(S^{A \times B}, +, 0)$, relative residuals always exist

**Problem**: How to compute them?

- We solved the problem for $M = 2^{B \times C}$, $N = 2^{A \times B}$ – Boolean matrices
Boolean residuals

- **relative right residual of** $\gamma$ **by** $\alpha$ **w.r.t.** $2^{B \times C}$ **exists**
  - we call it the **Boolean right residual**, and denote it by $\alpha \wedge \gamma$
- **relative left residual of** $\gamma$ **by** $\beta$ **w.r.t.** $2^{A \times B}$ **exists**
  - we call it the **Boolean left residual**, and denote it by $\gamma \lceil \beta$
- for any assertion $\Psi$ of a classical Boolean logic, $[\Psi]$ denotes its **truth value**

**Theorem (Boolean residuals)**

$$(\alpha \wedge \gamma)(b, c) = [\alpha(\cdot, b) \leq \gamma(\cdot, c)], \quad (\gamma \lceil \beta)(a, b) = [\beta(b, \cdot) \leq \gamma(a, \cdot)]$$

For all $\xi \in 2^{B \times C}$ and $\eta \in 2^{A \times B}$ we have

$$\alpha \cdot \xi \leq \gamma \iff \xi \leq \alpha \wedge \gamma, \quad \eta \cdot \beta \leq \gamma \iff \eta \leq \gamma \lceil \beta$$
- **Boolean simulations and bisimulations**
  solutions of systems which define simulations and bisimulations in the class of **Boolean matrices**

- **Test of existence:** similar as for simulations and bisimulations for automata over a unital quantale

- **Difference:** we compute a sequence \( \{ \gamma_k \}_{k \in \mathbb{N}} \) of Boolean matrices

  - since \( 2^{A \times B} \) is finite, the sequence stabilizes at some \( \gamma_k \)

  - if \( \gamma_k \) passes the test, then there is a Boolean simulation (bisimulation) and \( \gamma_k \) is the greatest one

  - if \( \gamma_k \) does not pass the test, then there is no any Boolean simulation (bisimulation)
Past results and future work

★ **Past results:**
★ state reduction, simulation and bisimulation for

  - fuzzy automata over a complete residuated lattice
  - nondeterministic automata
  - weighted automata over an additively idempotent semiring
    - relative residuation – Boolean residuation

★ **Further work:**

  - weighted automata over a max-plus semiring
V

Weighted Networks

◊ Positional Analysis ◊
Social Network Analysis

★ mathematical methods for the study of social structures
★ they can also be applied to many other types of networks that arise in computer science, physics, biology, etc.

★ a social network is made up of
  – a set of social actors (individuals or organizations)
  – ties or social interactions between actors
★ most often, ties are modeled by Boolean-valued relations or Boolean matrices
★ in many real-world networks, not all ties have the same strength, intensity, duration, or some other quantitative property
★ in these cases, it is natural to assign weights to ties, to model these quantitative properties
★ $S$ – unital quantale

★ **Weighted network** (one-mode network): $\mathcal{N} = (A, \{q_i\}_{i \in I})$

$A$ – set of *actors*

$\{q_i\}_{i \in I} \subseteq S^{A \times A}$ – family of matrices

*matrices represent social relations between actors*

sometimes we consider $\mathcal{N} = (A, \{q_i\}_{i \in I}, \{\sigma_j\}_{j \in J})$

$\{\sigma_j\}_{j \in J} \subseteq S^A$ – family of vectors

*vectors represent certain individual properties of actors*

★ most often, *Boolean matrices* have been taken in account

★ another name: *valued networks* (usually integer weights)
Positional analysis

★ identify the **position** or **role** of actors in a network on the basis of mutual relationships

★ example: **terroristic group** or **criminal group** - identify roles (leaders, etc.) on the basis on communication between the group members (without insight into the content of the conversation)

Positional analysis (cont.)

★ closely related: blockmodeling – data reduction method
large and complex social networks are mapped into simpler structures – blockmodel images
blockmodel image – structural summary of the original network

★ common idea
to cluster actors who have substantially similar patterns of relationships with others
to interpret the pattern of relationships among the clusters

★ behavior of individuals is often determined by the affiliation of the group

★ such influence of the group to the behavior of an individual can be very important
★ **Structural equivalences** (Lorrain and White, 1971)

Two actors are considered to be structurally equivalent if they have identical neighborhoods in our terminology – greatest solutions of the system

\[
\mu \cdot q_i \leq q_i, \quad \mu^\top \cdot q_i \leq q_i, \quad q_i \cdot \mu \leq q_i, \quad q_i \cdot \mu^\top \leq q_i
\]

This concept has shown oneself to be too strong.

