halshs-00587690, version 1 - 21 Apr 2011

Documents de Travail du
Centre d'Economie de la Sorbonne

CES Working Papers

Computations on Simple Games using REL VIEW

Rudolf BERGHAMMER, Agnieszka RUSINOWSKA, Harrie De SWART

2011.14

Maison des Sciences Économiques, 106-112 boulevard de L'Hôpital, 75647 Paris Cedex 13
http://centredeconomiesorbonne.univ-paris1.fr/bandeau-haut/documents-de-travail/
ISSN : 1955-611X

halshs-00587690, version 1 - 21 Apr 2011

Computations on Simple Games using RelView

Rudolf Berghammer[1], Agnieszka Rusinowska[2], and Harrie de Swart[3]

[1] Institut für Informatik, Universität Kiel, Olshausenstraße 40, 24098 Kiel, Germany
rub@informatik.uni-kiel.de
[2] Centre d'Economie de la Sorbonne, CNRS – Université Paris 1
106-112 Bd de l'Hôpital, 75647 Paris, France
agnieszka.rusinowska@univ-paris1.fr
[3] Department of Philosophy, Erasmus University Rotterdam
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
deSwart@fwb.eur.nl

Abstract. Simple games are a powerful tool to analyze decision-making and coalition formation in social and political life. In this paper we present relational models of simple games and develop relational algorithms for solving some game-theoretic basic problems. The algorithms immediately can be transformed into the language of the Computer Algebra system RelView and, therefore, the system can be used to solve the problems and to visualize the results of the computations. As an example, we consider the German parliament after the 2009 election.

Résumé. Les jeux simples constituent un outil d'analyse puissant pour la décision et la formation de coalitions dans la vie sociale et politique. Dans cet article nous présentons les modèles relationnels des jeux simples et développons des algorithmes relationnels pour résoudre certains problèmes de base en théorie des jeux. Les algorithmes peuvent être transformés immédiatement dans le langage RelView d'algèbre computationnelle, et de ce fait, le système peut être utilisé pour résoudre les problèmes et visualiser les résultats des calculs. Nous étudions en tant qu'exemple le parlement allemand après l'élection de 2009.

JEL Classification: C71, C63, C88

Keywords: simple games, relation algebra, RelView

Mots-clés : jeux simples, algèbre relationnelle, RelView

1 Introduction

In game theory (starting with [12]) a distinction is made between non-cooperative games and cooperative games. In non-cooperative games each player (agent, party etc.) must decide individually, while in cooperative games players are allowed to act jointly and to decide within a group what strategy will be followed.

Cooperative games (see e.g. [13, 14] for an introduction) are a very useful tool for modeling the cooperation of players and for measuring the outcome

halshs-00587690, version 1 - 21 Apr 2011

caused by this. Simple games are a special class of cooperative games. Here the numerical payoff of a coalition is either 1 or 0. Hence, only two classes of coalitions are possible. The winning ones (payoff 1) take all and the losing ones (payoff 0) receive nothing. This kind of games is very important, e.g., in social choice theory, for the comparison and measurement of influence and power of agents in decision-making processes and for the analysis of social and political situations. In respect of the latter application domain we refer to [7, 8, 17], for example.

As demonstrated e.g., in [9, 15], a lot of important problems on simple games are known to be intractable in terms of complexity theory. In the recent years we successfully have combined relation algebra (cf. [16]) and the BDD-based specific purpose Computer Algebra system RelView (cf. [3, 4]) for the formation of coalitions and alliances and to measure the strength of agents in social and political networks. See [5, 6] for details. In [2] this approach is extended to the solution of some standard problems on simple games like the detection of some key players, the test of some fundamental properties of simple games, and the computation of some power indices. All relation-algebraic solutions of [2] and, hence, also the corresponding RelView-programs, base on the so-called relational vector-model of simple games. An alternative model, called relational membership-model, is only used to facilitate the input of the RelView-programs and to visualize their computed results.

The present paper is a continuation of [2]. Besides the vector-model and the membership-model we study a third relational model of simple games, viz. the seat-distribution-model of the important sub-class of voting games. We also show how some standard problems on simple games can be solved using the membership-model instead of the vector-model. There are situations where the membership-model is more appropriate than the vector-model. Because RelView has a very efficient BDD implementation of relations, the tool is able to deal with non-trivial simple games that, for instance, appear in practical political life. In addition, the tool has visualization facilities which are not easily found in other software tools and which are most helpful for fully comprehending difficult concepts and for understanding and testing the programs. We will demonstrate the visualization of results by an example from the present practical political life, viz. the game that models the German parliament after the 2009 election.

2 Relational Preliminaries

We denote the set (type) of all relations with source X and target Y (i.e., the powerset $2^{X \times Y}$) by $[X \leftrightarrow Y]$ and write $R : X \leftrightarrow Y$ instead of $R \in [X \leftrightarrow Y]$. If X and Y are finite sets, then we may consider R also as a Boolean matrix. This interpretation is well suited for many purposes and Boolean matrices are also used as one of the graphical representations of relations within RelView. Therefore, in this paper we often use Boolean matrix terminology and notation. In particular, we speak of rows, columns and components/entries of relations and write $R_{x,y}$ instead of $\langle x, y \rangle \in R$ or $x\,R\,y$. We will employ the following basic

operations on relations: $\overline{R}$ (*complement*), $R \cup S$ (*union*), $R \cap S$ (*intersection*), R^T (*transposition*) and RS (*composition*). Furthermore, we will use the special relations O (*empty relation*), L (*universal relation*) and I (*identity relation*). Here we overload the symbols, i.e., avoid the binding of types to them.

By $syq(R,S) = \overline{R^\mathsf{T}\,\overline{S}} \cap \overline{\overline{R}^\mathsf{T} S}$ the *symmetric quotient* of $R : X \leftrightarrow Y$ and $S : X \leftrightarrow Z$ is defined. The type of $syq(R,S)$ is $[Y \leftrightarrow Z]$, and transforming its definition into a component-wise notation, we have for all $y \in Y$ and $z \in Z$ that $syq(R,S)_{y,z}$ iff for all $x \in X$ it holds $R_{x,y}$ iff $S_{x,z}$.

