The Journal of Logic and Algebraic Programming 81 (2012) 625–642

Contents lists available at SciVerse ScienceDirect

The Journal of Logic and Algebraic Programming

journal homepage: www.elsevier.com/locate/jlap

Relation-algebraic modeling and solution of chessboard independence and domination problems

Rudolf Berghammer *

Institut für Informatik, Christian-Albrechts-Universität zu Kiel, Olshausenstraße 40, 24098 Kiel, Germany

ARTICLE INFO

Article history:
Available online 6 June 2012

Keywords:
Relation algebra
Relational modeling
RelView tool
Chessboard relation
Independence
Domination
Bishops independence number

ABSTRACT

We describe a simple computing technique for solving independence and domination problems on rectangular chessboards. It rests upon relational modeling and uses the BDD-based specific purpose computer algebra system RelView for the evaluation of the relation-algebraic expressions that specify the problems' solutions and the visualization of the computed results. The technique described in the paper is very flexible and especially appropriate for experimentation. It can easily be applied to other chessboard problems.

1. Introduction

Combinatorial problems concerning chessboards have been of interest for puzzle solvers as well as for mathematicians and computer scientists for a very long time. A well-known chessboard problem is the 8-queens problem, posed by the chess champion Bezzel in the year 1848 in the chess journal "Deutsche Schachzeitung". It asks for the number of possibilities to place 8 non-attacking queens on the classical 8×8 chessboard with 64 squares. In the year 1850 the 92 possibilities were found by Nauck (see [21]), and Pauls (see [22]) showed about 20 years later that 92 is indeed the total number of possibilities. Later on the 8-queens problem was modified and generalized in manifold ways by considering other chessboard topologies (rectangles, toroidal chessboards, staircase boundaries, chessboards with holes) and chess pieces (kings, rooks, bishops and knights) and even combinations of the latter, like queens and pawns. See, for example, [1,26,28] for more details.

The 8-queens problem is related to the independence (that is, mutually exclusive attacks) of chess pieces. Another interesting class of chessboard problems concerns domination (also called covering), In the case of queens and the classical chessboard the queens domination problem asks for the smallest number of queens needed to dominate (i.e., attack) all squares of the chessboard and the number of possibilities to achieve such smallest dominations. It is known that 5 queens suffice and there are 4860 possibilities. Also chessboard domination has been studied already in the 19th century. According to [15], the first explicit statement of this problem was due to Abbe Durand in the year 1861, followed one year later by C.F. de Jaenisch (see [12]). Domination in view of queens and bishops is e.g., investigated in [11] and the rook domination problem is e.g., studied in [10]. As in the case of chessboard independence problems also the monographs [26,28] have to be mentioned because of their fundamental results.

Backtracking is one of the oldest algorithmic techniques for solving combinatorial chessboard problems. In this paper we propose another technique, viz. modeling combined with tool support. It is based on (heterogeneous) relation algebra in the sense of [23,24] as methodological modeling means and consists essentially in the relation-algebraic specification of extremal independent and dominating sets of vertices of a given graph and the enumeration of these sets, as well as the specification

* Tel.: +49 431 7272; fax: +49 431 7613.
E-mail address: rub@informatik.uni-kiel.de

1567-8326/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jlap.2012.05.001

of the "attack graphs" of the chess pieces in the same way. To evaluate the resulting relation-algebraic expressions and to visualize the solutions, the tool RelView (see [3,4]) is used. Integral parts of our technique are membership relations and size-comparison relations on powersets. If these specific relations are implemented via simple Boolean arrays, then exponential space is required and classical backtracking is head and shoulders above the relation-algebraic approach. However, based on the results of [19,20], in RelView relations and many operations on them are implemented very efficiently via binary decision diagrams (abbreviated as BDDs; many details about them can be found in [7,8,27]) and, therefore, our method is no longer considerably slower than backtracking. We believe that it possesses some advantages. First, it is simple. Furthermore, the correctness proofs of the algorithms are very formal and they use only elementary properties of logic and relations. All this drastically reduces the danger of making errors. Finally, our method is very flexible and can easily be adapted to other chessboard topologies and kinds of moves. In combination with the concise RelView programs and the tool's visualization and animation facilities this is ideal for experimenting while avoiding unnecessary overhead.

The remainder of the paper is organized as follows. In Section 2 we present some preliminaries on relation algebra and RelView. The chessboard independence and domination problems we deal with in this paper are described in Section 3. How to solve them with relation-algebraic means is shown in the Sections 4 and 5. They constitute the methodological core of the paper. In Section 4 we develop relation-algebraic specifications for certain independent and dominating sets and in Section 5 we do the same for the attack graphs. Some experimental results we have obtained with the help of RelView are presented in Section 6 for independence and in Section 7 for domination. In each case we consider rectangular chessboards and compare the RelView results with known mathematical facts for quadratic chessboards. As a mentionable result we prove in Section 8 a conjecture of [25] concerning the bishops independence number of proper rectangular chessboards. Section 9 contains some concluding remarks.

2. Relation algebra and the RelView system

We assume the reader to be familiar with the fundamentals of binary relations. Details can be found, for example, in [23,24]. Nevertheless, we provide in this section the notions and notations as used throughout this paper. In particular, we focus on the relation-algebraic modeling of sets and a relation-algebraic specification of direct products and parallel compositions, which are not commonly used and require some detailed explanation.

If X and Y are sets, then a subset R of the direct product $X \times Y$ is a (binary) relation with source X and target Y. We denote the set – in this context also called type – of all relations with source X and target Y (i.e., the powerset $2^{X\times Y}$ of $X \times Y$) by $[X \leftrightarrow Y]$ and write $R : X \leftrightarrow Y$ instead of $R \in [X \leftrightarrow Y]$. A (typed) relation $R : X \leftrightarrow Y$ corresponds to a predicate on $X \times Y$. If X and Y are finite, we may consider R also as a Boolean matrix. These interpretations are 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 predicate and Boolean matrix terminology and notation. In particular, we speak of rows, columns and components of relations and write $R(x, y)$ instead of $\langle x, y\rangle \in R$ or $x\,R\,y$. We will use the following basic operations on relations: $\overline{R}$ (*complement*), $R \cup S$ (*union*), $R \cap S$ (*intersection*), R^{T} (*transposition*) and $R; S$ (*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. Finally, if $R : X \leftrightarrow Y$ is included in $S : X \leftrightarrow Y$ we write $R \subseteq S$ and equality of R and S is denoted as $R = S$.

A *vector* is a relation v with the specific set $\mathbf{1} := \{\bot\}$ as target. In $v(x, \bot)$ the argument $\bot$ is irrelevant, since there is no other element y in $\mathbf{1}$ such that $v(x, y)$. Therefore, for reasons of simplification we write in the following $v(x)$ instead of $v(x, \bot)$. Vectors correspond to predicates on their sources and in the Boolean matrix model they are 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* $\mathsf{M} : X \leftrightarrow 2^X$. These specific relations are defined by $\mathsf{M}(x, Y)$ iff $x \in Y$, for all $x \in X$ and $Y \in 2^X$. A Boolean matrix representation of M requires exponential space. However, in [19] a BDD-implementation of M is presented, the number of vertices of which is linear in the size of X. A combination of embedding relations and membership relations allows a *column-wise enumeration* of subsets of powersets. More specifically, 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 $(\mathsf{M}; inj(v)^{\mathsf{T}})(x, Y)$ iff $x \in Y$. Using Boolean matrix terminology this means that the elements of $\mathfrak{S}$ are described precisely by the columns of the relation $\mathsf{M}; inj(v)^{\mathsf{T}} : X \leftrightarrow \mathfrak{S}$.

Given 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. Throughout this paper, pairs u are assumed to be of the form $u = \langle u_1, u_2\rangle$, i.e., the first component of u is denoted by u_1 and the second component by u_2. For a relation-algebraic approach, it is very useful to consider instead of the projection functions 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 that $\pi(u, x)$ iff $u_1 = x$ and that $\rho(u, y)$ iff $u_2 = y$. Projection relations allow us to specify algebraically the *parallel composition* $R \parallel S : X \times X' \leftrightarrow Y \times Y'$ of relations $R : X \leftrightarrow Y$ and $S : X' \leftrightarrow Y'$ in such a way that the logical equivalence of $(R \parallel S)(u, v)$ and $R(u_1, v_1)$ and $S(u_2, v_2)$ can be shown for all $u \in X \times X'$ and $v \in Y \times Y'$. We get this property by means of the relation-algebraic definition $R \parallel S := \pi; R; \sigma^{\mathsf{T}} \cap \rho; S; \tau^{\mathsf{T}}$, where $\pi : X \times X' \leftrightarrow X$ and $\rho : X \times X' \leftrightarrow X'$ are the two projection relations of $X \times X'$ and $\sigma : Y \times Y' \leftrightarrow Y$ and $\tau : Y \times Y' \leftrightarrow Y'$ are the projection relations of $Y \times Y'$.

As already mentioned, we use RelView to evaluate the relation-algebraic expressions we will develop in this paper. RelView (see [3,4,16,20]) is a specific purpose computer algebra system for the visualization and manipulation of relations and for relational programming. It is written in the C programming language and makes full use of the X-windows graphical user interface. In RelView 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 representation. Because RelView often works on (very) large data, it uses, as already mentioned, a very efficient implementation of relations based on BDDs. The main purpose of the RelView system is the evaluation of relation-algebraic expressions. These are constructed from the relations of its workspace by using pre-defined operations and tests and user-defined functions and programs.[1] RelView functions are defined as usual in mathematics, i.e., in the form $F(X_1, \dots, X_n) = E$ with F as the name of the function, $X_1, \dots, X_n$ as the list of parameters (standing for relations) and E as a relation-algebraic expression that specifies the outcome. A RelView program is much like a function procedure in Modula 2, except that it only uses relations as a data type. It starts with a head line containing the program name and the list of formal parameters. Then the declaration of the local relational domains, functions and variables follows. Declarations of product domains allow one to introduce projection relations and parallel compositions. 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. More details about RelView and many applications of the system can be found, for example, in [2,3,5,6,19,20].

Concerning the implementation of relations using BDDs symbolic model checking must be mentioned, since it precedes RelView's use of BDDs by many years (see e.g., [9]). The difference between model checking tools and RelView in regard to BDDs is that the latter tool completely is geared to heterogeneous relation algebra in the sense of [23,24] and its extensions (by membership relations, embedding relations, relational domains and so forth) and, hence, includes besides the BDD-implementation of relations also those of the most important relation-algebraic constants and operations.

3. Chessboard independence and domination problems

Given a classical chessboard with 8 rows and 8 columns and a chess piece P, an undirected (attack or chessboard) graph may be formed with the 64 squares of the chessboard as its vertices and with two vertices being adjacent precisely when they are different and the chess piece P situated at one is able to move by one step to the other. This directly generalizes to chessboards with $m > 0$ rows and $n > 0$ columns. For example, if the mn squares of the $m \times n$ chessboard correspond to the elements of the direct product $V := X \times Y$ of the sets $X := \{1, \dots, m\}$ and $Y := \{1, \dots, n\}$, then the pairs $u, v \in V$ form an edge $\{u, v\}$ in the (undirected) *rooks graph* iff they are different and, furthermore, $u_1 = v_1$ or $u_2 = v_2$, i.e., iff the corresponding rooks are arranged on different squares and the squares are in the same row or the same column. In a similar way the *kings graph*, the *bishops graph*, the *queens graph* and the *knights graph* may be defined by means of the chess pieces' moves.

