

**CDMTCS  
Research  
Report  
Series**

**Boole-Bell-type Inequalities in  
*Mathematica***

**Stefan Filipp and Karl Svozil**  
Theoretische Physik, Technische  
Universität Wien, Viena Austria

CDMTCS-154  
May 2001

Centre for Discrete Mathematics and  
Theoretical Computer Science

# Boole-Bell-type inequalities in *Mathematica*

Stefan Filipp and Karl Svozil

Institut für Theoretische Physik, Technische Universität Wien  
Wiedner Hauptstraße 8-10/136, A-1040 Vienna, Austria  
e-mail: [svozil@tuwien.ac.at](mailto:svozil@tuwien.ac.at)

## Abstract

Classical Pitowsky correlation polytopes are reviewed with particular emphasis on the Minkowski-Weyl representation theorem. The inequalities representing the faces of polytopes are Boole's "conditions of possible experience." Many of these inequalities have been discussed in the context of Bell's inequalities. We introduce CddIF, a *Mathematica* package created as an interface between *Mathematica* and the `cdd` program by Komei Fukuda, which represents a highly efficient method to solve the hull problem for general classical correlation polytopes.

# Contents

<b>1</b>	<b>Boole-Bell Type Inequalities</b>	<b>3</b>
1.1	Simple urn model . . . . .	3
1.2	Geometrical interpretation . . . . .	3
1.3	Minkowski-Weyl representation theorem . . . . .	4
1.4	From vertices to inequalities . . . . .	5
1.5	From inequalities to vertices . . . . .	6
1.6	Quantum mechanical context . . . . .	6
<b>2</b>	<b>Installation</b>	<b>8</b>
2.1	<i>Mathematica</i> . . . . .	8
2.2	cdd . . . . .	8
2.3	Installation on Windows-platforms . . . . .	9
<b>3</b>	<b>Description Of Functions</b>	<b>9</b>
3.1	CddFormat . . . . .	9
3.2	ToCddExtFile . . . . .	10
3.3	TruthTable . . . . .	10
3.4	RunCdd . . . . .	11
3.5	ShowVRep . . . . .	12
3.6	ConvToHRep . . . . .	12
3.7	ReadInHRep . . . . .	13
3.8	GetInequFromHRep . . . . .	13
3.9	InequToRead . . . . .	14
3.10	GetViolInequalities . . . . .	14
3.11	PlotInequalities . . . . .	15
3.12	ContPlotInequalities . . . . .	17
3.13	Cdd . . . . .	19
<b>4</b>	<b>Examples</b>	<b>20</b>
4.1	Three particles and two measurement directions . . . . .	20
4.2	Violations of inequalities . . . . .	21
4.3	Graphical representation . . . . .	22
4.4	Two particles and three measurement directions . . . . .	23
4.5	Violations of inequalities . . . . .	24
4.6	Graphical representation . . . . .	26

# 1 Boole-Bell Type Inequalities And Their Geometric Representation

In the middle of the 19th century the English mathematician George Boole formulated a theory of "conditions of possible experience" [1, 2, 3, 4, 5]. These conditions are related to relative frequencies of logically connected events and are expressed by certain equations or inequalities. More recently, similar equations for a particular setup which are relevant in the quantum mechanical context have been discussed by Clauser and Horne and others [6, 7, 8]. Pitowsky has given a geometrical interpretation in terms of correlation polytopes [9, 4, 10, 5].

## 1.1 Simple urn model

Consider an urn containing some balls of different colors and styles: Each ball can be described by two properties, let us say "yellow" and "wooden", so each ball can have the property "yellow" or the property "wooden", but it can also have both "yellow and wooden". The state of the urn can be given by the probabilities to draw a ball with one of these properties:  $p_1$  is the proportion of yellow balls in the urn,  $p_2$  the proportion of wooden ones and  $p_{12}$  denotes the proportion of yellow and wooden balls. If there are enough balls in the urn these proportions are in fact the probabilities to get a ball with the special property by drawing. Clearly the inequalities

$$0 \leq p_{12} \leq p_2 \leq 1 \quad \text{and} \quad 0 \leq p_{12} \leq p_1 \leq 1 \quad (1)$$

