# Relation Algebra and RelView Applied to Approval Voting

**Rudolf Berghammer, Nikita Danilenko and Henning Schnoor**

Institut für Informatik
Christian-Albrechts-Universität zu Kiel

24. April 2014

# Introduction

Voting procedures are used in situations if a group of individuals has to come to a common decision:

- Elections of political parliaments.
- Ballots in committees.
- Definition of winners in sports tournaments.
- Awarding of contracts.
- Granting of funds.
- ...
- What to do during the annual works outing?
- What language is used in the beginners lecture of Computer Science?
- ...

# Common Background: Voting Systems

- There is a finite and non-empty set $N$ of **voters** (agents, individuals, parties etc.). To simplify things one uses:

$$N = \{1, 2, \ldots, n\}$$

- There is a finite and non-empty set $A$ of **alternatives** (proposals, candidates etc.).
- Each voter $i$ possesses an **individual preference** $I_i$ in view of the given alternatives.
- There is a **voting rule** that specifies
  - how to aggregate the voter's individual preferences to a **collective preference**,
  - how then to get the **set of winners**.

Instances $(N, A, (I_i)_{i \in N})$ are called **elections**.

# Example: Approval Voting

- Here the individual preferences are sets of alternatives

$$A_i \in 2^A$$

and $a \in A_i$ is interpreted as "voter $i$ approves alternative $a$".

- The collective preference is specified via a **dominance relation**

$$D : A \leftrightarrow A,$$

such that for all $a, b \in A$ it holds

$$D_{a,b} \iff |\{i \in N \mid a \in A_i\}| \geq |\{i \in N \mid b \in A_i\}|.$$

- There always exist alternatives which dominate all alternatives; these are called the **approval winners**.

**Weak dominance** and **multiple-winners** condition.

# Example: Condorcet Voting

- Here the individual preferences are linear strict-order relations

$$>_i \,:\, A \leftrightarrow A$$

  and $a >_i b$ is interpreted as "voter $i$ ranks alternative $a$ better than $b$".
- The collective preference is specified via a **dominance relation**

$$D : A \leftrightarrow A,$$

  such that for all $a, b \in A$ it holds

$$D_{a,b} \iff |\{i \in N \mid a >_i b\}| \geq |\{i \in N \mid b >_i a\}|.$$

- There is not always an alternative that dominates all alternatives; if such an alternative exists it is called the **Condorcet winner**.
- If there is no Condorcet winner, then the winners are specified via so-called **choice sets** (top cycle, uncovered set, Banks set etc.).

# Example: Borda Voting

- Here the individual preferences are injective functions

$$f_i : A \to \{0, 1, \ldots, |A| - 1\}$$

  and the value $f_i(a)$ is interpreted as "voter $i$ assigns $f_i(a)$ points to alternative $a$".
- The collective preference is specified via a **dominance relation**

$$D : A \leftrightarrow A,$$

  such that for all $a, b \in A$ it holds

$$D_{a,b} \iff \sum_{i \in N} f_i(a) \geq \sum_{i \in N} f_i(b).$$

- There always exist alternatives which dominate all alternatives; these are called the **Borda winners**.

# Control of Elections

Here it is assumed that the authority conducting the election, called the **chair**, knows the individual preferences of the voters and is able

- **to remove voters from the election** (by dirty tricks, like mistimed meetings)
- **to remove alternatives from the election** (by excuses, like "too expensive" or "legally not allowed").

Using **constructive control**, the chair tries

- to make a specific alternative $a^* \in A$ to a winner by a removal of voters / of alternatives
- and (to hide his mind) to remove as few as possible voters / alternatives to reach this goal.

Using **destructive control**, with the same actions the chair tries to prevent $a^*$ from winning.

Control may be hard or easy. E.g., in case of **approval voting** we have:

- Constructive control via the removal of voters is NP-hard.
- There are efficient algorithms for the constructive control via the removal of alternatives.