With regard to independence, for the chess piece P and an $m \times n$ chessboard the following two questions are then equivalent:

- What is the largest number of non-attacking copies of P that can be placed on the chessboard?
- What is the *independence number* $\alpha(G_P)$ of the chessboard graph G_P for P?

The independence number $\alpha(G)$ of an undirected graph $G = (V, E)$ is the size of a largest independent set, where an *independent* (or stable) set is a subset of the set V of vertices in which no pair of different vertices is adjacent. Furthermore, the number of possibilities to arrange a largest number of non-attacking copies of P on the chessboard equals the size of the set of all largest independent sets of the undirected graph G_P.

In the same way the chess domination problem for the chess piece P can be reduced to another classical graph-theoretic problem that concerns a further important graph parameter. Namely, if a set of vertices of an undirected graph $G = (V, E)$ is called *dominating* (or covering, absorbing) if for all vertices outside of it there exists at least one adjacent vertex inside of it, then the following two questions are equivalent:

- What is the least number of copies of P that have to stand on the chessboard to ensure that each empty square is attacked by at least one copy of P?
- What is the *domination number* $\gamma(G_P)$ of the chessboard graph G_P for P?

The domination number $\gamma(G)$ of an undirected graph $G = (V, E)$ is the size of a smallest dominating set of vertices. Again the number of possibilities to arrange a least number of copies of P on a chessboard such that all squares, where no chess piece stands, are attacked equals the size of the set of all smallest dominating sets of G_P.

[1] It explicitely should be mentioned that the present version of RelView does not have numbers as data type. A possibility to provide numbers, e.g., as arguments of operations, functions and programs, is to model them by the types of relations, i.e., numbers of rows and columns.

The *upper domination number* $\Gamma(G)$ of an undirected graph $G = (V, E)$ is the largest size of a minimal (with respect to set inclusion) dominating set. In the literature independence and domination are also combined, leading to the problem of determining the smallest size of a subset that is at the same time independent and dominating, i.e., a smallest *kernel* of the given graph. The corresponding graph parameter is called *independent domination number* and denoted by $i(G)$. In terms of chessboards, its determination means the following:

- What is the least number of non-attacking copies of P that have to stand on the chessboard to ensure that each empty square is attacked by at least one copy of P?

Again in view of chessboard problems a further variant of graph domination is interesting, viz., the following question.

- Given an undirected graph $G = (V, E)$ and a subset X of the set V of vertices, what is the size of a smallest dominating set that is contained in X?

If $G_Q = (V, E)$ is the queens graph of an $m \times m$ chessboard and X is the set of vertices corresponding to the squares on a diagonal, then the last problem is equivalent to the determination of the so-called *queens diagonal domination number*.

4. Specification of independent and dominating sets

Having reduced the chessboard problems we consider in this paper to classical graph-theoretic problems, in this section we show how to solve the latter ones using relation algebra. The remaining task of relation-algebraically specifying the undirected graphs for the given row and column numbers of the chessboard and the chess pieces rook, bishop, queen, king and knight is postponed to the next section.

Assume $G = (V, E)$ to be an undirected graph. Then we can construct from G a directed graph $G_* = (V, R)$ with the same set of vertices and a relation $R : V \leftrightarrow V$ on it by defining $R(x, y)$ iff $\{x, y\} \in E$, i.e., iff x and y are adjacent in G, for all $x, y \in V$. The relation R is symmetric, that is, we have $R = R^{\mathsf{T}}$. Since edges of undirected graphs are 2-element sets $\{x, y\}$ of vertices, the relation R is also irreflexive, i.e., it holds $R \subseteq \overline{\mathsf{I}}$. Obviously, there is a 1-1-correspondence between the set of undirected graphs on V and the set of directed graphs on V with symmetric and irreflexive edges relations. Since directed graphs are nothing else than relations on sets of vertices, in the following we identify the directed graph $G_* = (V, R)$ with the relation $R : V \leftrightarrow V$ and investigate independence and domination in the context of symmetric and irreflexive relations only. We, furthermore, assume the carrier sets of all relations to be finite. This assumption is needed when we ask for extremal sets with respect to size. Finiteness of the carrier sets is also a prerequisite for the use of the RelView tool.

So, assume $R : V \leftrightarrow V$ to be a symmetric and irreflexive relation on the finite set V. If we specify independence and domination within predicate logic, then the set $Y \in 2^V$ is independent iff the following formula (I) holds, and dominating iff the following formula (D) holds.

$$\text{(I) } \forall x, y : x \in Y \wedge y \in Y \rightarrow \neg R(x, y) \qquad \text{(D) } \forall x : x \notin Y \rightarrow \exists y : y \in Y \wedge R(x, y)$$

In both predicate logic formulae (I) and (D) the quantifiers range over the set V. Starting with (I), we can calculate as given below, where the set $Y \in 2^V$ is arbitrarily given, $\mathsf{M} : V \leftrightarrow 2^V$ is a membership relation and the relation L has type $[1 \leftrightarrow V]$, i.e., is a transposed universal vector:

$$\begin{aligned}
\forall x, y : x \in Y \wedge y \in Y \rightarrow \neg R(x, y) &\iff \forall x : x \in Y \rightarrow \forall y : y \in Y \rightarrow \neg R(x, y) \\
&\iff \forall x : \mathsf{M}(x, Y) \rightarrow \neg \exists y : \mathsf{M}(y, Y) \wedge R(x, y) \\
&\iff \forall x : \mathsf{M}(x, Y) \rightarrow \neg (R; \mathsf{M})(x, Y) \\
&\iff \neg \exists x : \mathsf{M}(x, Y) \wedge (R; \mathsf{M})(x, Y) \\
&\iff \neg \exists x : (\mathsf{M} \cap R; \mathsf{M})(x, Y) \\
&\iff \neg \exists x : \mathsf{L}(\perp, x) \wedge (\mathsf{M} \cap R; \mathsf{M})(x, Y) \\
&\iff \neg (\mathsf{L}; (\mathsf{M} \cap R; \mathsf{M}))(\perp, Y) \\
&\iff \overline{\mathsf{L}; (\mathsf{M} \cap R; \mathsf{M})}^{\mathsf{T}}(Y)
\end{aligned}$$

This calculation replaces step-by-step all logical constructions by relation-algebraic ones. The multiplication of the universal relation $\mathsf{L} : 1 \leftrightarrow V$ from the left leads to a transposed vector, as can be seen from the argument pair $(\perp, Y)$. To obtain a vector, hence, a final transposition is necessary.

As a consequence of the above calculation, the set Y is independent iff the Y-component of the vector $\overline{\mathsf{L}; (\mathsf{M} \cap R; \mathsf{M})}^{\mathsf{T}}$ is true so that, by the definition of the notion "vector description" given in Section 2, the vector

$$indset(R) := \overline{\mathsf{L}; (\mathsf{M} \cap R; \mathsf{M})}^{\mathsf{T}} \tag{1}$$

of type $[2^V \leftrightarrow \mathbf{1}]$ describes the set $\mathfrak{I}$ of all independent sets of V as a subset of 2^V in the sense of Section 2. To obtain from the formula (D) a relation-algebraic specification of the set $\mathfrak{D}$ of all dominating sets, we calculate for an arbitrarily given set $Y \in 2^V$ as follows:

$$\begin{aligned}
\forall x : x \notin Y \rightarrow \exists y : y \in Y \wedge R(x,y) &\iff \forall x : \neg\mathsf{M}(x,Y) \rightarrow \exists y : \mathsf{M}(y,Y) \wedge R(x,y) \\
&\iff \forall x : \overline{\mathsf{M}}(x,Y) \rightarrow (R;\mathsf{M})(x,Y) \\
&\iff \neg\exists x : \overline{\mathsf{M}}(x,Y) \wedge \neg(R;\mathsf{M})(x,Y) \\
&\iff \neg\exists x : \mathsf{L}(\perp,x) \wedge (\overline{\mathsf{M}} \cap \overline{R;\mathsf{M}})(x,Y) \\
&\iff \neg(\mathsf{L};(\overline{\mathsf{M}} \cap \overline{R;\mathsf{M}}))(\perp,Y) \\
&\iff \overline{\mathsf{L};(\overline{\mathsf{M}} \cap \overline{R;\mathsf{M}})}^{\mathsf{T}}(Y)
\end{aligned}$$

This leads to the following vector $domset(R) : 2^V \leftrightarrow \mathbf{1}$ that describes $\mathfrak{D}$ as a subset of the powerset 2^V, where the relations M and L in the specification (2) are as in (1):

$$domset(R) := \overline{\mathsf{L};(\overline{\mathsf{M}} \cap \overline{R;\mathsf{M}})}^{\mathsf{T}} \tag{2}$$

To obtain from the relation-algebraic specifications (1) and (2) relation-algebraic specifications of vectors that describe the set $\mathfrak{I}_{max}$ of all largest independent sets and the set $\mathfrak{D}_{min}$ of all smallest dominating sets, respectively, there are (at least) two possibilities.

The first possibility is to use relation-algebraic specifications of greatest elements and least elements in combination with the so-called *size-comparison relation* $\mathsf{C} : 2^V \leftrightarrow 2^V$ that relates two sets $X, Y \in 2^V$ iff $|X| \le |Y|$. Then we get for the set of all largest independent sets the vector description $maxindset(R) : 2^V \leftrightarrow \mathbf{1}$, with the definition

$$maxindset(R) := max(\mathsf{C}, indset(R)). \tag{3}$$

In the relation-algebraic specification (3) the vector $max(S, v) = v \cap \overline{\overline{S}^{\mathsf{T}}; v} : X \leftrightarrow \mathbf{1}$ describes for a pre-order relation $S : X \leftrightarrow X$ and a vector $v : X \leftrightarrow \mathbf{1}$ the set of all greatest elements; see also [24] for example. Analogously, for the set of all smallest dominating sets we obtain the vector description

$$mindomset(R) := min(\mathsf{C}, domset(R)) \tag{4}$$

of type $[2^V \leftrightarrow \mathbf{1}]$, where now $min(S, v) = max(S^{\mathsf{T}}, v) = v \cap \overline{\overline{S}; v} : X \leftrightarrow \mathbf{1}$ describes, for S and v as above, the set of all least elements.