are fulfilled by the proportions and so  $p_1$ ,  $p_2$  and  $p_{12}$  can be seen as probabilities of two events and their joint event only if these inequalities are satisfied. Simply by taking some appropriate numbers ( $p_1 = 0.6$ ,  $p_2 = 0.72$  and  $p_{12} = 0.32$ ) we can see, that equations (1) are not sufficient. If we take the probability to draw a ball which is either yellow or wooden ( $p_1 + p_2 - p_{12}$ ) into consideration, a new inequality can be found that is not satisfied by the numbers chosen:

$$0 \leq p_1 + p_2 - p_{12} \leq 1 \quad (2)$$

It can be shown that the inequalities (1) and (2) are necessary and sufficient for the numbers  $p_1$ ,  $p_2$  and  $p_{12}$  to represent probabilities of two events and their joint [4].

## 1.2 Geometrical interpretation

Itamar Pitowsky [9, 4, 10, 5] has suggested a geometric interpretation. Consider the truth table 1 of the above urn model, in which  $a_1$  and  $a_2$  represent the statements that "the ball drawn from the urn is yellow," "the ball drawn from the urn is wooden," and in which  $a_{12}$  represent the statement that "the ball drawn from the urn is yellow and wooden." The third "component bit" of the vector is a function of the first components. Actually, the function is a multiplication, since we are dealing with the classical logical "and" operation here. Let us take the set of all numbers ( $p_1$ ,  $p_2$ ,  $p_{12}$ ) satisfying the inequalities stated above as a set of vectors in a three-dimensional real space. This

$a_1$	$a_2$	$a_{12}$
0	0	0
1	0	0
0	1	0
1	1	1

Table 1: Truth table for two propositions  $a_1, a_2$  and their joint proposition  $a_{12} = a_1 \wedge a_2$

amounts to interpreting the rows of the truth table as vectors; the entries of the rows being the vector components. This procedure yields a closed convex polytope with vertices  $(0,0,0)$ ,  $(1,0,0)$ ,  $(0,1,0)$  and  $(1,1,1)$  (cf. Figure 1). The extreme points (vertices) can be interpreted as follows:

- $(0,0,0)$  is a case where no yellow and no wooden balls are in the urn at all,
- $(1,0,0)$  is representing the configuration that all balls are yellow and no one is wooden.
- $(0,1,0)$  is representing the configuration that all balls are wooden and no one is yellow.
- $(1,1,1)$  is a case with only yellow and at the same time wooden balls.

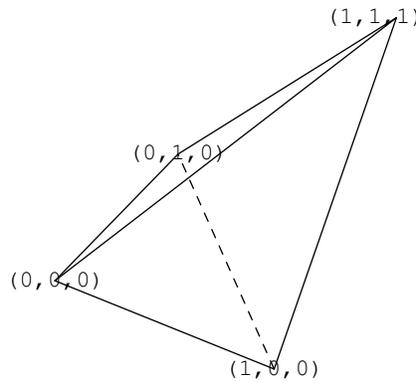


Figure 1: Polytope associated with the urn model

### 1.3 Minkowski-Weyl representation theorem

The Minkowski-Weyl representation theorem (e.g., [11, p. 29]) states that compact convex sets are “spanned” by their extreme points; and furthermore that the representation of this polytope by the inequalities corresponding to the planes of their faces is an equivalent one.

Stated differently, every convex polytope in an Euclidean space has a dual description: either as the convex hull of its vertices (V-representation), or as the intersection of

a finite number of half-spaces, each one given by a linear inequality (H-representation) This equivalence is known as the *Weyl-Minkowski* theorem.

The problem to obtain all inequalities from the vertices of a convex polytope is known as the *hull problem*. One solution strategy is the Double Description Method [12] which we shall use but not review here.

## 1.4 From vertices to inequalities

For the above simple urn model, the inequalities are rather intuitive and can be easily obtained by guessing. This is impossible in the general case involving more events and more joint probabilities thereof. In order to obtain the relevant inequalities—Boole’s “conditions of possible experience”—we have to find a hopefully constructive way to derive them.