A *vector* is a relation v with the specific set $\mathbf{1} := \{\bot\}$ as target. Since in $v_{x,\bot}$ the second index $\bot$ is irrelevant, we write in the following v_x instead of $v_{x,\bot}$. Vectors correspond to Boolean column vectors. We say that $v : X \leftrightarrow \mathbf{1}$ *describes* the subset Y of X if for all $x \in X$ we have $x \in Y$ iff v_x. In such a case $inj(v) : Y \leftrightarrow X$ denotes the *embedding-relation* of Y into X. This means that for all $y \in Y$ and $x \in X$ we have $inj(v)_{y,x}$ iff $y = x$. To model sets we also will use the relation-level equivalents of the set-theoretic symbol "$\in$", i.e., *membership-relations* $\mathsf{E} : X \leftrightarrow 2^X$ defined by $\mathsf{E}_{x,Y}$ iff $x \in Y$, for all $x \in X$ and $Y \in 2^X$. A combination of embedding-relations and membership-relations allows a *column-wise enumeration* of a subset of a powerset. If $v : 2^X \leftrightarrow \mathbf{1}$ describes a subset $\mathfrak{S}$ of 2^X in the sense defined above, then for all $x \in X$ and $Y \in \mathfrak{S}$ we have $(\mathsf{E}\,inj(v)^\mathsf{T})_{x,Y}$ iff $x \in Y$. Using Boolean matrix terminology this means that the elements of $\mathfrak{S}$ are described precisely by the columns of $\mathsf{E}\,inj(v)^\mathsf{T} : X \leftrightarrow \mathfrak{S}$.

A non-empty vector $v : X \leftrightarrow \mathbf{1}$ is a *point* if $vv^\mathsf{T} \subseteq \mathsf{I}$. This means that it describes a singleton subset of X or an element from X if we identify a singleton set with the only element it contains. In the Boolean matrix model, hence, a point $p : X \leftrightarrow \mathbf{1}$ is a Boolean column vector in which exactly one entry (component) is 1. If it describes $x \in X$, then for all $y \in X$ it holds p_y iff $x = y$.

For a direct product $X \times Y$ there are the projections which decompose a pair $u = \langle u_1, u_2 \rangle$ into its first component u_1 and its second component u_2. Within relation algebra it is very useful to consider instead of projections the corresponding *projection relations* $\pi : X \times Y \leftrightarrow X$ and $\rho : X \times Y \leftrightarrow Y$ such that, given any $u \in X \times Y$, it holds $\pi_{u,x}$ iff $u_1 = x$ and $\rho_{u,y}$ iff $u_2 = y$. Projection relations enable us to describe the well-known pairing operation of functional programming relation-algebraically as follows: For given relations $R : Z \leftrightarrow X$ and $S : Z \leftrightarrow Y$ we define their *pairing* (frequently also called *fork* or *tupling*) $[R,S] : Z \leftrightarrow X \times Y$ by $[R,S] := R\pi^\mathsf{T} \cap S\rho^\mathsf{T}$. Then for all $z \in Z$ and pairs $u \in X \times Y$ a simple reflection shows that $[R,S]_{z,u}$ iff R_{z,u_1} and S_{z,u_2}.

We also will employ a function *rel* (in the usual mathematical sense) which establishes a Boolean lattice isomorphism between the types $[X \times Y \leftrightarrow \mathbf{1}]$ and $[X \leftrightarrow Y]$. It is defined by $rel(v) = \pi^\mathsf{T}(\rho \cap v\mathsf{L}^\mathsf{T})$ for all vectors $v : X \times Y \leftrightarrow \mathbf{1}$, where $\pi : X \times Y \leftrightarrow X$ and $\rho : X \times Y \leftrightarrow Y$ are the projection relations of $X \times Y$ and L is a universal vector of type $[Y \leftrightarrow \mathbf{1}]$. Using a component-wise notation, the definition says that for all $x \in X$ and $y \in Y$ we have $v_{\langle x,y \rangle}$ iff $rel(v)_{x,y}$.

halshs-00587690, version 1 - 21 Apr 2011

halshs-00587690, version 1 - 21 Apr 2011

3 The Computer Algebra System RelView

RelView (see [4, 11]) is a specific purpose Computer Algebra system for the visualization and manipulation of relations. In it all data are represented as relations, which the tool visualizes in different ways. It offers several algorithms for pretty-printing a relation for which source and target coincide as a directed graph. Alternatively, an arbitrary relation may be displayed as a Boolean matrix which is very useful for visual editing and also for discovering structural properties that are not evident from a graphical presentation. Because RelView often works on (very) large data, it uses a very efficient implementation of relations based on reduced ordered binary decision diagrams (see [10, 11]). E.g., a membership-relation $\mathsf{E} : X \leftrightarrow 2^X$ requires $\mathcal{O}(|X|)$ BDD-vertices only. Besides it, we will also use a vector $\mathit{cardfilter}(Q) : 2^X \leftrightarrow \mathbf{1}$ that describes the subset $\{Y \in 2^X \mid |Y| < Q\}$ of 2^X. Its BDD-implementation requires $\mathcal{O}(|X|^2)$ vertices.