In [20] it is shown that a size-comparison relation $\mathsf{C} : 2^V \leftrightarrow 2^V$ can be implemented by a BDD with $\mathcal{O}(|V|^2)$ vertices. By a modification of this implementation in the same thesis also an operation *filter* is developed[2] such that for all natural numbers $k > 0$ the vector $filter(k) : 2^V \leftrightarrow \mathbf{1}$ describes the subset $\{Y \in 2^V \mid |Y| < k\}$ of the powerset 2^V. This offers an alternative method for obtaining the sets of extremal sets of $\mathfrak{I}$ and $\mathfrak{D}$. Consider the vector

$$card(k) := filter(k+1) \cap \overline{filter(k)}$$

of type$[2^V \leftrightarrow \mathbf{1}]$. As it describes the set of all subsets of V with size k, by the vector

$$indset(R, k) := indset(R) \cap card(k) \tag{5}$$

of type $[2^V \leftrightarrow \mathbf{1}]$ the set $\mathfrak{I}_k$ of all sets of $\mathfrak{I}$ with size k is described, and by

$$domset(R, k) := domset(R) \cap card(k), \tag{6}$$

[2] The pre-defined RelView version of *filter* is called *cardfilter*. It possesses two arguments, a vector $v : 2^V \leftrightarrow \mathbf{1}$ and a vector $k : K \leftrightarrow \mathbf{1}$, and yields a vector of type $[2^V \leftrightarrow \mathbf{1}]$ as result. If v describes the subset $\mathfrak{S}$ of 2^V, then $cardfilter(v, k)$ describes the subset $\{Y \in \mathfrak{S} \mid |Y| < |K|\}$. This is a typical situation where in RelView a number is modeled by a type of a relation.

which introduces a vector of the same type, the set $\mathfrak{D}_k$ of all sets of $\mathfrak{D}$ with size k is described, both as subsets of 2^V. Practical experiments with the RelView system have shown that in the case of larger chessboards the relation-algebraic specifications (5) and (6) are much more efficient than the specifications (3) and (4) and lead, even when applied iteratively, much faster to the solutions of our chessboard problems. The efficiency of (5) and (6) even can be increased if the filter-process via the vector $card(k)$ and the descriptions of the two sets $\mathfrak{I}$ and $\mathfrak{D}$ are intertwined. In the case of independent sets we worked with the variant

$$indset'(R,k) := \overline{\mathsf{L};\ \overline{\overline{\mathsf{M}\cap R;\mathsf{M}} \cap (card(k);\mathsf{L})^{\mathsf{T}}}}^{\mathsf{T}} \tag{7}$$

that follows from (5) by simple relation-algebraic reasoning. Let c abbreviate the expression $card(k)$. Then we obtain the equivalence of (5) and (7) by the calculation

$$\begin{aligned}
\overline{\mathsf{L};(\mathsf{M}\cap R;\mathsf{M})}^{\mathsf{T}} \cap c &= \overline{\mathsf{L};(\mathsf{M}\cap R;\mathsf{M})}^{\mathsf{T}} \cap \overline{\mathsf{L};\ \overline{c;\mathsf{L}}^{\mathsf{T}}}^{\mathsf{T}} \\
&= \overline{\mathsf{L};(\mathsf{M}\cap R;\mathsf{M}) \cup \mathsf{L};\ \overline{c;\mathsf{L}}^{\mathsf{T}}}^{\mathsf{T}} \\
&= \overline{\mathsf{L};((\mathsf{M}\cap R;\mathsf{M}) \cup \overline{c;\mathsf{L}}^{\mathsf{T}})}^{\mathsf{T}} \\
&= \overline{\mathsf{L};(\ \overline{\overline{\mathsf{M}\cap R;\mathsf{M}}}\ \cup \overline{c;\mathsf{L}}^{\mathsf{T}})}^{\mathsf{T}} \\
&= \overline{\mathsf{L};\ \overline{\overline{\mathsf{M}\cap R;\mathsf{M}} \cap (c;\mathsf{L})^{\mathsf{T}}}}^{\mathsf{T}},
\end{aligned}$$

where the equality $c = \overline{\overline{c;\mathsf{L}};\mathsf{L}} = \overline{\mathsf{L};\ \overline{c;\mathsf{L}}^{\mathsf{T}}}^{\mathsf{T}}$ is used in the first step. The latter property follows from the vector property of c. For determining the dominating sets of size k, we used

$$domset'(R,k) := \overline{\mathsf{L};\ \overline{(\mathsf{M}\cup R;\mathsf{M}) \cap (card(k);\mathsf{L})^{\mathsf{T}}}}^{\mathsf{T}}. \tag{8}$$

This variant can be obtained from the relation-algebraic specification (6) in a way similar to the derivation of variant (7) from (5). The position of the filter expression $(card(k);\mathsf{L})^{\mathsf{T}}$ in (7) and (8), respectively, was found with the help of RelView experiments, since during the evaluation of (1) we noticed that the explosion of the number of BDD-vertices was caused by the composition of L and $\mathsf{M}\cap R;\mathsf{M}$ and during the evaluation of (2) we observed the same behaviour when evaluating the outermost composition in $\mathsf{L};(\overline{\mathsf{M}}\ \cap\ \overline{R;\mathsf{M}})$. But we won't go into details here.

It is obvious that the relation-algebraic specifications we have developed in this section so far also can be used to compute largest minimal dominating sets, i.e., the graph parameter $\Gamma(G)$, as well as smallest kernels, i.e., the graph parameter $i(G)$, and smallest dominating sets that are contained in a given set of vertices, provided the undirected graph G is described by a symmetric and irreflexive edges relation. We demonstrate this by means of the last example only. Let $R : V \leftrightarrow V$ be a symmetric and irreflexive relation and assume the subset X of V to be described by the vector $v : V \leftrightarrow \mathbf{1}$. Then we have for all $Y \in 2^V$ the equivalence

$$Y \subseteq X \iff \forall x : x \in Y \to x \in X \iff \neg\exists x : \mathsf{M}(x,Y) \wedge v(x) \iff \overline{\mathsf{M}^{\mathsf{T}};v}\,(Y).$$

As an immediate consequence, the variant $mindomset'(R,v) : 2^V \leftrightarrow \mathbf{1}$ of the relation-algebraic specification (4), that is defined by

$$mindomset'(R,v) := min(\mathsf{C}, domset(R) \cap \overline{v^{\mathsf{T}};\mathsf{M}}^{\mathsf{T}}), \tag{9}$$

solves the posed problem. In the case of the filter approach to compute smallest dominating sets, we obtain in the same way $domset''(R,k,v) : 2^V \leftrightarrow \mathbf{1}$, with the definition

$$domset''(R,k,v) := domset'(R,k) \cap \overline{v^{\mathsf{T}};\mathsf{M}}^{\mathsf{T}}, \tag{10}$$

as vector description of the set of all dominating subsets that have size k and are contained in X. The use of the expression $\overline{v^{\mathsf{T}};\mathsf{M}}^{\mathsf{T}}$ in the relation-algebraic specifications (9) and (10) instead of $\overline{\mathsf{M}^{\mathsf{T}};v}$ is caused by the BDD-implementation of relations. A transposition of the relation $\mathsf{M} : V \leftrightarrow 2^V$ is very expensive and leads, even for a medium-sized set V, to a huge BDD. But a transposition of a relation with source or target $\mathbf{1}$ is trivial. It only means to exchange source and target, the BDD remains unchanged. See [20] for details.

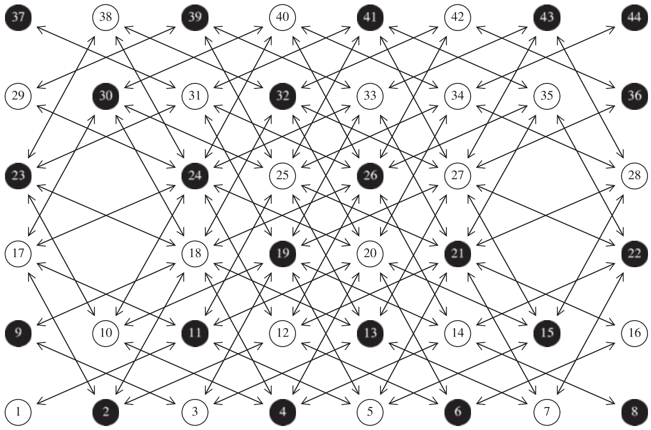

Fig. 1. Largest arrangement of non-attacking knights on a chessboard with holes.

It also should be mentioned that all sets of sets we have specified in this section by means of vector descriptions immediately can be enumerated by the columns of relations using the technique described in Section 2. For instance, the relation $\mathsf{M}; inj(indset(R))^{\mathsf{T}} : V \leftrightarrow \mathfrak{I}$ column-wisely enumerates the set $\mathfrak{I}$ of all independent sets of R and the relation $\mathsf{M}; inj(maxindset(R))^{\mathsf{T}} : V \leftrightarrow \mathfrak{I}_{max}$ column-wisely enumerates the subset $\mathfrak{I}_{max}$ of all largest ones. Executing such column-wise enumerations with RelView not only immediately leads to the graph parameters and solutions of the chessboard problems we are interested in. The tool also allows us to mark vertices of graphs with vectors, i.e., columns of relations. This is very useful for the visualization of results as we will demonstrate now.

In RelView relations and graphs interactively can be drawn and manipulated on the screen. This feature allows to play and experiment with many chessboard topologies without large effort. To give an example, in Fig. 1 the knights graph for a 6×8 chessboard with four holes is shown, together with a largest arrangement of 22 knights; their positions are indicated by black vertices. In the picture we assume, guided by the labeling of the classical 8×8 chessboard, that the rows of the chessboard are numbered from bottom to top and the columns are numbered from left to right. Hence, the four squares $\langle 3, 2\rangle$, $\langle 3, 7\rangle$, $\langle 4, 2\rangle$ and $\langle 4, 7\rangle$ are missing. The labels of the vertices of the RelView graph correspond to the numbering of the squares from 1 to 44 in the fashion beginning with 1 in the lower left-hand corner. For this example there exist exactly 8 largest independent arrangements. Their column-wise enumeration in the sense of Section 2 is shown as a Boolean RelView matrix in Fig. 2. In such a matrix representation of a relation a black square denotes the truth value 1 and a white square denotes the truth value 0. To save space, the matrix of the figure represents the transpose of the relation $\mathsf{M}; inj(maxindset(R))^{\mathsf{T}} : V \leftrightarrow \mathfrak{I}_{max}$, with R as the relation corresponding to the above graph. The transpose of the first (i.e., top) row of the RelView matrix of Fig. 2 describes the largest independent set that is indicated in the RelView graph of Fig. 1 by black vertices.

5. Specification of chessboard relations for rectangular chessboards

To model an $m \times n$ chessboard, we assume sets $X := \{1, \dots, m\}$ and $Y := \{1, \dots, n\}$ for the rows and columns, respectively, and represent the squares by the elements of the direct product $V := X \times Y$. In Section 3 we have already mentioned the notion of an (undirected) chessboard graph for a chess piece P and in Section 4 that it suffices to consider instead of the graphs the corresponding symmetric and irreflexive relations. Like the relation-algebraic specifications of (extremal) independent and dominating sets we have developed in Section 4, also the relation-algebraic specifications of the *chessboard relations* we will develop in this section are based on relation algebra in the sense of Section 2, but now supported by one additional fact.