Recall that a vector is an element of the polytope if and only if it can be represented as a certain bounded convex combination, i.e., a bounded linear span, of the vertices. More precisely, let us denote the *convex hull*  $\text{conv}(K)$  of a finite set of points  $K = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in R^d$  by

$$\text{conv}(K) = \left\{ \lambda_1 \mathbf{x}_1 + \dots + \lambda_n \mathbf{x}_n \mid \mathbf{n} \geq \mathbf{1}, \lambda_i \geq 0, \sum_{i=1}^n \lambda_i = \mathbf{1} \right\}. \quad (3)$$

In the probabilistic context, the coefficients  $\lambda_i$  are interpreted as the probability that the event represented by the extreme point  $\mathbf{x}_i$  occurs, whereby  $K$  represents the complete set of all atoms of a Boolean algebra. The geometric interpretation of  $K$  is the set of all extreme points of the correlation polytope.

In summary, the connection between the convex hull of the extreme points of a correlation polytope and the inequalities representing its faces is guaranteed by the Minkowski-Weyl representation theorem. A constructive solution of the corresponding hull problem exists (but is NP-hard [10]).

For the special urn model introduced above this means that any three numbers ( $p_1$ ,  $p_2$  and  $p_{12}$ ) must fulfill an equation dictated by Kolmogorov’s probability axioms [13]:

$$(p_1, p_2, p_{12}) = \lambda_1(0, 0, 0) + \lambda_2(0, 1, 0) + \lambda_3(1, 0, 0) + \lambda_4(1, 1, 1) = (\lambda_2 + \lambda_4, \lambda_3 + \lambda_4, \lambda_4). \quad (4)$$

It is important to realize that these logical possibilities are exhaustive. By definition, there cannot be any other classical case which is not already included in the above possibilities  $(0, 0, 0), (1, 0, 0), (0, 1, 0), (1, 1, 1)$ . Indeed, if one or more cases would be omitted, the corresponding polytope would not be optimal; i.e., it would be embedded in the optimal one. Therefore, any “state” of a physical system represented by a probability distribution has to satisfy the constraint

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1. \quad (5)$$

The four extreme cases  $\lambda_i = 1, \lambda_j = 0$  for  $i \in \{1, 2, 3, 4\}$  and  $j \neq i$  just correspond to the vertices spanning the classical correlation polytope as the convex sum (3).

A generalization to arbitrary configurations is straightforward. To solve the hull problem for more general cases, an efficient algorithm has to be used. There are some

algorithms to solve this problem, but they run in exponential time in the number of events, thus it can be solved only for small enough cases to get a solution in conceivable time.

### 1.5 From inequalities to vertices

Conversely, a vector is an element of the convex polytope if and only if its coordinates satisfy a set of linear inequalities which represent the supporting hyper-planes of that polytope. The problem to find the extreme points (vertices) of the polytope from the set of inequalities is dual to the hull problem considered above.

### 1.6 Quantum mechanical context

In the quantum mechanical case the elementary irreducible events are clicks in particle detectors and the probabilities have to be calculated using the formalism of quantum mechanics. It is by no means trivial that these probabilities satisfy Eq. (5), in particular if one realizes that quantum Hilbert lattices are nonboolean and have an infinite number of atoms. As it turns out, Boole’s “conditions of possible experience” are violated if one considers probabilities associated with complementary events, thereby assuming counterfactuality. (This is a development and a generalization Boole could have hardly foreseen!)

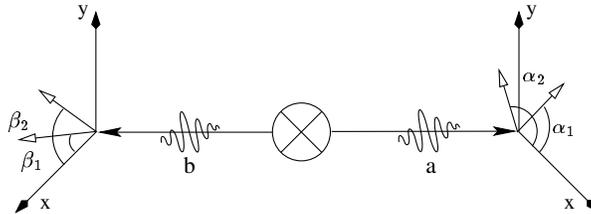


Figure 2: Experimental setting to test the violation of Boole - Bell type inequalities

As an example we take a source that produces pairs of spin- $\frac{1}{2}$  particles in a singlet-state ( $|\psi\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$ ). The particles fly apart along the z axis and after the particles have separated, measurements on spin components along one out of two directions are made. If, for simplicity, the measurements are made in the x-y plane perpendicular to the trajectory of the particles, the direction of the measurement can be given by angles measured from the vertical x axis ( $\alpha_1$  and  $\alpha_2$  on the one side,  $\beta_1$  and  $\beta_2$  on the other side). On each side the measurement angle is chosen randomly for each pair of incoming particles and each measurement can yield two results - in  $\frac{\hbar}{2}$  units: “+1” for spin up and “-1” for spin down (cf. Figure 2).