The main purpose of RelView is the evaluation of relation-algebraic expressions. These are constructed from the relations of its workspace using pre-defined operations and tests and user-defined functions and programs. A RelView-program is much like a function procedure in Modula 2, except that it only uses relations as data type. It starts with a head line containing the program name and the formal parameters. Then the declaration of the local relational domains, functions and variables follows. Declarations of product domains allow to introduce projection relations and pairings. The main part of a program is the body, a while-program over relations. As a program computes a value, it contains a return-clause, which is a relation-algebraic expression whose value after the execution of the body is the result. For instance, the RelView-programs corresponding to the definition of $\mathit{rel}(v)$ in Sect. 2 looks as follows:

```
rel(v,S)
  DECL XY = PROD(S*S^,S^*S);
       pi, rho, L
  BEG  pi = p-1(XY); rho = p-2(XY); L = L1n(rho)^
       RETURN pi^*(rho & v*L^)
  END.
```

In this program the declarations introduce `XY` as name for the direct product $X \times Y$ and three variables `pi`, `rho` and `L`. Since polymorphism is not part of the present version of RelView, the type $[X \leftrightarrow Y]$ is made available via a second argument S of this type. Using the product domain `XY`, in the body the projection relations $\pi : X \times Y \leftrightarrow X$ and $\rho : X \times Y \leftrightarrow Y$ and the universal vector $\mathsf{L} : Y \leftrightarrow \mathbf{1}$ are computed and stored in `pi`, `rho` and `L`, respectively. The return-clause is a direct translation of the definition of $\mathit{rel}(v)$ into RelView-code.

4 Relational Models of Simple Games

A *cooperative game* is a pair (N, f), where $N = \{1, \ldots, n\}$ is the set of *players* and $f : 2^N \to \mathbb{R}$ is the game's *characteristic function*. A subset C of N is called a *coalition* and $f(C)$ represents its payoff. The game (N, f) is *simple* if

$f(C) \in \{0,1\}$ for all $C \in 2^N$. In this case, a coalition C with $f(C) = 1$ is *winning* and one with $f(C) = 0$ is *losing*. A function from 2^N into $\{0,1\}$ can be seen as a vector of type $[2^N \leftrightarrow \mathbf{1}]$ such that the function maps $C \in 2^N$ to 1 iff the entry of the vector in the row corresponding to C is 1. Hence, the above definition immediately leads to a first relational model of simple games.

Definition 4.1 *Given a simple game (N, f), a vector $v : 2^N \leftrightarrow \mathbf{1}$ is called its* relational vector-model *if for all $C \in 2^N$ it holds $f(C) = 1$ iff v_C.*

The vector-model $v : 2^N \leftrightarrow \mathbf{1}$ of a simple game (N, f) describes the set $\mathcal{W}$ of the game's winning coalitions as subset of 2^N in the sense of Sect. 2. Characteristic functions are not the only possibility do define simple games. Another natural way to introduce them is to use pairs $(N, \mathcal{W})$ with $N = \{1, \ldots, n\}$ again as set of players and $\mathcal{W}$ as subset of 2^N that specifies the set of winning coalitions. If we enumerate the set $\mathcal{W}$ via the columns of a relation as described in Sect. 2, we obtain another relational model of simple games.

Definition 4.2 *Let (N, f) be a simple game and $\mathcal{W}$ denote the set of its winning coalitions. Then $M : N \leftrightarrow \mathcal{W}$ is called the game's* relational membership-model *if for all $k \in N$ and $C \in \mathcal{W}$ it holds $M_{k,C}$ iff $k \in \mathcal{W}$.*

Since the columns of the membership-model enumerate the set of winning coalitions, with regard to the use of RelView this model is in particular appropriate for input and output purposes. Which coalitions are winning can hardly be seen from the vector-model. Concerning the efficiency of algorithms. experiments with RelView have shown that in the case of a high percentage of winning coalitions typically the vector-model is superior and for games with smaller sets of winning coalitions prevalently the membership-model wins.

From Sect. 2 we already know how to get the membership-model from the vector-model. This transformation is described again in the "⇒"-part of the following theorem. In its "⇐"-part it is shown how to obtain the vector-model back from the membership-model. For a proof of this implication, see [2].

Theorem 4.1 *Let (N, f) be a simple game with set $\mathcal{W}$ of winning coalitions and $\mathsf{E} : N \leftrightarrow 2^N$ be a membership-relation. If $v : 2^N \leftrightarrow \mathbf{1}$ is the game's vector-model, then its membership-model is $M := \mathsf{E}\, inj(v)^{\mathsf{T}} : N \leftrightarrow \mathcal{W}$, and if $M : N \leftrightarrow \mathcal{W}$ is the game's membership-model, then its vector-model is $v := syq(\mathsf{E}, M)\mathsf{L} : 2^N \leftrightarrow \mathbf{1}$.*

Assume (N, f) to be a simple game and let $w_1, \ldots, w_n, Q$ be natural numbers. In this context, Q is called the *quota* and w_k is called the *weight* of player k. Then the linear list $[Q; w_1, \ldots, w_n]$ constitutes a *weighted realization* of the game if for all coalitions $C \in 2^N$ it holds that C is winning iff $\sum_{k \in C} w_k \geq Q$. A simple game is called a *weighted voting game* or a *weighted majority game* if it has a weighted realization. This type of simple games plays a prominent role if game theory is used to model and analyze real political situations. See [7, 8], for example.

To obtain a specification of weighted voting games within relation algebra, the players are interpreted as the parties of a parliament and the weights as the

halshs-00587690, version 1 - 21 Apr 2011

number of the parliament seats the parties hold, i.e., in the very same way as in real political life. This leads to the following seat-distribution-model.

Definition 4.3 *If (N, f) is a weighted voting game with the weighted realization $[Q; w_1, \ldots, w_n]$, a relation $D : S \leftrightarrow N$ models the game's* seat-distribution *if it is a mapping (in terms of relation algebra this may be specified by $D^\mathsf{T} D \subseteq \mathsf{I}$ and $D\mathsf{L} = \mathsf{L}$; cf. [16]), and for all $k \in N$ it holds $w_k = |\{s \in S \mid D_{s,k}\}|$.*

In the concrete case of real political parties and parliaments, for all $s \in S$ and $k \in N$ the relationship $D_{s,k}$ (or $D(s) = k$ in conventional notation) is interpreted as "seat s is owned by party k". I.e., the weight of a party equals the number of its seats. In Theorem 4.2 we show how to obtain from the seat-distribution-model the vector-model and, hence, via Theorem 4.1 also the membership model.