As already mentioned in Section 4, all carrier sets of the relations we consider in the context of chessboard problems are assumed to be finite. In the following, we additionally suppose them to be equipped with a linear strict-order and the corresponding partial successor function to be available as a relation. To be more precise, if $z_1 < z_2 < \cdots < z_n$ is the

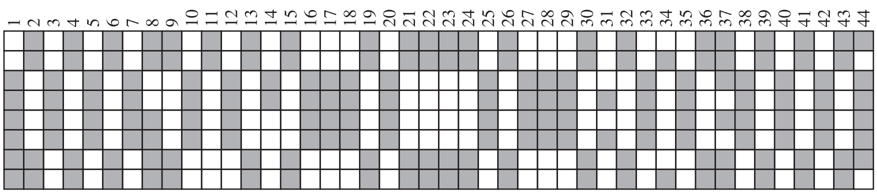

Fig. 2. Column-wise enumeration of all largest arrangements.

ordering of a given finite set Z, then we suppose a relation $\mathsf{S}_Z : Z \leftrightarrow Z$ to be available such that for all $x, y \in Z$ it holds $\mathsf{S}_Z(x, y)$ iff there exists $i \in \mathbb{N}$ such that $x = z_i$ and $y = z_{i+1}$. In terms of order theory, S_Z is the *cover relation* (or the Hasse diagram) of the strict-order $<$ and, thus, the latter relation equals the transitive closure of the former. In RelView, the strict-order is implicitly given via the internal enumeration of the set Z within the tool, and the relation S_Z may be computed by the pre-defined operation `succ`.

For given natural numbers $m, n > 0$, we assume $1 < 2 < \cdots < m$ to be the ordering of the $m \times n$ chessboard's row set X and $1 < 2 < \cdots < n$ to be the ordering of its column set Y. Graphically, we suppose the rows to be numbered from bottom to top and the columns from left to right, both in ascending order so that $\langle 1, 1\rangle$ is the lowermost-leftmost square and $\langle m, n\rangle$ is the uppermost-rightmost one. Using the parallel composition construction of Section 2, the building blocks for the construction of the chessboard relations (except the knights relation) can be specified, viz., by the equations

$$up(\mathsf{S}_X) := \mathsf{S}_X \parallel \mathsf{I} \qquad right(\mathsf{S}_Y) := \mathsf{I} \parallel \mathsf{S}_Y \tag{11}$$

for the unidirectional vertical upward ("↑") and horizontal rightward ("→") one-square moves, respectively, and by the equations

$$pdiag(\mathsf{S}_X, \mathsf{S}_Y) := \mathsf{S}_X \parallel \mathsf{S}_Y \qquad ndiag(\mathsf{S}_X, \mathsf{S}_Y) := \mathsf{S}_X^{\mathsf{T}} \parallel \mathsf{S}_Y \tag{12}$$

for the unidirectional one-square moves in the positive diagonal direction ("↗") and the negative diagonal direction ("↘") of the chessboard, respectively. That the four relations introduced in (11) and (12) (each of them has $[V \leftrightarrow V]$ as type) in fact specify the claimed moves can be seen as follows. Assume $u = \langle u_1, u_2\rangle \in V$ and $v = \langle v_1, v_2\rangle \in V$ to be squares of the given chessboard. With the help of the point-wise description of the parallel composition given in Section 2, we can calculate as follows:

$$\begin{aligned} up(\mathsf{S}_X)(u, v) &\iff (\mathsf{S}_X \parallel \mathsf{I})(u, v) \\ &\iff \mathsf{S}_X(u_1, v_1) \wedge \mathsf{I}(u_2, v_2) \\ &\iff u_1 + 1 = v_1 \wedge u_2 = v_2 \end{aligned}$$

This equivalence shows that $up(\mathsf{S}_X)$ specifies for all squares $u \in V$ the move from u to the square $\langle u_1 + 1, u_2\rangle$, i.e., to the vertical upper neighbour square, if it exists. Furthermore, for all $u, v \in V$ the subsequent equivalence holds:

$$\begin{aligned} pdiag(\mathsf{S}_X, \mathsf{S}_Y)(u, v) &\iff (\mathsf{S}_X \parallel \mathsf{S}_Y)(u, v) \\ &\iff \mathsf{S}_X(u_1, v_1) \wedge \mathsf{S}_Y(u_2, v_2) \\ &\iff u_1 + 1 = v_1 \wedge u_2 + 1 = v_2 \end{aligned}$$

So, $pdiag(\mathsf{S}_X, \mathsf{S}_Y)$ specifies for all squares $u \in V$ the move to its neighbour square $\langle u_1 + 1, u_2 + 1\rangle$ with respect to the positive diagonal direction, again only if it exists. In the same way it can be shown that $right(\mathsf{S}_Y)$ specifies the moves to neighbour squares on the right, i.e., it holds $right(\mathsf{S}_Y)(u, v)$ iff $u_1 = v_1$ and $u_2 + 1 = v_2$, for all $u, v \in V$, and $ndiag(\mathsf{S}_X, \mathsf{S}_Y)$ specifies the moves to the neighbour squares with respect to the negative diagonal direction, i.e., it holds $ndiag(\mathsf{S}_X, \mathsf{S}_Y)(u, v)$ iff $u_1 - 1 = v_1$ and $u_2 + 1 = v_2$, for all $u, v \in V$.

Having specified the basic moves relation-algebraically, it is rather simple to specify the rooks and the bishops chessboard relations. A rook on a chessboard precisely attacks all squares of its row and its column. Using the building blocks (11) for the vertical and horizontal one-square moves, this leads to the rooks chessboard relation (or rooks "attack relation") $rook(\mathsf{S}_X, \mathsf{S}_Y) : V \leftrightarrow V$ as

$$rook(\mathsf{S}_X, \mathsf{S}_Y) := up(\mathsf{S}_X)^{\diamond} \cup right(\mathsf{S}_Y)^{\diamond}, \tag{13}$$

where $R^{\diamond} := R^{+} \cup (R^{+})^{\mathsf{T}}$ denotes the *symmetric closure* of the *transitive closure* $R^{+} := \bigcup_{i>0} R^i$ of a relation R (the powers of R are inductively defined by $R^1 := R$ and $R^{i+1} := R; R^i$). In (13) the expression $up(\mathsf{S}_X)^{\diamond}$ specifies the moves on the columns and the expression $right(\mathsf{S}_Y)^{\diamond}$ those on the rows. The rule that a bishop precisely attacks the squares of the two diagonals he stands on, can be expressed with the help of the building blocks of (12) for diagonal moves as

$$bishop(\mathsf{S}_X, \mathsf{S}_Y) := pdiag(\mathsf{S}_X, \mathsf{S}_Y)^{\diamond} \cup ndiag(\mathsf{S}_X, \mathsf{S}_Y)^{\diamond}; \tag{14}$$

this specifies the bishops chessboard relation $bishop(\mathsf{S}_X, \mathsf{S}_Y) : V \leftrightarrow V$. A queen can move as a rook and as a bishop. As a consequence, the queens chessboard relation $queen(\mathsf{S}_X, \mathsf{S}_Y) : V \leftrightarrow V$ is the union of the rooks chessboard relation and the bishops chessboard relation, i.e., we have

$$queen(\mathsf{S}_X, \mathsf{S}_Y) := rook(\mathsf{S}_X, \mathsf{S}_Y) \cup bishop(\mathsf{S}_X, \mathsf{S}_Y). \tag{15}$$

A king precisely attacks all squares of a chessboard that are adjacent to the square he stands on. If we translate this chess rule into the language of relation algebra, use $R^{\bowtie} := R \cup R^{\mathsf{T}}$ as notation for the symmetric closure of a relation R and the law $(R \cup S)^{\bowtie} = R^{\bowtie} \cup S^{\bowtie}$, then we get

$$king(\mathsf{S}_X, \mathsf{S}_Y) := (up(\mathsf{S}_X) \cup right(\mathsf{S}_Y) \cup pdiag(\mathsf{S}_X, \mathsf{S}_Y) \cup ndiag(\mathsf{S}_X, \mathsf{S}_Y))^{\bowtie} \tag{16}$$

as a relation-algebraic specification of the kings chessboard relation $king(\mathsf{S}_X, \mathsf{S}_Y) : V \leftrightarrow V$.

What remains is the relation-algebraic specification of the knights chessboard relation $knight(\mathsf{S}_X, \mathsf{S}_Y) : V \leftrightarrow V$. A knight precisely attacks those squares which can be reached by moving two squares horizontally and then one square vertically or by moving two squares vertically and then one square horizontally. Obviously, the property $(\mathsf{S}_X \parallel \mathsf{S}_Y^2)(u, v)$ holds iff the square $v \in V$ is reached from the square $u \in V$ by vertically moving one square upwards and then horizontally moving two squares to the right. In the same way all other possibilities for a knight's move can be specified. If we use again the symmetric closure notation and the law $(R \cup S)^{\bowtie} = R^{\bowtie} \cup S^{\bowtie}$ for symmetric closures, then we get the knights chessboard relation as

$$knight(\mathsf{S}_X, \mathsf{S}_Y) := ((\mathsf{S}_X^2 \parallel \mathsf{S}_Y) \cup (\mathsf{S}_X \parallel \mathsf{S}_Y^2) \cup ((\mathsf{S}_X^2)^{\mathsf{T}} \parallel \mathsf{S}_Y) \cup (\mathsf{S}_X^{\mathsf{T}} \parallel \mathsf{S}_Y^2))^{\bowtie}. \tag{17}$$

The informal arguments we have used to obtain the relation-algebraic specifications (13) to (17) also can be replaced by formal reasoning. We will demonstrate this for the rooks chessboard relation as example. Since we only consider 'concrete', that is, set-theoretic relations, the parallel composition of relations fulfills for all relations of appropriate type the so-called 'sharpeness-condition' (see [13])

$$(R_1 \parallel R_2); (S_1 \parallel S_2) = (R_1; S_1) \parallel (R_2; S_2).$$

Consequently, it holds that $(R \parallel S)^n = R^n \parallel S^n$ for all $n > 0$. Using this fact, the equation $\mathsf{I} = \mathsf{I}^n$ for all $n \geq 0$, and that the strict-orders on X and Y are the transitive closures of their covering relations S_X and S_Y, respectively, we can calculate for all $u, v \in V$ as follows, where n ranges over the set $\mathbb{N}$:

$$\begin{aligned}
(\mathsf{S}_X \parallel \mathsf{I})^+(u, v) &\iff \exists n : n > 0 \wedge (\mathsf{S}_X \parallel \mathsf{I})^n(u, v) \\
&\iff \exists n : n > 0 \wedge (\mathsf{S}_X^n \parallel \mathsf{I}^n)(u, v) \\
&\iff \exists n : n > 0 \wedge \mathsf{S}_X^n(u_1, v_1) \wedge \mathsf{I}^n(u_2, v_2) \\
&\iff (\exists n : n > 0 \wedge \mathsf{S}_X^n(u_1, v_1)) \wedge \mathsf{I}(u_2, v_2) \\
&\iff \mathsf{S}_X^+(u_1, v_1) \wedge u_2 = v_2 \\
&\iff u_1 < v_1 \wedge u_2 = v_2
\end{aligned}$$

If we apply this equivalence and the corresponding equivalences for the remaining three expressions $(\mathsf{S}_X \parallel \mathsf{I})^+(v, u)$, $(\mathsf{I} \parallel \mathsf{S}_Y)^+(u, v)$ and $(\mathsf{I} \parallel \mathsf{S}_Y)^+(v, u)$, then we get the following result, the last line of which precisely is the formalization of the fact that a rook on square u attacks the square v,

$$\begin{aligned}
rook(\mathsf{S}_X, \mathsf{S}_Y)(u, v) &\iff up(\mathsf{S}_X)^{\diamond}(u, v) \vee right(\mathsf{S}_Y)^{\diamond}(u, v) \\
&\iff (\mathsf{S}_X \parallel \mathsf{I})^+(u, v) \vee (\mathsf{S}_X \parallel \mathsf{I})^+(v, u) \vee (\mathsf{I} \parallel \mathsf{S}_Y)^+(u, v) \vee (\mathsf{I} \parallel \mathsf{S}_Y)^+(v, u) \\
&\iff (u_1 < v_1 \wedge u_2 = v_2) \vee (v_1 < u_1 \wedge v_2 = u_2) \vee (u_1 = v_1 \wedge u_2 < v_2) \\
&\qquad \vee (v_1 = u_1 \wedge v_2 < u_2) \\
&\iff (u_1 \neq v_1 \wedge u_2 = v_2) \vee (u_1 = v_1 \wedge u_2 \neq v_2) \\
&\iff u \neq v \wedge (u_1 = v_1 \vee u_2 = v_2).
\end{aligned}$$

Very regular non-standard chessboards, like cylindric and toroidal ones and chessboards with holes given by a specific pattern, can be constructed even automatically using variants of the relation-algebraic specifications we have developed in this section.