Deploying this configuration we get probabilities to find a particle measured along the axis specified by the angles  $\alpha_1$ ,  $\alpha_2$ ,  $\beta_1$  and  $\beta_2$  either in spin up or in spin down state denoted as  $p_{a1}$ ,  $p_{a2}$ ,  $p_{b1}$ ,  $p_{b2}$  - and we also take the joint event of finding a particle on one side at the angle  $\alpha_1$  ( $\alpha_2$ ) in a specific spin state and the other particle on the other

side along the vector  $\beta_1$  ( $\beta_2$ ) in a specific spin state, denoted as  $p_{a1b1}$ ,  $p_{a2b1}$ ,  $p_{a1b2}$  and  $p_{a2b2}$ . To construct the convex polytope to this experiment we build up a truth table of all possible events using a “1” as “spin up is detected along the specific axis” and a “0” as “spin down is detected along the specific axis” (table 2). The rows of this table

$\alpha_1$	$\alpha_2$	$\beta_1$	$\beta_1$	$\alpha_1\beta_1$	$\alpha_1\beta_2$	$\alpha_2\beta_1$	$\alpha_2\beta_2$
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
1	0	1	0	1	0	0	0
0	1	1	0	0	0	1	0
1	1	1	0	1	0	1	0
0	0	0	1	0	0	0	0
1	0	0	1	0	1	0	0
0	1	0	1	0	0	0	1
1	1	0	1	0	1	0	1
0	0	1	1	0	0	0	0
1	0	1	1	1	1	0	0
0	1	1	1	0	0	1	1
1	1	1	1	1	1	1	1

Table 2: Truth table for four propositions

are then identified with the vertices of the convex polytope. By using the Minkowski-Weyl theorem and by solving the hull problem, the vertices determine the hyper-planes confining the polytope, i.e. the inequalities which the probabilities have to satisfy. As a result the following inequalities are gained:

$$\begin{aligned} 0 \leq p_{aibi} \leq p_{ai} \leq 1, 0 \leq p_{aibi} \leq p_{bi} \leq 1 \quad & i = 1, 2 \\ p_{ai} + p_{bi} - p_{aibi} \leq 1 \quad & i = 1, 2 \end{aligned} \quad (6)$$

$$\begin{aligned} -1 \leq p_{a1b1} + p_{a1b2} + p_{a2b2} - p_{a2b1} - p_{a1} - p_{b2} \leq 0 \\ -1 \leq p_{a2b1} + p_{a2b2} + p_{a1b2} - p_{a1b1} - p_{a2} - p_{b2} \leq 0 \\ -1 \leq p_{a1b2} + p_{a1b1} + p_{a2b1} - p_{a2b2} - p_{a1} - p_{b1} \leq 0 \\ -1 \leq p_{a2b2} + p_{a2b1} + p_{a1b1} - p_{a1b2} - p_{a2} - p_{b1} \leq 0 \end{aligned} \quad (7)$$

The last four inequalities are known as *Clauser-Horne inequalities*. As noticed above the probabilities have to be seen in a quantum mechanical context. If we consider the singlet state of spin- $\frac{1}{2}$  particles  $|\Psi\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$  it is well known that the probability to find the particles both either in spin up or in spin down states is given by  $P^{\uparrow\uparrow}(\theta) = P^{\downarrow\downarrow}(\theta) = \frac{1}{2}\sin^2(\theta/2)$  - where  $\theta$  is the angle between the measurement directions. The single event probability is clearly  $p_i = \frac{1}{2}$ . By choosing