In case of **plurality voting** (another well-known voting system) the complexities change, i.e.:

- Constructive control via the removal of alternatives is NP- hard.
- There are efficient algorithms for the constructive control via the removal of voters.

**Our goal:** Use of relation algebra and the BDD-based tool RelView

- for computing dominance relations and winners,
- for the solution of non-trivial instances of hard control problems.

Here: **Approval voting** and **constructive control** by a removal of voters.

# Specific Relational Constructions

- The **symmetric quotient** of $R : X \leftrightarrow Y$ and $S : X \leftrightarrow Z$ is defined as $syq(R,S) = \overline{R^{\mathsf{T}};\overline{S}} \cap \overline{\overline{R}^{\mathsf{T}};S} : Y \leftrightarrow Z$ and from this we get:

$$syq(R,S)_{y,z} \iff \forall x \in X : R_{x,y} \leftrightarrow S_{x,z}$$

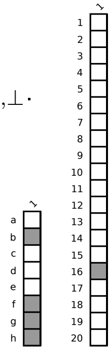
- If the target of a relation is a singleton set, here always $\mathbf{1} = \{\bot\}$, it is called a **vector**.

  We denote vectors by small letters and write $v_x$ instead of $v_{x,\bot}$.

  A vector $v : X \leftrightarrow \mathbf{1}$ **describes** the subset $\{x \in X \mid v_x\}$ of its source.

- A **point** $p : X \leftrightarrow \mathbf{1}$ is a vector which describes a singleton subset $\{x\}$ of $X$.

  We then say that it **describes** the element $x$ of $X$.



- The **membership relation** $\mathsf{M} : X \leftrightarrow 2^X$ is defined as follows:

$$\mathsf{M}_{x,Y} \iff x \in Y$$

- The **size-comparison relation** $\mathsf{S} : 2^X \leftrightarrow 2^X$ is defined as follows:

$$\mathsf{S}_{Y,Z} \iff |Y| \leq |Z|$$

- The **projection relations** $\pi : X \times Y \leftrightarrow X$ and $\rho : X \times Y \leftrightarrow Y$ are defined as follows:

$$\pi_{(x,y),z} \iff x = z \qquad\qquad \rho_{(x,y),z} \iff y = z$$

- The **pairing** (or fork) of $R : Z \leftrightarrow X$ and $S : Z \leftrightarrow Y$ is defined as the relation $[R, S]\!] = R; \pi^\mathsf{T} \cap S; \rho^\mathsf{T} : Z \leftrightarrow X \times Y$ and from this we get:

$$[R, S]\!]_{z,(x,y)} \iff R_{z,x} \wedge S_{z,y}$$

All that is available in the programming language of RelView.

# A Relational Model of Approval Voting

- A relation $P : N \leftrightarrow A$ is called a **relational model** of $(N, A, (A_i)_{i \in N})$ if for all $i \in N$ and $a \in A$

$$P_{i,a} \iff a \in A_i.$$

- If $P : N \leftrightarrow A$ is a relational model, then we get

$$\begin{aligned} \{i \in N \mid P_{i,c}\} = Z &\iff \forall\, i \in N : P_{i,c} \leftrightarrow i \in Z \\ &\iff \forall\, i \in N : P_{i,c} \leftrightarrow \mathsf{M}_{i,Z} \\ &\iff syq(P, \mathsf{M})_{c,Z} \end{aligned}$$

for all $c \in A$ and $Z \in 2^N$ and this shows for the **dominance relation**

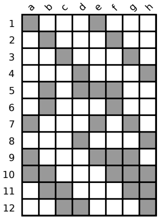
$$D = syq(P, \mathsf{M}); \mathsf{S}^{\mathsf{T}}; syq(P, \mathsf{M})^{\mathsf{T}} : A \leftrightarrow A.$$

- The **set of winners** is described by the vector

$$win = \overline{\overline{D}; \mathsf{L}} : A \leftrightarrow \mathbf{1}.$$

# An Example

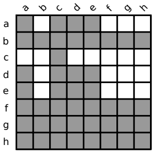
- Relational model $P : N \leftrightarrow A$ as RelView-matrix:

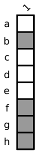



Voters $N = \{1, 2, \ldots, 12\}$

Alternatives $A = \{a, b, \ldots, h\}$

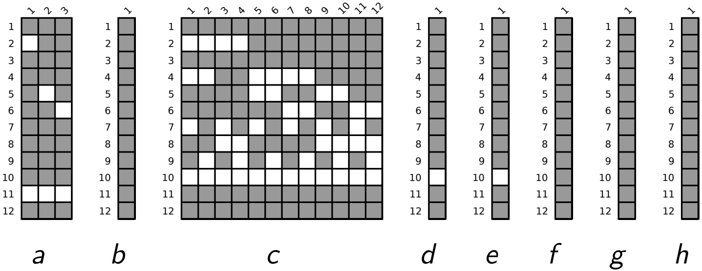
- Dominance relation $D : A \leftrightarrow A$ and vector $win : A \leftrightarrow \mathbf{1}$ as computed by RelView:




How many voters need to be removed such that, e.g., alternative $e$ wins?

The answer to the last question for all alternatives $a, b, \ldots, h$ as computed and shown in a column-wise fashion by RelView (in the same order):




- Positions 2, 6, 7 and 8: No voter needs to be removed to ensure win for $b$, $f$, $g$ and $h$.
- Position 4 and 5: Voter 10 needs to be removed to ensure win for $d$ and $e$.
- Position 1: Two voters need to be removed to ensure win for $a$, viz. $2, 11$ or $5, 11$ or $6, 11$.
- Position 3: Four voters need to be removed to ensure win for $c$ and there are 12 possibilities for this.

# Relational Control of Approval Voting

We assume that $P : N \leftrightarrow A$ is a model of $(N, A, (A_i)_{i \in N})$ and $a^* \in A$ shall win, where the point $p : N \leftrightarrow \mathbf{1}$ describes $a^*$. Our solution of the control problem consists of three steps:

- Formulation as maximization-problem: Compute a maximum $X \in 2^N$ such that $a^*$ wins in the restricted election $(X, A, (A_i)_{i \in X})$. Then all alternatives from $N \setminus X$ are to remove.
- Relation-algebraic specification of the **vector of candidetes sets**

$$cand : 2^N \leftrightarrow \mathbf{1}$$

  such that $cand_X$ iff $a^*$ wins in $(X, A, (A_i)_{i \in X})$.
- Relation-algebraic specification of the **vector of solutions**

$$sol = cand \cap \overline{\overline{\mathsf{S}}^{\mathsf{T}} ; cand} : 2^N \leftrightarrow \mathbf{1}$$

  that describes the maximum sets in the set of sets described by *cand*.

# Specification of the Vector of Candidates Sets

Let an arbitrary set $X \in 2^N$ be given. Since

$$(P;p)_i \iff \exists\, a \in A : P_{i,a} \wedge p_a \iff \exists\, a \in A : P_{i,a} \wedge a = a^* \iff P_{i,a^*}$$

for all $i \in N$, we get for all $Y \in 2^N$ that

$$\begin{array}{lll}
 & \{i \in X \mid a^* \in A_i\} = Y & \\
\iff & \{i \in X \mid P_{i,a^*}\} = Y & P \text{ model} \\
\iff & \forall\, i \in N : (i \in X \wedge P_{i,a^*}) \leftrightarrow i \in Y & \\
\iff & \forall\, i \in N : (i \in X \wedge (P;p)_i) \leftrightarrow i \in Y & \text{see above} \\
\iff & \forall\, i \in N : (\mathsf{M}_{i,X} \wedge (P;p;\mathsf{L})_{i,X}) \leftrightarrow \mathsf{M}_{i,Y} & \text{definition } \mathsf{M} \\
\iff & \forall\, i \in N : (\mathsf{M} \cap P;p;\mathsf{L})_{i,X} \leftrightarrow \mathsf{M}_{i,Y} & \\
\iff & \underbrace{syq(\mathsf{M} \cap P;p;\mathsf{L}, \mathsf{M})}_{E}{}_{X,Y} & \text{property } syq
\end{array}$$