6. Experimental results concerning independence

The relation-algebraic specifications we have developed in the last two sections can be transformed immediately into the programming language of RelView and, thus, the tool can be used to compute chessboard relations as well as extremal

	3	4	5	6	7	8
3	6	24	60	120	210	336
4		24	120	360	840	1680
5			120	720	2520	6720
6				720	5040	20160
7					5040	40320
8						40320

	3	4	5	6	7	8
3	4/1	4/9	6/1	6/16	8/1	8/25
4		4/79	6/27	6/408	8/81	8/1847
5			9/1	9/64	12/1	12/125
6				9/3600	12/256	12/26040
7					16/1	16/625
8						16/281571

Fig. 3. Independence results for rooks and kings.

independent and dominating sets and to perform experiments. To give an impression how such RelView programs look like, we present in the following some examples. As a first example for a RelView program we show the code for the relation-algebraic specification *indset* of (1).

```
indset(R)
  DECL M, L
  BEG  M = epsi(R);
       L = L1n(R)
       RETURN -(L * (M & R * M))^
  END.
```

This RelView program uses two variables, `M` for the membership relation $\mathsf{M} : V \leftrightarrow 2^V$ and `L` for the transposed universal vector $\mathsf{L} : \mathbf{1} \leftrightarrow V$. The first assignment computes M with the help of the pre-defined RelView operation `epsi` and stores the result in `M`; the second assignment does the same with L and `L` via the pre-defined RelView operation `L1n`. Both operations need the argument `R` only for typing reasons. It provides the source for M and the target for L. Finally, the expression of the return-clause is a direct translation of the right-hand side of (1), where "`-`" denotes complement, "`*`" denotes composition, "`&`" denotes intersection and "`^`" denotes transposition.

We also want to demonstrate how to compute chessboard relations for rectangular chessboards. For that task, we first have to formulate a RelView program for the parallel composition $R \parallel S : X \times X' \leftrightarrow Y \times Y'$. In the following code the first two declarations introduce `XX` and `YY` as names for the direct products $X \times X'$ and $Y \times Y'$. The third declaration introduces two variables `Q1` and `Q1` for relations. Having the direct products as local relational domains at hand, together with the pre-defined RelView operations `p-1` and `p-2` for computing the projection relations, a direct translation of the relation-algebraic specification of $R \parallel S$ into the programming language of *RelView* is trivial.

```
parcomp(R,S)
  DECL XX = PROD(R*R^,S*S^);
       YY = PROD(R^*R,S^*S);
       Q1, Q2
  BEG  Q1 = p-1(XX) * R * p-1(YY)^;
       Q2 = p-2(XX) * S * p-2(YY)^
       RETURN Q1 & Q2
  END.
```

With the help of the parallel composition program and the pre-defined RelView operation `I` that computes the identity relation I of the type of its argument, we now immediately can implement the relation-algebraic specifications *up*, *right*, *pdiag* and *ndiag* of Section 5 as RelView functions as follows:

```
   up(Sx) = parcomp(Sx,I(Sx)).     pdiag(Sx,Sy) = parcomp(Sx,Sy).
right(Sy) = parcomp(I(Sy),Sy).     ndiag(Sx,Sy) = parcomp(Sx^,Sy).
```

For the computation of the chessboard relations it is advantageous to use again RelView programs with local variables. By means of variables it is possible to avoid multiple computations of auxiliary relations. Here is the program for the rook relation, where `trans` is a pre-defined RelView operation for the computation of transitive closures and `|` is a pre-defined RelView operation that realizes unions.

```
rook(Sx,Sy)
  DECL TU, TR
  BEG  TU = trans{up(Sx));
       TR = trans{right(Sy))
       RETURN TU | TU^ | TR | TR^
  END.
```

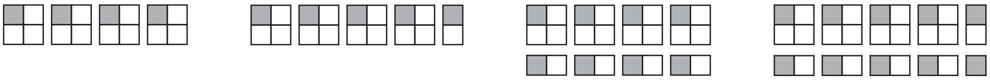

Fig. 4. Non-attacking kings via partitionings of rectangular chessboards.

In a similar way RELVIEW programs for the chessboard relations for bishops, queens, kings and knights can be obtained from the relation-algebraic specifications of Section 5.

Having shown how to translate the specifications of the Sections 4 and 5 into RELVIEW code, for rooks, kings, knights and bishops we present in the following some of the results on chessboard independence we have obtained with the tool for rectangular $m \times n$ chessboards. We also compare them with the results on the same chess pieces presented in [26,28] for the quadratic case. The corresponding results concerning chessboard dominance can be found in Section 7. For symmetry reasons, we may assume $m \leq n$ throughout both sections.

It is rather obvious that for such a chessboard the largest number of non-attacking rooks equals m and that there are exactly $n(n-1)\cdots(n-m+1)$, that is, $\frac{n!}{(n-m)!}$ possibilities to arrange m such rooks on the chessboard. This result was confirmed by our RELVIEW experiments, as shown by the left-hand table of Fig. 3 for the values $3 \leq m \leq n \leq 8$. If $m = n$, then the number of possibilities is $m!$ as already mentioned in [26,28].

In the other table of Fig. 3 for the same values of m and n the largest numbers of non-attacking kings and, separated by a slash, the numbers of possibilities to arrange them, are shown. The largest number of non-attacking kings is given by $\lfloor \frac{m+1}{2} \rfloor \lfloor \frac{n+1}{2} \rfloor$, where the floor expression $\lfloor r \rfloor$ specifies the integer part of the real number r. This generalizes the result of [28] for quadratic chessboards. Our result can be proved by a simple generalization of the proof of [26,28], since also an $m \times n$ chessboard can be partitioned into $\lfloor \frac{m}{2} \rfloor \lfloor \frac{n}{2} \rfloor$ 2×2 parts and, if m and/or n are odd, additional 1×2 parts for the bottom row, 2×1 parts for the rightmost column, and a 1×1 part for the lower right-hand corner. For the four chessboard sizes 2×8, 2×9, 3×8, and 3×9, respectively, this partitioning is illustrated in Fig. 4 (that again was again produced with the help of RELVIEW), where the black squares denote the positions of the 4, 5, 8, and 10 kings, respectively. Very little is known about the number of possibilities to arrange a largest number of non-attacking kings on a chessboard. The only result we are aware of is from [26] and says, generalized to our case, that for m and n being odd numbers there is only one such arrangement. This is due to the fact that in this case a king has to be placed on each corner square of the chessboard and the placement of the king in the 1×1 part (the lowermost-rightmost square) of the partitioning uniquely determines the arrangement of all other kings on the chessboard. See again Fig. 4 and compare the result also with the entries of the right-hand table of Fig. 3.

The two tables of Fig. 5 show some of the experimental results we have obtained for the arrangement of a largest number of non-attacking knights (left-hand table) and non-attacking bishops (right-hand table). Again, we restrict us to the chessboard sizes $3 \leq m \leq n \leq 8$ and use a slash to separate for each experiment the number of chess pieces and the number of arrangements.

In [26] it is shown that for $3 \leq m = n$ the largest number of non-attacking knights equals $\lfloor \frac{m^2+1}{2} \rfloor$. The used arguments can be modified in such a way that they also prove the equality of $\lfloor \frac{mn+1}{2} \rfloor$ and the largest number of non-attacking knights under the assumption $3 \leq m \leq n$. The number $\lfloor \frac{mn+1}{2} \rfloor$ is the maximum of the number w of white squares and the number b of black squares, respectively. If both m and n are odd and, as usual, the lowermost-leftmost square $\langle 1, 1 \rangle$ is black, then it holds $w = b - 1$ and there exists for $m, n \geq 5$ exactly one largest arrangement: Put the knights on the black squares. Otherwise, i.e., if one of the numbers is even, we have $w = b$. If in this case additionally $m \geq 5$ holds, then the arrangement of all knights on the black squares or, alternatively, on the white squares, are the only largest independent arrangements. In the case $m = 4$ and $n \geq 5$ besides these two possibilities there is a third one: Place n knights on row 1 and n knights on row 4 (note, that $\lfloor \frac{4n+1}{2} \rfloor = \frac{4n}{2} = 2n$). For $m = n = 4$ even three further possibilities exist, as demonstrated by the RELVIEW pictures of Fig. 6. Each of them shows the 4×4 knights chessboard graph and three of the six possibly largest independent sets; the latter are indicated by black vertices.

What happens if $1 \leq m \leq 2$? For $m = 1$ and $n \geq 1$ it is obvious that it is possible to place n non-attacking knights on the chessboard. The case $m = 2$ is rather irregular. For example, if we consider here the situation $n = 10$, then the largest number of non-attacking knights is 12 and there exists exactly one such arrangement. It puts on each of the six columns 1, 2, 5, 6, 9 and 10 two knights. Also $n = 11$ leads to 12 as the largest number of non-attacking knights. But now, there are 16 such arrangements possible.

	3	4	5	6	7	8
3	5/2	6/3	8/2	9/4	11/1	12/2
4		8/6	10/3	12/3	14/3	16/3
5			13/1	15/2	18/1	20/2
6				18/2	21/2	24/2
7					25/1	28/2
8						32/2

	3	4	5	6	7	8
3	4/8	6/1	7/3	8/4	9/5	10/9
4		6/16	8/1	8/81	10/9	10/400
5			8/32	10/1	11/9	12/25
6				10/64	12/1	12/729
7					12/128	14/1
8						14/256

Fig. 5. Independence results for knights and bishops.