Theorem 4.2 *Assume (N, f) to be a weighted voting game with the weighted realization $[Q; w_1, \ldots, w_n]$ and let the mapping $D : S \leftrightarrow N$ model its seat-distribution. If $\mathsf{E} : N \leftrightarrow 2^N$ and $\mathsf{E}' : S \leftrightarrow 2^S$ are membership-relations, then the vector-model of the game is $v := syq(D\mathsf{E}, \mathsf{E}')\, \overline{cardfilter(Q)} : 2^N \leftrightarrow \mathbf{1}$.*

Proof. Let c abbreviate $cardfilter(Q)$. From the component-wise descriptions of c and of symmetric quotients (see Sect. 3 and 2) we get for all $C \in 2^N$ that

$$
\begin{aligned}
v_C &\Leftrightarrow (syq(D\mathsf{E}, \mathsf{E}')\, \overline{c}\,)_C \\
&\Leftrightarrow \exists\, X \in 2^S : syq(D\mathsf{E}, \mathsf{E}')_{C,X} \wedge \overline{c}_{\,X} \\
&\Leftrightarrow \exists\, X \in 2^S : syq(D\mathsf{E}, \mathsf{E}')_{C,X} \wedge |X| \geq Q \\
&\Leftrightarrow \exists\, X \in 2^S : (\forall\, s \in S : (D\mathsf{E})_{s,C} \leftrightarrow \mathsf{E}'_{s,X}) \wedge |X| \geq Q \\
&\Leftrightarrow \exists\, X \in 2^S : (\forall\, s \in S : (\exists\, k \in N : D_{s,k} \wedge \mathsf{E}_{k,C}) \leftrightarrow \mathsf{E}'_{s,X}) \wedge |X| \geq Q \\
&\Leftrightarrow \exists\, X \in 2^S : (\forall\, s \in S : (\exists\, k \in N : D_{s,k} \wedge k \in C) \leftrightarrow s \in X) \wedge |X| \geq Q \\
&\Leftrightarrow \exists\, X \in 2^S : X = \{s \in S \mid \exists\, k \in C : D_{s,k}\} \wedge |X| \geq Q \\
&\Leftrightarrow \exists\, X \in 2^S : X = \textstyle\bigcup_{k \in C} \{s \in S \mid D_{s,k}\} \wedge |X| \geq Q \\
&\Leftrightarrow |\textstyle\bigcup_{k \in C} \{s \in S \mid D_{s,k}\}| \geq Q \\
&\Leftrightarrow \textstyle\sum_{k \in C} |\{s \in S \mid D_{s,k}\}| \geq Q \\
&\Leftrightarrow \textstyle\sum_{k \in C} w_k \geq Q.
\end{aligned}
$$

Hence, the vector v describes the set of all winning coalitions. □

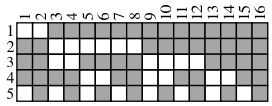
If the weights are small, then it is easy to obtain the seat-distribution relation D of a weighted voting game using RelView's facilities for interactively constructing relations on the system's screen using command buttons and the mouse. In the case of larger weights such a procedure may become cumbersome. Here it is advantageous to employ the ASCII file-format of RelView in order to load a relation W into the system that consists of the pairs $(1, w_1), \ldots, (k, w_k)$ and then to apply a simple RelView-program that transforms W into D.

In practical political live the number of winning coalitions of a simple game that models a certain situation can grow rapidly with the number of players. Therefore, in the following example we deal with a rather small game.

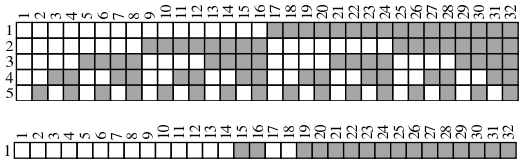
halshs-00587690, version 1 - 21 Apr 2011

Example 4.1. We consider a weighted voting game with five players, that models the parliament of Germany (the German Bundestag) after the September 2009 election. Its weighted realization is $[312; 239, 146, 93, 76, 68]$, with 312 as quota (for absolute majority; the number of seats of the present German parliament is 622) and then the numbers of seats of the five parties. These are, from left to right, labeled with 1, 2, 3, 4 and 5 and correspond (in the same order) to the parties CDU/CSU, SPD, FDP, Die Linke and Bündnis 90 / Die Grünen. All data are taken from the official web site `www.bundeswahlleiter.de`.

Depicted by RelView, the membership-model $M : N \leftrightarrow \mathcal{W}$ of this game looks as follows; in this Boolean 5×16 matrix a black square means a 1-entry and a white square means a 0-entry. The row labels of M denote the players.



Column 5 of this matrix represents the coalition that forms the present German government. If we transform M into the vector-model vector $v : 2^N \leftrightarrow \mathbf{1}$, we obtain in RelView a Boolean 32×1 matrix in which exactly 16 entries are 1. The following two pictures show the membership-relation $\mathsf{E} : N \leftrightarrow 2^N$ and, below it, the transpose of v, that is, the row vector $v^\mathsf{T} : \mathbf{1} \leftrightarrow 2^N$.



The 32 columns of E describe all coalitions. A comparison of the pictures (here the transposition of v is adequate) shows that the 1-entries of the vector-model precisely designate the columns of E that belong to the membership-model.

5 Three Applications Concerning Power of Players

As already mentioned in the introduction, simple games are very useful for the comparison and measurement of power in decision-making processes. In this section we consider three different possibilities to describe power and show how they can be specified using relation algebra and the membership-model. We also demonstrate how RelView can used to evaluate the relation-algebraic specifications. Our starting point is the following notion, first introduced in [12].