★ **Regular equivalences** (White and Reitz, 1983)

Less restrictive than structural equivalences and more appropriate for modeling social positions

Two actors are considered to be regularly equivalent if they are equally related to equivalent others.
Regular matrices

★ **Regular matrix** – solution of the system

\[ Q_i \cdot \mu = \mu \cdot Q_i, \quad Q_i \cdot \mu^\top = \mu^\top \cdot Q_i, \quad i \in I \]

weakly linear system

greatest solution – **regular equivalence matrix**

it can be computed using the previously described method

★ we can also use any of the following three systems

\[ Q_i \cdot \mu = \mu \cdot Q_i, \quad i \in I \]

\[ Q_i \cdot \mu \leq \mu \cdot Q_i, \quad i \in I \]

\[ \mu \cdot Q_i \leq Q_i \cdot \mu, \quad i \in I \]

the greatest solutions – **quasi-order matrices**

if we need the greatest solutions which are **equivalence matrices** – we have to add inequations with \( \mu^\top \)
★★ regular equivalence classes – \{1\}, \{2\}, \{3\}, \{4\}, \{5, 6\}

★★ simulation equivalence classes – \{1, 2\}, \{3\}, \{4\}, \{5, 6\}

Simulation equivalence ≡ natural equivalence of simulation quasi-order (its symmetric opening)

Simulation quasi-orders – (greatest) solutions of \( \mu \cdot \varphi_i \leq \varphi_i \cdot \mu \)

★★ regular equivalences can not identify group leaders – 1 and 2
Bisimulations in SNA

★ identify similar positions in different networks

★ networks: \( \mathcal{N} = (A, \{q_i\}_{i \in I}, \{\sigma_j\}_{j \in J}) \), \( \mathcal{N}' = (A', \{q'_i\}_{i \in I}, \{\sigma'_j\}_{j \in J}) \)

★ regular bisimulations – solutions of the system

\[
\sigma_j = \mu \cdot \sigma'_j, \quad j \in J
\]

\[
\sigma'_j = \mu^\top \cdot \sigma_j, \quad j \in J
\]

\[
q_i \cdot \mu = \mu \cdot q'_i, \quad i \in I
\]

\[
\mu^\top \cdot q_i = q'_i \cdot \mu^\top, \quad i \in I
\]

\( \mu \) – unknown taking values in \( S^{A \times B} \)

★ algorithm for testing the existence of a regular bisimulation and computing the greatest one

★ other types of simulations and bisimulations (unpublished)
Two-mode networks

★ **Two-mode network** – $\mathcal{T} = (A, B, \{q_i\}_{i \in I})$

$A, B$ – two sets of entities

$\{q_i\}_{i \in I} \subseteq S^{A \times B}$ – family of matrices (represent relationships)

**affiliation or bipartite networks**

★ **examples:** people attending events, organizations employing people, authors and articles, etc.

★ **Positional analysis** – identify positions in both modes of the network

★ **Indirect approach** – reduction to one-mode networks

single one-mode network on $A \cup B$

two one-mode networks $(A, \{q_i \cdot q_i^\top\}_{i \in I})$, $(B, \{q_i^\top \cdot q_i\}_{i \in I})$
Our direct approach

★ **Two-mode systems**

\[ \alpha \cdot q_i = q_i \cdot \beta, \quad i \in I \]
\[ \alpha \cdot q_i \leq q_i \cdot \beta, \quad i \in I \]
\[ q_i \cdot \beta \leq \alpha \cdot q_i, \quad i \in I \]

\[ \alpha \cdot q_i = q_i \cdot \beta, \quad \alpha^T \cdot q_i = q_i \cdot \beta^T, \quad i \in I \]
\[ \alpha \cdot q_i \leq q_i \cdot \alpha, \quad \alpha^T \cdot q_i \leq q_i \cdot \beta^T \quad i \in I \]
\[ q_i \cdot \beta \leq \alpha \cdot q_i, \quad q_i \cdot \beta^T \leq \alpha^T \cdot q_i \quad i \in I \]

\[ \{q_i\}_{i \in I} \subseteq S^{A \times B} \] – given family of matrices
\[ \alpha \text{ and } \beta \] – unknowns taking values in \( S^{A \times A} \) and \( S^{B \times B} \)

★ **solutions** – pairs of matrices (ordered coordinatewise)

★ **algorithms for computing the greatest solutions for all two-mode systems**
Multi-mode networks