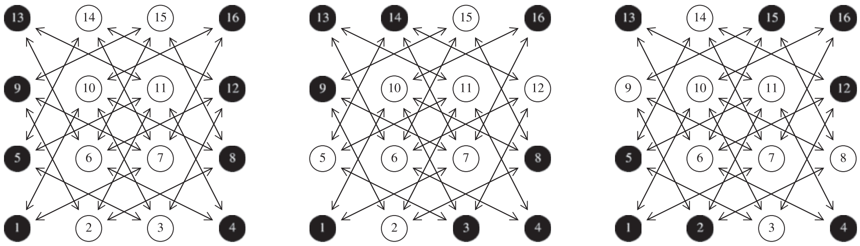

Fig. 6. Three of the 6 possible arrangements of 8 non-attacking knights on the 4 × 4 chessboard.

The largest numbers of non-attacking bishops for $m = n$ is $2m - 2$, and there are 2^m possibilities to place the $2m - 2$ bishops on the $m \times m$ chessboard. This result of [28] is indicated by the diagonal of the right-hand table of Fig. 5. Experimental results for the arrangement of bishops on rectangular chessboards also can be found in [25]. In almost all cases they coincide with our results; we only have corrected the case $m = 4$ and $n = 8$ (in [25] $m = 8$ and $n = 4$) from 10/144 to 10/400 and the case $m = 6$ and $n = 8$ (in [25] $m = 8$ and $n = 6$) from 12/324 to 12/729. Our results also indicate that for $m < n$ it is possible to arrange at most $m + n - 2$ non-attacking bishops if m and n are even, and $m + n - 1$ non-attacking bishops otherwise. This corresponds with the conjecture given in [25]. As already mentioned, we will prove the conjecture in Section 8.

A bishop always stays on squares of a single colour. Hence, the bishops graph possesses precisely two connected components. This fact can be used to reduce the costs for solving the bishops independence problem. Suppose $R : V \leftrightarrow V$ to be the bishops chessboard relation. If we use the vector $p : V \leftrightarrow \mathbf{1}$ to describe the singleton set $\{\langle 1, 1\rangle\}$ (i.e., the black lowermost-leftmost square of the chessboard), then the vector $b := p \cup R^+; p : V \leftrightarrow \mathbf{1}$ describes the subset V_b of V consisting of the black squares and, hence, its complement $\overline{b} : V \leftrightarrow \mathbf{1}$ describes the subset V_w consisting of the white squares. Via the relation-algebraic construction $R_b := \mathit{inj}(b); R; \mathit{inj}(b)^\top$, that defines a relation of type $[V_b \leftrightarrow V_b]$, the restriction of the relation $R : V \leftrightarrow V$ to the set V_b is obtained, and via $R_w := \mathit{inj}(\overline{b}); R; \mathit{inj}(\overline{b})^\top$, that now defines a relation of type $[V_w \leftrightarrow V_w]$, its restriction to the set V_w is given. If we identify undirected graphs and their edges relations, then from the independence numbers of R_b and R_w we get that of R as their sum. Furthermore, the number of largest independent sets of R is the product of the numbers of largest independent sets of R_b and R_w. We have computed via this approach e.g., the numbers for the relation R of the 13 × 16 chessboard. RelView delivered for R_b as well as for R_w the bishops independence number 14 and exactly 233 largest independent sets. Hence, for the 13 × 16 chessboard the largest number of non-attacking bishops is $14 + 14 = 28$, as expected, and the number of possibilities to place them is $233 \cdot 233 = 54289$. These results also can be found in [25].

7. Experimental results concerning domination

In this section we consider the same four chess pieces rook, king, knight and bishop and the same chessboard sizes $3 \le m \le n \le 8$ as in the last section. We also present the experimental results obtained for chessboard domination via the RelView tool in the same tabular way.

As in the case of independence, it is rather obvious that for an $m \times n$ chessboard with $m \le n$ the smallest number of dominating rooks is also m. In [28] it is shown that for $m = n$ the number of possibilities to place m dominating rooks on a chessboard is $m^m + m^m - m!$. This number equals the number of functions from the rows to the columns plus the number of functions from the columns to the rows minus the number of bijective functions between the rows and columns. Namely, a function from the rows to the columns precisely corresponds to an arrangement, where on each row stands a rook, and a function from the columns to the rows precisely corresponds to an arrangement, where on each column stands a rook. Since, however, with this technique all arrangements of the m rooks, where on each row and on each column exactly one rook is placed, are counted twice, and each such specific arrangement precisely corresponds to a bijective function between the rows and columns, their number $m!$ must be substracted to obtain the correct number of possibilities. In the case $m < n$ the situation is much more simple and the number of possibilities to place m dominating rooks on the $m \times n$ chessboard coincides with the number n^m of functions from the rows to the columns. This fact was confirmed by the experiments we have performed with RelView. Some computed numbers are shown in the left-hand table of Fig. 7. The – compared with our results – different number 33514312 of [26, p. 99], for the case $m = n = 8$ seems to be a typing error.

	3	4	5	6	7	8
3	48	64	125	216	343	512
4		488	625	1296	2401	4096
5			6130	7776	16807	32768
6				92592	117649	262144
7					1642046	2097152
8						33514112

	3	4	5	6	7	8
3	1/1	2/4	2/3	2/1	3/8	3/4
4		4/256	4/144	4/16	6/4096	6/1024
5			4/79	4/9	6/1656	6/408
6				4/1	6/64	6/16
7					9/243856	9/29744
8						9/3600

Fig. 7. Domination results for rooks and kings.

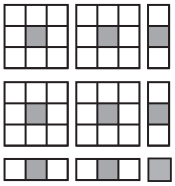

Fig. 8. Dominating kings and a partitioning of the 7×7 chessboard.

In the right-hand table of Fig. 7 the corresponding experimental RelView results for the kings domination problem are presented. The table contains the smallest numbers of kings needed to dominate all squares and the corresponding numbers of possibilities. For quadratic $m \times m$ chessboards, in [26] a surprisingly simple solution of the kings domination problem is given. It is similar to that of the kings independence problem. Assuming $m = 3k$, the given chessboard is divided into k^2 3×3 parts and the $k^2 = (\frac{m}{3})^2 = \lfloor \frac{m+2}{3} \rfloor^2$ kings standing in the centers of the parts constitute a smallest dominating arrangement of the entire chessboard. If m is of the form $3k+1$ or $3k+2$, additional parts and kings for the domination of their squares are needed. In Fig. 8 the partitioning of the 7×7 chessboard into 2^2 3×3 parts and 5 additional parts is depicted, where black squares denote the positions of the 9 dominating kings. But all appearing cases lead to the same expression $\lfloor \frac{m+2}{3} \rfloor^2$ for the smallest number of dominating kings. A simple generalization of the proof of [26] yields $\lfloor \frac{m+2}{3} \rfloor \lfloor \frac{n+2}{3} \rfloor$ as kings domination number of the $m \times n$ chessboard and precisely these numbers appear in the kings domination table. In [26] only one rather trivial result about the number of smallest dominating arrangements of kings is mentioned, viz., that it is one if $m = n = 3k$. This immediately generalizes to the rectangular case if $m = 3k_1$ and $n = 3k_2$; see once again the right-hand table of Fig. 7. Until now, we have not been able to solve the general case.

The left-hand table of Fig. 9 shows experimental RelView results we have obtained for the arrangement of a smallest number of dominating knights. Domination via this chess piece still seems to be an unsolved problem, even in the case of quadratic chessboards. Neither in the standard references [26,28] nor in all other papers on chessboard problems we have studied we have found a general expression for the knights domination number. In [26] only some particular cases of quadratic chessboards are considered.

We close this section with a discussion of the entries of the right-hand table of Fig. 9, where bishops domination numbers and the corresponding numbers of arrangements are shown. In [26,28] it is proved that the domination problem for bishops on quadratic chessboards can be reduced to two domination problems for rooks on diamond-shaped chessboards (i.e., chessboards with staircase boundaries), one for the chessboard's white squares and one for the chessboard's black squares, and, as a consequence, the bishops domination number of the $m \times m$ chessboard is m. At this place it should be mentioned that, by sophisticated counting arguments, in [28] also the numbers of possibilities are determined. But the result is rather difficult. For instance, to place m dominating bishops on an $m \times m$ chessboard, there are $(\frac{(4k+1)(2k)!}{2})^2$ possibilities if m is of the form $4k$. In the case $m = 8$ this yields $(\frac{9(4)!}{2})^2 = 108^2 = 11664$ possibilities, and this was confirmed by the RelView experiments as the lowermost–rightmost number of the bishops domination table of Fig. 9 shows.

Also in the case of a proper rectangular chessboard the bishops domination problem can be reduced to two rook domination problems on chessboards with staircase boundaries. However, these two chessboards do not have the nice diamond-shape as in the quadratic case and, therefore, the arguments of [26,28] do not work anymore. For instance, in the case of the 4×6 chessboard we obtain the two chessboards depicted in Fig. 10 besides the original one. One is determined by the black squares and one by the white squares. The connection to the original chessboard becomes clear if the later is rotated by 45° in clockwise direction. A little reflection shows that each of the non-rectangular chessboards of Fig. 10 can be dominated with 3 rooks, leading to 6 as bishops domination number of the 4×6 chessboard. Compare this fact again with the right-hand table of Fig. 9. Until now, however, we have not found a general expression for the bishops domination number for arbitrary rectangular chessboards.

Finally, it should be remarked that, as in the case of independence, also in the case of domination the property of the bishops graph to consist of two connected components can be used to reduce the costs for computing bishops domination numbers and numbers of corresponding arrangements.

	3	4	5	6	7	8
3	4/8	4/15	4/6	4/2	6/10	8/1192
4		4/9	4/3	4/1	6/1	8/579
5			5/47	6/46	7/47	7/1
6				8/127	8/4	8/1
7					10/10	11/2
8						12/2

	3	4	5	6	7	8
3	3/6	4/36	4/4	6/289	6/66	6/4
4		4/25	4/4	6/576	6/25	8/8100
5			5/104	6/900	6/48	8/1664
6				6/484	6/36	8/26896
7					7/2136	8/28224
8						8/11664

Fig. 9. Domination results for knights and bishops.

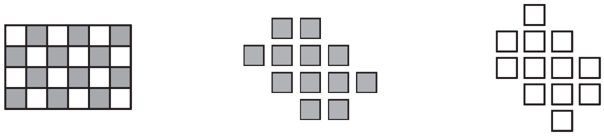

Fig. 10. Reduction of bishops domination to two rooks domination problems.

8. Proof of the bishops independence number for rectangular chessboards

As result of their experiments, in [25] the authors mention that for $m < n$ it is possible to put at most $m + n - 1$ non-attacking (independent) bishops on an $m \times n$ chessboard if m is odd or n is odd, and at most $m + n - 2$ non-attacking bishops otherwise. But [25] does not contain a proof, In this section we present a formal proof of this bishops independence number. In contrast to rooks, kings and knights, where, as mentioned in Section 6, proofs of the corresponding independence numbers can be obtained by modifications of the proofs of [26,28], for bishops new ideas and a more complex construction are necessary. Only the basic idea is the same: To prove that α is the independence number of a chess piece P, give an arrangement of α non-attacking copies of P on the chessboard and then show that an arrangement of more than α non-attacking copies is impossible.