Definition 5.1 *A winning coalition of a simple game is* minimal winning *if every proper subset is losing.*

In the following theorem we show how to specify the set $\mathcal{W}^\diamond$ of all minimal winning coalitions as a subset of the set $\mathcal{W}$ of all winning ones via a vector of type $[\mathcal{W} \leftrightarrow \mathbf{1}]$. An immediate consequence is the column-wise enumeration of $\mathcal{W}^\diamond$.

halshs-00587690, version 1 - 21 Apr 2011

Theorem 5.1 *If $M : N \leftrightarrow \mathcal{W}$ is the membership-model of a simple game, then $m := \overline{(\overline{\mathsf{I}} \cap \overline{\overline{M}^{\mathsf{T}} M})\mathsf{L}} : \mathcal{W} \leftrightarrow \mathbf{1}$ describes the minimal winning coalitions $\mathcal{W}^{\diamond}$ as subset of $\mathcal{W}$ and $M^{\diamond} := Minj(m)^{\mathsf{T}} : N \leftrightarrow \mathcal{W}^{\diamond}$ column-wisely enumerates $\mathcal{W}^{\diamond}$.*

Proof. In the specification of the vector m the type of L is $[\mathcal{W} \leftrightarrow \mathbf{1}]$ and the type of I is $[\mathcal{W} \leftrightarrow \mathcal{W}]$. Now, we obtain for all $C \in \mathcal{W}$ that

$$\begin{array}{rcl} m_C & \Leftrightarrow & \overline{(\overline{\mathsf{I}} \cap \overline{\overline{M}^{\mathsf{T}} M})\mathsf{L}}_C \\ & \Leftrightarrow & \neg\exists\, D \in \mathcal{W} : \overline{\mathsf{I}}_{C,D} \wedge (\overline{\overline{M}^{\mathsf{T}} M})_{C,D} \wedge \mathsf{L}_D \\ & \Leftrightarrow & \neg\exists\, D \in \mathcal{W} : C \neq D \wedge (\overline{\overline{M}^{\mathsf{T}} M})_{C,D} \\ & \Leftrightarrow & \neg\exists\, D \in \mathcal{W} : C \neq D \wedge \neg\exists\, i \in N : \overline{M}_{i,C} \wedge M_{i,D} \\ & \Leftrightarrow & \neg\exists\, D \in \mathcal{W} : C \neq D \wedge \forall\, i \in N : M_{i,D} \rightarrow M_{i,C} \\ & \Leftrightarrow & \neg\exists\, D \in \mathcal{W} : C \neq D \wedge \forall\, i \in N : i \in D \rightarrow i \in C \\ & \Leftrightarrow & \neg\exists\, D \in \mathcal{W} : D \subset C. \end{array}$$

This shows that m describes $\mathcal{W}^{\diamond}$ as subset of $\mathcal{W}$. The second claim follows from the fact that for all players $k \in N$ and minimal winning coalitions $C \in M^{\diamond}$ we have (due to the component-wise description of embedding-relations in Sect. 2)

$$\begin{array}{rcl} M^{\diamond}_{k,C} & \Leftrightarrow & (Minj(m)^{\mathsf{T}})_{k,C} \\ & \Leftrightarrow & \exists\, D \in \mathcal{W} : M_{k,D} \wedge inj(m)_{C,D} \\ & \Leftrightarrow & \exists\, D \in \mathcal{W} : k \in D \wedge C = D \\ & \Leftrightarrow & k \in C. \end{array}$$

□

When investigating the power of players, one possibility is to identify some key players having different strength and then to classify the set of players accordingly. In this paper we consider the following key players.

Definition 5.2 *Player k of a simple game is a* dictator *if $\{k\}$ is the only minimal winning coalition, a* veto player *if it belongs to all minimal winning coalitions, and a* dummy player *if it does not belong to any minimal winning coalition.*

A dictator is the most powerful player of a simple game. He can enforce any decision without help of the other players. There exists at most one dictator. A veto player is needed to win, but he cannot win on his own. If, however, any coalition that contains this player is winning, then he is a dictator. Finally, a dummy player is a player without any power. Next, we present relation-algebraic specifications of these key players.

Theorem 5.2 *Let the relation $M^{\diamond} : N \leftrightarrow \mathcal{W}^{\diamond}$ be as in Theorem 5.1 If we define $a := syq(\mathsf{I}, M^{\diamond})\ \overline{\mathsf{I}\mathsf{L}} : N \leftrightarrow \mathbf{1}$, $b := \overline{\overline{M^{\diamond}}\mathsf{L}} : N \leftrightarrow \mathbf{1}$ and $c := \overline{M^{\diamond}\mathsf{L}} : N \leftrightarrow \mathbf{1}$, then for all players $k \in N$ it holds that it is a dictator iff a_k, a veto player iff b_k, and a dummy player iff c_k.*

Proof. Notice, that in the specification of a besides $\mathsf{L} : \mathcal{W}^{\diamond} \leftrightarrow \mathbf{1}$ two different identity relations appear, viz. $\mathsf{I} : N \leftrightarrow N$ in $syq(\mathsf{I}, M^{\diamond})$ and $\mathsf{I} : \mathcal{W}^{\diamond} \leftrightarrow \mathcal{W}^{\diamond}$ in $\overline{\mathsf{I}\mathsf{L}}$.

halshs-00587690, version 1 - 21 Apr 2011

Now, the first claim follows from the calculation

$$\begin{aligned} a_k &\Leftrightarrow (syq(\mathsf{I}, M^\diamond)\ \overline{\mathsf{I}\,\mathsf{L}}\,)_k \\ &\Leftrightarrow \exists\, C \in \mathcal{W}^\diamond : syq(\mathsf{I}, M^\diamond)_{k,C} \wedge (\ \overline{\mathsf{I}\,\mathsf{L}}\,)_C \\ &\Leftrightarrow \exists\, C \in \mathcal{W}^\diamond : (\forall\, i \in N : \mathsf{I}_{i,k} \leftrightarrow M^\diamond_{i,C}) \wedge \neg\exists\, D \in \mathcal{W}^\diamond : \bar{\mathsf{I}}_{C,D} \wedge \mathsf{L}_D \\ &\Leftrightarrow \exists\, C \in \mathcal{W}^\diamond : (\forall\, i \in N : i = k \leftrightarrow i \in C) \wedge \neg\exists\, D \in \mathcal{W}^\diamond : C \neq D \\ &\Leftrightarrow \exists\, C \in \mathcal{W}^\diamond : C = \{k\} \wedge \neg\exists\, D \in \mathcal{W}^\diamond : C \neq D \\ &\Leftrightarrow k \text{ is a dictator} \end{aligned}$$