★ **Multi-mode network** – $\mathcal{T} = (A_1, \ldots, A_n, \mathcal{R})$
★ $A_1, \ldots, A_n$ – multiple non-empty sets
★ $\mathcal{R}$ – system of $A_j \times A_k$-matrices defined for some pairs $(j, k)$
★ formally: $J \subseteq [1, n] \times [1, n]$ satisfying

$$(\forall j \in [1, n])(\exists k \in [1, n]) (j, k) \in J \text{ or } (k, j) \in J$$
★ $\{I_{j,k}\}_{(j,k)\in J}$ – collection of non-empty sets
★ $\mathcal{R} = \{\varrho^j_{i,k} \mid (j, k) \in J, i \in I_{j,k}\}$, $\varrho^j_{i,k} \in S^{A_j \times A_k}$, for all $(j, k) \in J, i \in I_{j,k}$
★ **complex synthesis of one-mode and two-mode networks**
★ **Positional analysis** – identify positions in all modes of the network
Multi-mode networks – examples

Simple organization network
Multi-mode networks – examples

Network of academic publications

- Citations
- Authorship
- Appearance
- Publishing
- Editing
- Words
- Journals, proceedings
- Researchers
- Papers
- Semantics
Multi-mode networks – examples

Genetic regulatory (interaction) network
Multi-mode systems

\[ \alpha_j \cdot \varrho_i^{j,k} = \varrho_i^{j,k} \cdot \alpha_k, \quad (j, k) \in J, \ i \in I_{j,k}, \]
\[ \alpha_j \cdot \varrho_i^{j,k} \leq \varrho_i^{j,k} \cdot \alpha_k, \quad (j, k) \in J, \ i \in I_{j,k}, \]
\[ \alpha_j \cdot \varrho_i^{j,k} \geq \varrho_i^{j,k} \cdot \alpha_k, \quad (j, k) \in J, \ i \in I_{j,k}, \]

★ \( \alpha_1, \ldots, \alpha_n – unknowns, \quad \alpha_j \) takes values in \( S^{A_j \times A_j} \)
★ solutions – \( n \)-tuples of matrices from \( S^{A_1 \times A_1} \times \cdots \times S^{A_n \times A_n} \)
★ \( n \)-tuples are ordered coordinatewise

★ greatest solutions – \( n \)-tuples of quasi-order matrices (to get equivalence matrices we add (in)equations with \( \alpha_j^T \) and \( \alpha_k^T \))
★ algorithms for computing the greatest solutions for all multi-mode systems
Example 1

**Grouping employees and jobs**

- two-mode network $\mathcal{T} = (A, B, \{\varrho_i\}_{i \in I})$
- $A$ – the set of all *employees* of some company
- $B$ – the set of all *jobs* which this company performs for other companies
- $I$ set of these other companies
- the jobs for the company $i \in I$ are allotted to employees by a relation $\varrho_i \subseteq A \times B$

**Task:** group employees into *teams* and jobs into *groups of jobs* so that

- the teams and groups of jobs are as wide as possible
- for any company, a group of jobs $\gamma$ is assigned to a team $\theta$ if and only if
  
  * for every employee from $\theta$ there is a job from $\gamma$ which he has already performed for that company
  * for every job from $\gamma$ there is an employee from $\theta$ who has already performed that job for that company

**this can be done using the greatest solution of** $\alpha \cdot \varrho_i = \varrho_i \cdot \beta$, $\alpha^\top \cdot \varrho_i = \varrho_i \cdot \beta^\top$
Example 2

The modified problem

★ the teams and the groups of jobs have *wider* and *narrower parts*
★ the *narrower parts* – the *cores* of the teams and groups of jobs
★ for any company, a group of jobs $\gamma$ is assigned to a team $\theta$ if and only if
  * for every employee *from the core* of $\theta$ there is a job from $\gamma$ which he has already performed for that company
  * for every job *from the core* of $\gamma$ there is an employee from $\theta$ who has already preformed that job for that company
★ this can be done using the greatest solution of $\alpha \cdot \varrho_i = \varrho_i \cdot \beta$
★ the wider teams and groups of jobs are the *rows* of $\alpha$ and the *columns* of $\beta$
★ the narrower teams and groups of jobs are equivalence classes of the natural equivalences of $\alpha$ and $\beta$
★ the core of the team performs the main part of the assigned jobs, and the rest of the team assists the core in the jobs that they have not previously performed and in other cases when they need help
★ the core of the group of jobs assigned to the team are main jobs they have to perform, while the rest of this group are those jobs for which the members of the team could be engaged to assist
Example 3

Adding a third mode

★ third mode: \textit{skills}, e.g., knowledge of specific \textit{software packages}, if the considered company is a software company

★ \textit{groups of software packages} assigned to teams

★ for any company, a group of software packages \( \pi \) is assigned to a team \( \theta \) if and only if

\begin{itemize}
  \item for every employee from \( \theta \) there is a software package from \( \pi \) for which the employee is qualified
  \item for every software package from \( \pi \) there is an employee from \( \theta \) who is qualified for that software package
\end{itemize}

★ such grouping can be done using the \textit{triples of equivalences} that are solutions of our three-mode systems

★ version with cores – \textit{triples of quasi-orders} that are solutions of our three-mode systems
Thank you for your attention!