In the case of bishops the first step is constructive and has been found with the help of RelView experiments by considering the cases $n = m + 1$, $n = m + 2$ and so forth. To enhance its presentation, we divide the construction into two parts and start with the following fact.

Proposition 8.1. *Assume an $m \times n$ chessboard to be given, where $0 < m < n \leq 2m$. Then it is possible to arrange on it $m+n-1$ non-attacking bishops if m is odd or n is odd, and $m + n - 2$ non-attacking bishops otherwise.*

Proof. We consider three cases. First, let m be odd and define $d := \frac{m+1}{2}$. Then $m + n - 1$ non-attacking bishops can be arranged in form of three groups as follows:

(a) Arrange m bishops along the chessboard's first column, that is, on the squares $\langle y, 1\rangle$, where $1 \leq y \leq m$.
(b) Arrange $n - m - 1$ bishops along row d of the chessboard starting with square $\langle d, d + 1\rangle$ and ending with square $\langle d, n - d\rangle$.
(c) Arrange m bishops along the chessboard's last column, that is, on the squares $\langle y, n\rangle$, where $1 \leq y \leq m$.

For $m = 7$ and $n = 12$ this arrangement is visualized in the left-hand picture of Fig. 11, where the black squares denote the positions of the 18 bishops. Notice, that for $n = m + 1$ the second group of bishops becomes empty, since in this case we get $n - d = m + 1 - \frac{m+1}{2} = \frac{m+1}{2} = d < d + 1$. That by the rule (b) on row d of the chessboard exactly $n - m - 1$ bishops stand as second group follows from the calculation

$$n - d - (d + 1) + 1 = n - 2d = n - 2\frac{m+1}{2} = n - m - 1.$$

By a case analysis it can be checked that all

$$m + (n - m - 1) + m = m + n - 1$$

bishops of the entire chessboard are in fact non-attacking: Since the three groups of bishops are placed on the same column and row, respectively, there are no attacks possible within the groups. Attacks between the first and the third group are impossible, since the range of coverage to the right of a bishop on column 1 ends with column m and $m < n$, and the range of coverage to the left of a bishop on column n ends with column $n - m + 1$ and $1 < n - m + 1$. Let P_d be the positive diagonal through $\langle d, d\rangle$ and N_d be the negative diagonal through $\langle d, d\rangle$. Then attacks between the first and the second group are impossible, since bishops of the first group only can attack pieces standing on or above P_d or on or below N_d and, conversely, bishops of the second group only can attack pieces standing below P_d or above N_d. Similar considerations show that attacks between the third and the second group are impossible.

Next, we assume that m is even and n is odd. We define $d := \frac{n+1}{2}$ and proceed as in the first case. The only difference is that we now use the middle column of the chessboard for the arrangement of the second group of bishops, i.e., replace the rule (b) as follows:

(b') Arrange $n - m - 1$ bishops along column d of the chessboard starting with square $\langle m - d + 2, d\rangle$ and ending with square $\langle d - 1, d\rangle$.

In the case $n = m + 1$ we get $d - 1 = \frac{m}{2} < \frac{m}{2} + 1 = m - d + 2$ and the second group again becomes empty. The two estimations $1 \leq m - d + 2$ and $d - 1 \leq m$ necessary for the placement of (b') follow from the assumption $n \leq 2m$. Below

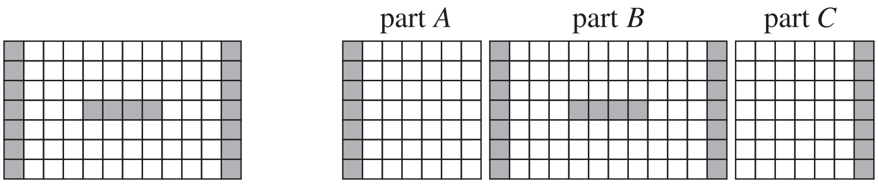

Fig. 11. Largest arrangements of non-attacking bishops.

is the verification of the first estimation; by a similar calculation the second estimation can be shown.

$$n \leq 2m \implies n-1 \leq 2m \iff \frac{n-1}{2} \leq m \iff \frac{n+1}{2} - 1 \leq m \iff d-1 \leq m \iff 1 \leq m-d+2$$

Again a simple case analysis shows that all bishops on the entire chessboard are non-attacking. Since the calculation

$$d-1-(m-d+2)+1 = 2d-m-2 = 2\frac{n+1}{2} - m - 2 = n+1-m-2 = n-m-1$$

proves that $n-m-1$ is in fact the number of bishops of the second group that, now vertically, is arranged on the chessboard's column d, we get the total number of non-attacking bishops on the entire chessboard as

$$m + (n-m-1) + m = m+n-1.$$

What remains is the case that m and n are even. Here we can arrange the second group of bishops either horizontally or vertically. If we use the first possibility, this means that we replace the above rule (b) as follows, using $d := \frac{m}{2}$ as number of the row the bishops are placed on.

(b") Arrange $n-m-2$ bishops along row d of the chessboard starting with square $\langle d, d+2\rangle$ and ending with square $\langle d, n-d-1\rangle$.

Here $n = m+2$ leads to an empty second group, due to $n-d-1 = \frac{m}{2}+1 < \frac{m}{2}+2 = d+2$. By the calculation

$$n-d-1-(d+2)+1 = n-2d-2 = n-2\frac{m}{2}-2 = n-m-2$$

we prove that there stand in fact $n-m-2$ bishops along row d. Hence, on the entire chessboard there are

$$m + (n-m-2) + m = m+n-2$$

bishops. Again their independence can be proved by a simple case analysis. □

Next, we show that the restriction $n \leq 2m$ in Proposition 8.1 is unnecessary, although we have applied it during the proof to show that the range of the placement (b') is correct. The proof of the following proposition can be seen as a recursive arrangement algorithm, with the arrangement described in the proof of Proposition 8.1 as the termination case.

Proposition 8.2. *Given an $m \times n$ chessboard with $0 < m < n$, it is possible to arrange on it $m+n-1$ non-attacking bishops if m is odd or n is odd, and $m+n-2$ non-attacking bishops otherwise.*

Proof. We use induction on $d := n-2m$. The induction base is $d \leq 1$. In this case we have $n \leq 2m$ and the claim follows from Proposition 8.1.

For the induction step, assume $d > 1$. Then it holds that $2m < n$. We divide the given chessboard into three parts. Part A consists of the m columns from 1 until m, part B of the $n-2m$ columns from $m+1$ until $n-m$, and part C of the m columns from $n-m+1$ until n. Since $m > 0$ implies $n-2m-2m < n-2m = d$ and $n-2m$ is odd iff n is odd, from the induction hypothesis we get for the $m \times (n-2m)$ chessboard given by part B that it is possible to arrange on it $m+n-2m-1$ non-attacking bishops if m is odd or n is odd, and $m+n-2m-2$ non-attacking bishops if m is even and n is even, respectively.

We complete these two arrangements in each case by m non-attacking bishops along the leftmost column of part A, i.e., the first column of the entire chessboard, and m non-attacking bishops along the rightmost column of part C, i.e., the last column of the entire chessboard. For $m = 7$ and $n = 26$ the result is visualized in the right-hand picture of Fig. 11, where part B equals the chessboard of the figure's left-hand picture. By a case analysis it is again easy to verify that all positioned bishops are non-attacking. Their number is

$$m + (m + n - 2m - 1) + m = m + n - 1$$

if m is odd or n is odd, and

$$m + (m + n - 2m - 2) + m = m + m - 2$$

if m is even and n is even. This concludes the induction step. $\square$

And here is the claimed result with regard to the independence of bishops on proper rectangular chessboards. Note that the assumption $m \neq n$ is necessary since for quadratic chessboards the largest number of non-attacking bishops is always $m + n - 2$, and does not depend on the fact that m and/or n are even or odd, respectively.

Proposition 8.3. *The largest number of non-attacking bishops that can be placed on an $m \times n$ chessboard, where $0 < m < n$, is $m + n - 1$ if m is odd or n is odd, and $m + n - 2$ if m is even and n is even.*

Proof. Assume that α denotes the largest number of non-attacking bishops standing on the chessboard. Since the chessboard has $m + n - 1$ positive diagonals and there can stand at most one non-attacking bishop on each diagonal, it holds that $\alpha \leq m + n - 1$.

If m is odd or n is odd, then Proposition 8.2 shows $m + n - 1 \leq \alpha$. In combination with $\alpha \leq m + n - 1$ this yields $m+n-1 = \alpha$. As we will show in a moment, if m and n are even it holds that $\alpha \leq m+n-2$. Hence, in this case Proposition 8.2 yields $\alpha = m + n - 2$.

We prove $\alpha \leq m+n-2$ for even m and n by contradiction. So, assume an arrangement of $m+n-1$ non-attacking bishops on the chessboard. Then there is exactly one bishop on each of the $m + n - 1$ positive diagonals. As already mentioned, we assume the lowermost-leftmost square $\langle 1, 1\rangle$ to be black. Since m and n are even, we have $\frac{m+n}{2} - 1$ black positive diagonals and $\frac{m+n}{2}$ white positive diagonals. Hence, $\frac{m+n}{2} - 1$ of the $m + n - 1$ bishops are on black squares and the remaining $\frac{m+n}{2}$ ones are on white squares. But there is also exactly one of the $m + n - 1$ non-attacking bishops on each of the $m + n - 1$ negative diagonals. In the given case we have $\frac{m+n}{2}$ black negative diagonals and $\frac{m+n}{2} - 1$ white negative diagonals. This leads to the contradiction that now $\frac{m+n}{2}$ of the $m+n-1$ bishops are on black squares and $\frac{m+n}{2} - 1$ are on white squares. $\square$

In [26,28] it is shown that, as already mentioned, for $m = n$ the largest number of non-attacking bishops is $2m - 2$. Furthermore, it is shown that all bishops of a largest independent arrangement have to be placed on the outer ring of squares. The later fact can be proved as follows. Let $2m - 2$ bishops stand on the $m \times m$ chessboard. Using V as symbol for the set of squares, we label each square $u \in V$ with $l_u := 1$ if a bishop stands on it, and with $l_u := a$ if it is empty and attacked by a bishops. The independence of the bishops and the maximality of the arrangement together imply $1 \leq l_u \leq 2$ for all $u \in V$. There are at least $2m$ squares u with $l_u = 1$, since $l_u = 1$ holds for the $2m - 2$ squares on which a bishop is placed and for the four corners and, furthermore, at least two of the corner squares are empty. This shows for the sum $S := \sum_{u \in V} l_u$ of all labels that

$$S \leq 2m + 2(m^2 - 2m) = (2m - 2)m,$$

since $2m$ labels are 1 and $m^2 - 2m$ labels are 1 or 2. Now, let o be the number of bishops on the outer ring and $i := 2m-2-o$ be the number of bishops in the interior. A bishop on the outer ring attacks precisely $m - 1$ squares and a bishop placed in the interior attacks at least $m + 1$ squares. Using this fact, we can calculate ad follows:

$$\begin{aligned} S &\geq o + o(m-1) + i + i(m+1) \\ &= om + i(m+2) \\ &= om + im + 2i \\ &= (o+i)m + 2i \\ &= (2m-2)m + 2i. \end{aligned}$$

Together with the above estimation $S \leq (2m - 2)m$ this result implies $S = S + 2i$ and, as a consequence of this equation, we get $i = 0$ as claimed.