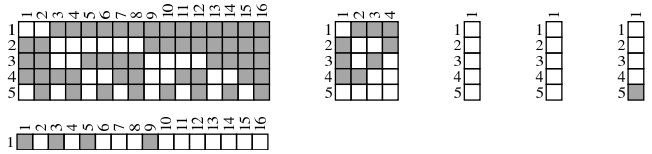
that uses the component-wise description of symmetric quotients given in Sect. 2. The relation L in the specification of b has type $[\mathcal{W}^\diamond \leftrightarrow \mathbf{1}]$, too, and

$$\begin{aligned} b_k &\Leftrightarrow (\ \overline{\overline{M^\diamond}\,\mathsf{L}}\,)_k \\ &\Leftrightarrow \neg\exists\, C \in \mathcal{W}^\diamond : \neg M^\diamond_{k,C} \wedge \mathsf{L}_C \\ &\Leftrightarrow \forall\, C \in \mathcal{W}^\diamond : M^\diamond_{k,C} \\ &\Leftrightarrow \forall\, C \in \mathcal{W}^\diamond : k \in C \\ &\Leftrightarrow k \text{ is a veto player} \end{aligned}$$

is a proof of the second claim. The third claim can be shown in a similar way. □

We have translated the relation-algebraic specifications given in the last two theorems into RelView-code. Then we have applied the RelView-programs to the relations of the game of Example 4.1. Here are the results.

Example 5.1. In the case of the parliament of Germany the RelView tool computed the results given in the following pictures.



The two leftmost pictures show again the membership-model $M : N \leftrightarrow \mathcal{W}$ and (below it) the transpose of the vector $m : \mathcal{W} \leftrightarrow \mathbf{1}$. The four 1-entries of the row vector $m^\mathsf{T} : \mathbf{1} \leftrightarrow \mathcal{W}$ precisely designate those rows of M which represent minimal winning coalitions. If we assemble these four rows in form of a new matrix, we obtain the column-wise enumeration $M^\diamond : N \leftrightarrow \mathcal{W}^\diamond$ given in the next picture as Boolean 5×4 matrix. The remaining three vectors of type $[N \leftrightarrow \mathbf{1}]$ describe, from left to right, the sets of dictators, veto players and dummy players, respectively. Hence, in the present German parliament there exists neither a dictator nor a veto player. But there is a dummy player, viz. Bündnis 90 / Die Grünen.

Another means for measuring power is the desirability relation. It directly compares two players with regard to the strength of forming winning coalitions.

Definition 5.3 *Let (N, f) be a simple game with $\mathcal{W}$ as set of winning coalitions. Then $j \in N$ is* at least as desirable as *$i \in N$, denoted by $i \preceq_D j$, if for all $C \in 2^N$ from $i, j \notin C$ and $C \cup \{i\} \in \mathcal{W}$ it follows $C \cup \{j\} \in \mathcal{W}$.*

halshs-00587690, version 1 - 21 Apr 2011

In words $i \preceq_D j$ says that if i can form a winning coalition with some further players, then j can do either. The desirability relation $\preceq_D$ is a pre-order on the players. With its help a lot of problems on simple games easily can be solved. E.g., $\preceq_D$ is linear iff the game is swap-robust, which means that a one-for-one exchange of players between two winning coalitions leaves at least one of them winning. Next, we show how to specify $i \preceq_D j$ by means of relation algebra.

Theorem 5.3 *Let $M : N \leftrightarrow \mathcal{W}$ be the membership model of a simple game and assume the players $i, j \in N$ to be described by the points $p, q : N \leftrightarrow \mathbf{1}$, respectively. If $\mathsf{E} : N \leftrightarrow 2^N$ is a membership-relation, then $i \preceq_D j$ is equivalent to the inclusion $\overline{\mathsf{E}^\mathsf{T} p} \cap \overline{\mathsf{E}^\mathsf{T} q} \cap syq(\mathsf{E} \cup p\mathsf{L}, M)\mathsf{L} \subseteq syq(\mathsf{E} \cup q\mathsf{L}, M)\mathsf{L}$.*

Proof. Since the point p describes player i, we have for all $C \in 2^N$ that

$$(\overline{\mathsf{E}^\mathsf{T} p})_C \;\Leftrightarrow\; \neg\exists\, k \in N : \mathsf{E}_{k,C} \wedge p_k \;\Leftrightarrow\; \neg\exists\, k \in N : k \in C \wedge k = i \;\Leftrightarrow\; i \notin C$$

(for the second step, see Sect. 2) and also that (the L in $p\mathsf{L}$ has type $[\mathbf{1} \leftrightarrow 2^N]$ and the L composed from the right to the symmetric quotient has type $[\mathcal{W} \leftrightarrow \mathbf{1}]$)

$$\begin{aligned}
(syq(\mathsf{E} \cup p\mathsf{L}, M)\mathsf{L})_C \;&\Leftrightarrow\; \exists\, D \in \mathcal{W} : syq(\mathsf{E} \cup p\mathsf{L}, M)_{C,D} \wedge \mathsf{L}_D \\
&\Leftrightarrow\; \exists\, D \in \mathcal{W} : \forall\, k \in N : (\mathsf{E}_{k,C} \vee p_k) \leftrightarrow M_{k,D} \\
&\Leftrightarrow\; \exists\, D \in \mathcal{W} : \forall\, k \in N : (k \in C \vee k = i) \leftrightarrow k \in D \\
&\Leftrightarrow\; \exists\, D \in \mathcal{W} : C \cup \{i\} = D \\
&\Leftrightarrow\; C \cup \{i\} \in \mathcal{W}.
\end{aligned}$$