$$a_1 = -\frac{\pi}{3} \quad a_2 = b_1 = \frac{\pi}{3} \quad b_2 = \frac{\pi}{3} \quad (8)$$

as measurement directions, we get for  $p = (p_{a1}, p_{a2}, p_{b1}, p_{b2}, p_{a1b1}, p_{a2b1}, p_{a1b2}, p_{a2b2})$ :

$$p = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{3}{8}, \frac{3}{8}, 0, \frac{3}{8}\right) \quad (9)$$

and one of the inequalities found in (7) is violated:

$$p_{a1b1} + p_{a1b2} + p_{a2b2} - p_{a2b1} - p_{a1} - p_{b2} = \frac{3}{8} + \frac{3}{8} + \frac{3}{8} - 0 - \frac{1}{2} - \frac{1}{2} = \frac{1}{8} > 0 \quad (10)$$

The generalization is straightforward. Violations of certain inequalities involving classical probabilities—Boole’s “conditions of possible experience” [2]—also appear in higher dimensions in configurations containing more particles and/or more measurement directions. We shall consider more examples below.

## 2 Installation

### 2.1 *Mathematica*

All functions described in the following section can be found in the *Mathematica*-package *cddif.m*. In general this package has to be loaded into the current *Mathematica*-kernel by the command `<<'path to cddif.m'/cddif.m, short description and usage of the functions is available by entering ?' <function>'`.

To guarantee a proper run of all functions it is necessary (and hopefully sufficient) that `cdd` is located in any directory listed in the `PATH`-variable (usually `/bin`, `/usr/bin`, `/usr/local/bin`, ...) <sup>1</sup> or in the current working directory, which can be shown by evaluating `Directory []` or changed using the function `SetDirectory [directory.String]`. If one likes to avoid the frequent use of this function one can append this command to the package-file *cddif.m* before the line `End []` so that on each loading of the package the directory is set automatically to a personal working directory.

### 2.2 `cdd`

`cdd` is a C++ (ANSI C) implementation of the Double Description Method [12] by Komei Fukuda[14]. It generates all vertices (i.e. extreme points) and extreme rays of a general convex polyhedron given by a system of linear inequalities. Conversely, it solves the hull problem by generating a system of linear inequalities given all vertices.

At this point we refer to the documentation of the program for the installation of the `cdd` - package, in particular to the file *cdd.readme* included in the package and to the online documentation at <http://www.ifor.math.ethz.ch/~fukuda/cddman/cddman.html>. `cdd` is available for free and one can download it from the homepage of Komei Fukuda[14] ([http://www.ifor.math.ethz.ch/~fukuda/cdd\\_home/cdd.html](http://www.ifor.math.ethz.ch/~fukuda/cdd_home/cdd.html)), where one can also find a manual to the usage of `cdd`, especially descriptions to the format of the input- and output-files and to options that can be passed to `cdd`.

<sup>1</sup>for setting environment variables look at the manual of *set* or *env*

## 2.3 Installation on Windows-platforms

Currently a version of `cdd` executable on Windows platforms can be downloaded from <http://tph.tuwien.ac.at/~svozil/cdd/cdd.exe> (a different compilation is at <http://www.wis.kuleuven.ac.be/wis/algebra/kathleen/files/cdd061.exe>).

On Windows-systems *Mathematica* must be able to find `cdd` in a directory listed in the `PATH`-variable or in the current working directory, too. To set the `PATH`-variable in Windows 2000/NT go to the “control panel” and click on the “system properties”, then click “advanced” and there is a place where the variable `PATH` is specified. Here one can add the path to `cdd` (separated by a semicolon). In WindowsME one needs to go execute “`msconfig`” to get to the System Configuration Utility - in “Environment” one can edit the `PATH`-variable and for Windows98/95 one must edit the file “*autoexec.bat*” to get the path set.

Finally it can be necessary to rename the `cdd` - executable file (e.g. from *cdd061.exe*) to *cdd.exe*. The `CddIF`-Package uses `cdd` as default command to run `cdd`, using the function `SetCddCmd [cmd_String]` one can change this behavior. Like already stated above one can also add this commandline to the package-file *cddif.m* just before the `End []`-statement to change the default command automatically when loading the package.