Since in the quadratic case all $2m - 2$ non-attacking bishops stand on the outer ring of squares, the arrangement of the bishops on the, say, top row of the chessboard determines the arrangement of all $2m - 2$ non-attacking bishops. This follows from the fact that for all c with $1 \le c \le m$ a bishop on square $\langle m, c\rangle$ of the top row implies the attacked squares $\langle c, m\rangle$ (on the last column) and $\langle m - c + 1, 1\rangle$ (on the first column) to be empty and on the square $\langle 1, m - c + 1\rangle$ of the bottom row, hence, a bishop has to be placed because of the maximality of the arrangement. There are obviously precisely 2^m possible arrangements on the top row. Hence, there are all in all 2^m possible arrangements of the $2m - 2$ non-attacking bishops (compare again with the diagonal of the right-hand table of Fig. 5).

Our RelView experiments have shown that on proper rectangular chessboards largest arrangements of non-attacking bishops are possible such that some of the bishops are placed in the interior of the chessboard. Also the proof of Proposition 8.1 bases on such arrangements. As a consequence, in these cases the simple top row argument of [26,28] does not work anymore. Up to now, we have not found a general expression for the number of largest independent arrangements of bishops on proper rectangular chessboards. Only the case $m + 1 = n$ is simple. Here there exists exactly one arrangement for the $m + m + 1 - 1 = 2m$ non-attacking bishops. The rule is as follows: Place m of the bishops on column 1 and the remaining m bishops on column n. This arrangement is exactly that we have used in the proof of Proposition 8.1.

9. Concluding remarks

We have described a simple computing technique for solving independence and domination problems on rectangular chessboards. It bases on modeling, uses relation algebra as a methodological means (i.e., modeling language) and consists of the development of relation-algebraic specifications for certain independent and dominating sets of graphs and relations, respectively, and a representation of the chess pieces' graphs as relations. To evaluate the relation-algebraic specifications and to visualize the computed results we have used the specific purpose computer algebra system RelView. We have provided some of our experimental results concerning independence and domination and have used them to generalize many results of [26,28] for the quadratic case to the rectangular one. Based on the results on the independence of bishops, we have given a formal proof of the bishops independence number for proper rectangular chessboards.

In terms of the science of conceptual modeling theory (see e.g., [14]) we, strictly speaking, have used a three-stage vertical modeling process. In the first stage we have modeled chessboards by undirected graphs and reformulated the chessboard problems we are interested in as graph-theoretic problems in the commonly used terminology. In the next stage we have modeled undirected graphs via symmetric and irreflexive chessboard relations and formalized the graph-theoretic problems in the language of first-order logic. Finally, we have modeled the chessboard relations as well as the first-order problem specifications via expressions of relation algebra and, by that, the solutions of the original problems reduced to simple evaluations in an implementation of relation algebra.

There are still many open chessboard problems. We have mentioned some of them in the paper, e.g., to determine the number of largest independent arrangements of kings or the number of smallest dominating arrangements of knights. It also may lead to interesting combinatorial problems if one tries to count independent and dominating sets with a specified size. In [25] some connections between numbers of independent bishops and the Fibonacci numbers are conjectured, and also between numbers of independent bishops and a new series $(B_n)_{n>0}$ of natural numbers, called bishop numbers and inductively defined by $B_1 = B_2 = 1$ and $B_{n+2} = 2B_{n+1} + B_n$.

Because of the global approach that our approach takes, of course, it cannot compete for all intents and purposes with specifically tailored algorithms for chessboard problems – despite of the very efficient BDD-implementation of relations within RelView. An example for the latter is the FPGA-based approach to solve the n-queens problem for $n = 26$, yielding 22.317.699.616.364.044 possibilities (TU Dresden, 2009, for details see [17]). To give an impression of concrete RelView computing times, we present in the following some numbers for the n-queens problem. They have been obtained on an ordinary desktop PC with CPU AMD Phenom II X4 810, 2.6 GHz and 16 GB RAM, running Linux and using a RelView program that directly implements the relation-algebraic specification (7). In case of the 7×7 chessboard the tool computed 832 possibilities to place 6 independent queens, 40 possibilities to place 7 independent queens, and 0 possibilities to place 8 independent queens. The required computing times came to about 0.2 s in each case. For the 8×8 chessboard we obtained 3192 independent placements of size 7, 92 independent placement of size 8, and no independent placement of size 9. Here in each case approx. 4.3 s computing time were required. Finally, on the 9×9 chessboard RelView showed that 13848 independent placements of size 8 are possible, 352 independent placements of size 9 are possible, and again none of size 10 is possible; the corresponding RelView computing times were all about 16 s.

Following the lines of [25], it is easy to model chessboard problems as SAT solving problems using a Boolean variable for each square and propositional formulae to formulate the requirements. Without mentioning the used SAT solver, in [25] some computing times for the bishops independence problem are presented. On a small notebook for chessboard sizes up to 10×10 (i.e., 100 Boolean variables) the problem is solvable "in a couple of seconds". The required computing time for the 12×12 chessboard is 1136 s and can be reduced to 29 s using parallel computing with 64 processors. To solve the 12×12 problem with RelView, we have applied the reduction technique mentioned at the end of Section 6. The system required on the above mentioned PC about 1.6 s to solve each of the two sub-problems. With the same technique and on the same computer each sub-problem of the 13×16 chessboard mentioned at the end of Section 6, too, required approx. 900 s computing time. For the future we plan a detailed comparison of our approach with SAT solving and also with other modeling-oriented approaches to chessboard problems like integer programming and the use of declarative programming paradigms and programming languages. Especially the use of functional-logic programming languages seems to be promising. Hanus used a simple program in the Curry programming language (see [18]) to solve the n-queens problem on an ordinary PC for $n = 8, 10, 12, 14$ using 0.04, 0.4, 11, and 462 s computing time.

Science is a process for obtaining new insights and building new knowledge about the universe. In most cases knowledge is formulated as testable hypotheses. Nowadays, systematic, well-documented and reproducible experiments are accepted as a way for obtaining new insights, even in mathematics and computer science. As a consequence, tools for symbolic manipulation, prototypic computations, animation and visualization become increasingly important, especially if one proceeds in investigations. We hope to have provided insights in how a tool like RelView can be used in combinatorial reasoning. Our opinion is that the real attraction and general usefulness of our approach in this area lies in its flexibility, its large application area, the formal precision of the calculations and the concise form of the developed algorithms and, particular in view of RelView, the computational power of the tool as a result of the use of BDDs and the manifold animation and visualization possibilities.

Acknowledgement

I want to thank the unknown referees for carefully reading the paper and especially one of them for the valuable remarks and suggestions.

References

[1] J. Bell, B. Stevens, A survey of known results and research areas for n queens, Discrete Math. 309 (2008) 1–31.
[2] R. Berghammer, Applying relation algebra and RelView to solve problems on orders and lattices, Acta Inform. 45 (2008) 211–236.
[3] R. Berghammer, F. Neumann, RelView – an OBDD-based computer algebra system for relations, in: V.G. Gansha, E.W. Mayr, E. Vorozhtsov (Eds.), Computer Algebra in Scientific Computing, Lecture Notes in Computer Science, vol. 3718, Springer, 2006, pp. 40–51.
[4] R. Berghammer, G. Schmidt, The RelView-system, in: C. Choffrut, M. Jantzen (Eds.), Proc. 8th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, vol. 480, 1991, pp. 535–636.
[5] R. Berghammer, A. Rusinowska, H. de Swart, An interdisciplinary approach to coalition formation, Eur. J. Oper. Res. 195 (2009) 487–496.
[6] R. Berghammer, A. Rusinowska, H. de Swart, Applying relation algebra and RelView to measures in a social network, Eur. J. Oper. Res. 202 (2010) 182–198.
[7] R.E. Bryant, Graph-based algorithms for Boolean function manipulation, IEEE Trans. Comput. C-35 (8) (1986) 677–691.
[8] R.E. Bryant, Symbolic Boolean manipulation with ordered binary decision diagrams, ACM Comput. Surv. 24 (1992) 293–318.
[9] J.R. Burch, E.M. Clarke, K.L. McMillan, DE.I. Dill, L.I. Hwang, Symbolic model checking: 10^{20} states and beyond, in: Proc. Fifth Annual Symphosium on Logic in Computer Science, IEEE Computer Science (1990) 428–439.
[10] H.-C. Chen, T.-Y. Ho, The rook problem on saw-toothed chessboards, Appl. Math. Lett. 21 (2008) 1234–1237.
[11] E.J. Cockayne, Chessboard domination problems, Discrete Math. 86 (1990) 13–20.
[12] C.F. de Jaenisch, Applications de l'analyse mathematiques au jeu des echecs, Petrograd, 1882.
[13] J. Desharnais, Monomorphic characterization of n-ary direct products, Inform. Sci. 119 (1999) 275–288.
[14] D. Embley, B. Thalheim (Eds.), Handbook of Conceptual Modeling Theory, Springer, 2011.
[15] P.B. Gibbons, J.A. Webb, Some new results for the queens domination problem, Austral. J. Combin. 15 (1997) 145–160.
[16] http://www.informatik.uni-kiel.de/∼relview.shtml.
[17] http://queens.inf.tu-dresden.de.
[18] http://www-ps.informatik.uni-kiel.de/currywiki.
[19] B. Leoniuk, ROBDD-basierte Implementierung von Relationen und relationalen Operationen mit Anwendungen, Dissertation, Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel, 2001.
[20] U. Milanese, Zur Implementierung eines ROBDD-basierten Systems für die Manipulation und Visualisierung von Relationen. Dissertation, Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel, 2003.
[21] F. Nauck, Briefwechsel mit allen für alle, Illustrirte Zeitung 15 (1850) 182.
[22] E. Pauls, Das Maximalproblem der Damen auf dem Schachbrett, Deutsche Schachzeitung 29 (1874) 129–134.
[23] G. Schmidt, Relational mathematics, Encyclopedia of Mathematics and its Applications, vol. 132, Cambridge University Press, 2010.
[24] G. Schmidt, T. Ströhlein, Relations and Graphs, Discrete Mathematics for Computer Scientists, EATCS Monographs on Theoretical Computer Science, Springer, 1993.
[25] B. Steinbach, C. Posthoff, New results based on Boolean models, in: B. Steinbach (Ed.), Proc. 9th International Workshop on Boolean Problems, TU Bergakademie Freiberg, 2010, pp. 29–36.
[26] J.J. Watkins, Across the Board: The Mathematics of Chessboard Problems, Princeton University Press, 2004.
[27] I. Wegener, Branching Programs and Binary Decision Diagrams, Theory and Applications, SIAM Monographs on Discrete Mathematics and Applications, SIAM, 2000.
[28] A.M. Yaglom, I.M. Yaglom, Challenging Mathematical Problems with Elementary Solutions, Combinatorial Analysis and Probability Theory, vol. I, Holden-Day Inc., 1964.