The latter derivation employs the component-wise description of symmetric quotients given in Sect. 2. In the same way for all $C \in 2^N$ we get the equivalence of $(\overline{\mathsf{E}^\mathsf{T} q})_C$ and $j \notin C$ and of $(syq(\mathsf{E} \cup q\mathsf{L}, M)\mathsf{L})_C$ and $C \cup \{j\} \in \mathcal{W}$ from the fact that the point q describes player j. Using the just shown equivalences in

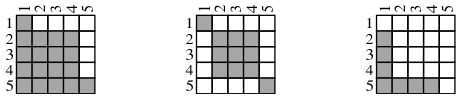
$$\begin{aligned}
i \preceq_D j \;&\Leftrightarrow\; \forall\, C \in 2^N : i \notin C \wedge j \notin C \wedge C \cup \{i\} \in \mathcal{W} \to C \cup \{j\} \in \mathcal{W} \\
&\Leftrightarrow\; \forall\, C \in 2^N : (\overline{\mathsf{E}^\mathsf{T} p} \cap \overline{\mathsf{E}^\mathsf{T} q} \cap syq(\mathsf{E} \cup p\mathsf{L}, M)\mathsf{L})_C \to (syq(\mathsf{E} \cup q\mathsf{L}, M)\mathsf{L})_C \\
&\Leftrightarrow\; \overline{\mathsf{E}^\mathsf{T} p} \cap \overline{\mathsf{E}^\mathsf{T} q} \cap syq(\mathsf{E} \cup p\mathsf{L}, M)\mathsf{L} \subseteq syq(\mathsf{E} \cup q\mathsf{L}, M)\mathsf{L},
\end{aligned}$$

we obtain the claimed result. □

As a consequence we get the desirability relation $\preceq_D : N \leftrightarrow N$ as the union of all compositions pq^T, where p and q range over all points from $[N \leftrightarrow \mathbf{1}]$ such that the right-hand side of Theorem 5.3 holds. This algorithm easily can be implemented in RelView via two nested loops. From $\preceq_D$ two further relations on players are derived, viz. the *more desirability relation* $\prec_D$ as intersection of $\preceq_D$ and the complement of its transpose, and the *equal desirability relation* $\equiv_D$ as intersection of $\preceq_D$ and its transpose.

Example 5.2. Let us consider again the simple game introduced in Example 4.1. The following pictures show, from left to right, the relations $\preceq_D$, $\prec_D$ and $\equiv_D$ as computed by RelView:

halshs-00587690, version 1 - 21 Apr 2011



Even though the players 2, 3 and 4 have different weights, they are equally desirable. Such players are also called *symmetric*.

Indices are a third means to measure power. A well-established index goes back to [1]. Under the assumption that all coalitions are equally likely and that each player votes 'yes' or 'no' with probability $\frac{1}{2}$, the power of k is defined as the probability that k is decisive for the outcome. If all indices are normalized in such a way that their sum equals to 1, this leads to the following specification.

Definition 5.4 *Let (N, f) be a simple game with set of winning coalitions $\mathcal{W}$. Then the pair $\langle k, C\rangle \in N \times \mathcal{W}$ is called a* swing *if $k \in C$ and $C \setminus \{k\} \notin \mathcal{W}$. The* Banzhaf power index *of $k \in N$ is defined as $\frac{\eta_k}{\eta}$, where η_k is the number of swings with first component k and η is the number of all swings.*

If we have a relation $B : N \leftrightarrow \mathcal{W}$ at hand that precisely contains the swings of a simple game, then η_k equals the number of 1-entries of row k of B and η equals the total number of 1-entries of B. The next theorem presents a relation-algebraic specification of this relation.

Theorem 5.4 *Assume $M : N \leftrightarrow \mathcal{W}$ to be the membership model of a simple game. If we define $B := M \cap \overline{rel(syq([\bar{\mathsf{I}}, M], M)\mathsf{L})} : N \leftrightarrow \mathcal{W}$, then for all $k \in N$ and $C \in \mathcal{W}$ the pair $\langle k, C\rangle$ is a swing iff $B_{k,C}$.*

Proof. Notice that in the definition of B the relations I and L have the types $[N \leftrightarrow N]$ and $[\mathcal{W} \leftrightarrow \mathbf{1}]$, respectively. We start the proof with

$$\begin{aligned}
(syq([\bar{\mathsf{I}}, M], M)\mathsf{L})_{\langle k,C\rangle} &\Leftrightarrow \exists\, D \in \mathcal{W} : syq([\bar{\mathsf{I}}, M], M)_{\langle k,C\rangle,D} \wedge \mathsf{L}_D \\
&\Leftrightarrow \exists\, D \in \mathcal{W} : \forall\, i \in N : [\bar{\mathsf{I}}, M]_{i,\langle k,C\rangle} \leftrightarrow M_{i,D} \\
&\Leftrightarrow \exists\, D \in \mathcal{W} : \forall\, i \in N : (\bar{\mathsf{I}}_{i,k} \wedge M_{i,C}) \leftrightarrow M_{i,D} \\
&\Leftrightarrow \exists\, D \in \mathcal{W} : \forall\, i \in N : (i \neq k \wedge i \in C) \leftrightarrow i \in D \\
&\Leftrightarrow \exists\, D \in \mathcal{W} : C \setminus \{k\} = D \\
&\Leftrightarrow C \setminus \{k\} \in \mathcal{W},
\end{aligned}$$

using the component-wise descriptions of symmetric quotients and pairings given in Sect. 2. If we combine this result with the component-wise description of the function *rel* given in Sect. 2, too, we can complete the proof by