... and for all $Z \in 2^N$ and $b \in A$ that

$$
\begin{array}{lll}
& Z = \{i \in X \mid b \in A_i\} & \\
\iff & Z = \{i \in X \mid P_{i,b}\} & P \text{ model} \\
\iff & \forall\, i \in N : i \in Z \leftrightarrow (i \in X \wedge P_{i,b}) & \\
\iff & \forall\, i \in N : \mathsf{M}_{i,Z} \leftrightarrow (\mathsf{M}_{i,X} \wedge P_{i,b}) & \text{definition } \mathsf{M} \\
\iff & \forall\, i \in N : \mathsf{M}_{i,Z} \leftrightarrow [\![\mathsf{M}, P]\!]_{i,(X,b)} & \text{property tupling} \\
\iff & \underbrace{\mathit{syq}(\mathsf{M}, [\![\mathsf{M}, P]\!])}_{F}{}_{Z,(X,b)} & \text{property } \mathit{syq}
\end{array}
$$

yielding the relations

$$E = \mathit{syq}(\mathsf{M} \cap P; p; \mathsf{L}, \mathsf{M}) : 2^N \leftrightarrow 2^N,$$

where $\mathsf{L} : \mathbf{1} \leftrightarrow 2^N$, and

$$F = \mathit{syq}(\mathsf{M}, [\![\mathsf{M}, P]\!]) : 2^N \leftrightarrow 2^N \times A$$

. . . and

$$
\begin{aligned}
& a^* \text{ wins in } \; (X, A, (A_i)_{i \in X}) \\
\iff \quad & \forall\, b \in A : |\{i \in X \mid a^* \in A_i\}| \geq |\{i \in X \mid b \in A_i\}| \\
\iff \quad & \neg\exists\, b \in A : |\{i \in X \mid a^* \in A_i\}| < |\{i \in X \mid b \in A_i\}| \\
\iff \quad & \neg\exists\, b \in A : (E; \overline{\mathsf{S}}^{\mathsf{T}}; F)_{X,(X,b)} \\
\iff \quad & \neg\exists\, U \in 2^N, b \in A : (E; \overline{\mathsf{S}}^{\mathsf{T}}; F)_{X,(U,b)} \wedge U = X \\
\iff \quad & \neg\exists\, U \in 2^N, b \in A : (E; \overline{\mathsf{S}}^{\mathsf{T}}; F)_{X,(U,b)} \wedge \pi_{(U,b),X} \\
\iff \quad & \neg\exists\, U \in 2^N, b \in A : (E; \overline{\mathsf{S}}^{\mathsf{T}}; F \cap \pi^{\mathsf{T}})_{X,(U,b)} \wedge \mathsf{L}_{(U,b)} \\
\iff \quad & \underbrace{\overline{(E; \overline{\mathsf{S}}^{\mathsf{T}}; F \cap \pi^{\mathsf{T}}); \mathsf{L}}}_{cand}{}_X
\end{aligned}
$$

yielding the vector

$$
cand = \overline{(E; \cap \overline{\mathsf{S}}^{\mathsf{T}}; F \cap \pi^{\mathsf{T}}); \mathsf{L}} \; : 2^N \leftrightarrow \mathbf{1},
$$

where $\mathsf{L} : 2^N \times A \leftrightarrow \mathbf{1}$.

# Concluding Remarks

Present and future work:

- Investigation of further voting systems.
  - Condorcet voting (AAMAS 2014, May 2014).
  - Plurality voting (CASC 2014, submitted).
  - ...
- Investigation of further types of manipulation.
  - Control by partition.
  - Bribery.
  - ...
- Investigation of further methods of solutions.
  - Functional programming.
  - Constraint programming.
  - Binary integer programming.
  - Bio-inspired techniques.
  - Heuristics
  - ...