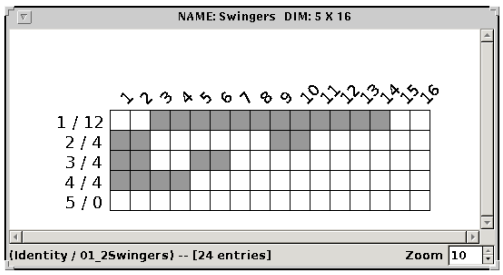
$$\begin{aligned}
B_{k,c} &\Leftrightarrow (M \cap \overline{rel(syq([\bar{\mathsf{I}}, M], M)\mathsf{L})}\,)_{k,C} \\
&\Leftrightarrow M_{k,C} \wedge \neg rel(syq([\bar{\mathsf{I}}, M], M)\mathsf{L})_{k,C} \\
&\Leftrightarrow k \in C \wedge \neg(syq([\bar{\mathsf{I}}, M], M)\mathsf{L})_{\langle k,C\rangle} \\
&\Leftrightarrow k \in C \wedge C \setminus \{k\} \notin \mathcal{W} \\
&\Leftrightarrow \langle k, C\rangle \text{ is a swing .}
\end{aligned}$$

□

halshs-00587690, version 1 - 21 Apr 2011

If the RelView tool depicts B as a Boolean matrix in the relation-window, then in the window's status bar the number of 1-entries of B is shown. Furthermore, it is able to mark the rows and columns of B for explanatory purposes. So far, we have only shown the possibility to attach consecutive row and column numbers to relations. But also the numbers of 1-entries can be attached as labels. This immediately allows to compute Banzhaf power indices using the tool.

Example 5.3. The following picture shows the relation-window of RelView, where the swing-relation $B : N \leftrightarrow \mathcal{W}$ for the parliament of Germany is depicted.



From the message `[24 entries]` at the bottom of this window and the second components of the row labels we obtain the following Banzhaf power indices: CDU/CSU $\frac{12}{24}$, SPD $\frac{4}{24}$, FDP $\frac{4}{24}$, Die Linke $\frac{4}{24}$ and Bündnis 90 / Die Grünen $\frac{0}{24}$.

6 Conclusions

In spite of the fact that RelView implements relations very efficiently, due of its general approach it cannot compete with special algorithms tailored for hard game-theoretic problems. Such algorithms even can tackle games like the US Federal System game with 537 players. We believe that the real attraction of RelView lies in its flexibility: New types and properties of games are introduced all the time and RelView proved to be ideal for experimenting with new concepts while avoiding unnecessary overhead. By now, systematic experiments are accepted as means for obtaining new scientific insights and results, and tools for this purpose become increasingly important as one proceeds in investigations.

In those cases where the present RelView tool is not effective, all hope is not lost. By our experiments we have noticed that in many cases the BDDs of the results are relatively small. This led to the insight that BDDs are an excellent means for solving efficiently game-theoretic problems, especially if they are manipulable in full generality and not only indirectly via the language of RelView. Consequently, the current direction in the development of RelView is to make it more extensible by expanding its interface in such a way that it is possible to outsource program logic into plug-ins. By specific game-theoretic plug-ins we hope to be able to treat successfully in the future also large problems.

halshs-00587690, version 1 - 21 Apr 2011

References

[1] J.F. Banzhaf: Weighted voting doesn't work: A mathematical analysis. Rutgers Law Review 19, 317-343 (1965)

[2] R, Berghammer, S. Bolus, A. Rusinowska, H. de Swart: A relation-algebraic approach to simple games. Europ. J. Operat. Res. 210, 68-80 (2011).

[3] R. Berghammer, B. Braßel: Computing and visualizing closure objects using relations and RelView. In: V.G. Gansha et al. (eds.): Computer Algebra in Scientific Computing, LNCS 5743, Springer, 29-44 (2009).

[4] R. Berghammer, F. Neumann: RelView – An OBDD-based Computer Algebra system for relations. In: V.G. Gansha et al. (eds.): Computer Algebra in Scientific Computing, LNCS 3718, Springer, 40-51 (2005).

[5] R. Berghammer, A. Rusinowska, H. de Swart: Applying relational algebra and RelView to coalition formation. Europ. J. Operat. Res. 178, 530-542 (2007).

[6] R. Berghammer, A. Rusinowska, H. de Swart: An interdisciplinary approach to coalition formation. Europ. J. Operat. Res. 195, 487-496 (2009).

[7] A. van Deemen: Dominant players and minimum size coalitions. Europ. J. Polit. Res. 17, 313-332 (1989).

[8] A. van Deemen: Coalition formation in centralized policy games. J. Theoret. Polit. 3, 139-161 (1991).

[9] E. Elkind, L.A. Goldberg, P.W. Goldberg, M. Wooldridge: On the computational complexity of weighted voting games. Ann. Math. Artif Intell. 56, 109-131 (2009).

[10] B. Leoniuk: ROBDD-basierte Implementierung von Relationen und relationalen Operationen mit Anwendungen. Diss., Univ. Kiel (2001).

[11] U. Milanese: Zur Implementierung eines ROBDD-basierten Systems für die Manipulation und Visualisierung von Relationen. Diss., Univ. Kiel (2003).

[12] J. von Neumann, O. Morgenstern: Theory of games and economic behaviour. Princeton University Press (1944).

[13] B. Peleg, P. Sudhölter: Introduction to the theory of cooperative games Springer (2003).

[14] H. Peters: Game theory: A Multi-leveled approach. Springer (2008).

[15] K. Prasad, J.S. Kelly: NP-completeness of some problems concerning voting games. Int. J. Game Theory 19, 1-9 (1990).

[16] G. Schmidt, T. Ströhlein: Relations and graphs, Springer (1993).

[17] A.D. Taylor: Mathematics and politics. Springer (1995).

halshs-00587690, version 1 - 21 Apr 2011