## 3 Description Of Functions

In this section all functions of the *CddIF* - package are listed. For each function the syntax including the necessary parameters (if parameters are optional, it has an “*opt*.” as prefix), a description and an example is given.

### 3.1 CddFormat

`CddFormat [vertices, opt: options]`

*vertices (List)*: List of  $m$  vertices in  $n$  dimensions of the form

$$\{\{x_{11}, x_{12}, \dots, x_{1n}\}, \{x_{21}, \dots, x_{2n}\}, \dots, \{x_{m1}, x_{m2}, \dots, x_{mn}\}\}$$

*options (List)*: Options to `cdd` (e. g. *adjacency*, *nondegenerate*, *minindex*,...) - see documentation to `cdd` (<http://www.ifor.math.ethz.ch/fukuda/cddman/cddman.html>)

**Description:** A list of vertices, which can be determined for example by `TruthTable [..., IncludeVars→False]`, are converted to a format recognized by `cdd`. Additionally options to `cdd` can be declared.

**Example:**

```
In[]= CddFormat [{{1,0,0},{0,1,0},{1,1,1}}]
```

```
Out[]= {V-representation begin,{3, 4, integer},
        {1, 1, 0, 0}, {1, 0, 1, 0}, {1,1, 1, 1},
        end}
```

### 3.2 ToCddExtFile

ToCddExtFile [ *file*, *vertices*, *opt: options*] or  
ToCddExtFile [ *file*, *particles*, *measurements*, *opt: options*]

*file* (*String*): Filename for output of H-representation (“*.ext*”-suffix is automatically appended)

*vertices* (*List*): List of *m* vertices in *n* dimensions of the form

$$\{\{x_{11}, x_{12}, \dots, x_{1n}\}, \{x_{21}, \dots, x_{2n}\}, \dots, \{x_{m1}, x_{m2}, \dots, x_{mn}\}\}$$

*options* (*List*): Options to `cdd` (e. g. *adjacency*, *nondegenerate*, *minindex*,...) - see documentation to `cdd` (<http://www.ifor.math.ethz.ch/fukuda/cddman/cddman.html>)

*particles* (*Integer*): Number of particles

*measurements* (*Integer*): Number of possible measurements to each particle (equivalent to number of detection angles)

**Description:** Creates a file with “*.ext*”-extension that contains the data of the given configuration to use in `cdd`. Either a list of vertices of the considered correlation polytop or the number of particles used and the possible measurements to each can be handed over. In the latter case the list of vertices is generated automatically.

#### Example:

```
In []= ToCddExtFile [ "test",2,3]
```

```
Out []= test.ext
```

writes the file “*test.ext*” to the current working directory, containing the vertices of the 2-particles 3-measurements configuration.

### 3.3 TruthTable

TruthTable [ *particles*, *measurements*, *opt: options*]

*particles* (*Integer*): Number of particles

*measurements* (*Integer*): Number of possible measurements to each particle (equivalent to number of detection angles)

*opt: options*: The only possible option is `IncludeVars`  $\rightarrow$  `True/False`. If `IncludeVars`  $\rightarrow$  `True` is defined, the function includes a list of variables belonging to the given configuration as titles of the columns and output will be made in `MatrixForm`, otherwise a list containing all vertices is returned. Default is `IncludeVars`  $\rightarrow$  `True`.

**Description:** Creates a truth table of the given configuration, containing all vertices of the corresponding correlation polytopes. For generating this table all possible single events are rated either 0 or 1 (i. e. true or false) and the joint events are evaluated using the logical *AND* operation.

**Example:**

In[]= TruthTable [2,2]

Out[]=

a1	a2	b1	b2	a1b1	a1b2	a2b1	a2b2
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
1	0	1	0	1	0	0	0
0	1	1	0	0	0	1	0
1	1	1	0	1	0	1	0
0	0	0	1	0	0	0	0
1	0	0	1	0	1	0	0
0	1	0	1	0	0	0	1
1	1	0	1	0	1	0	1
0	0	1	1	0	0	0	0
1	0	1	1	1	1	0	0
0	1	1	1	0	0	1	1
1	1	1	1	1	1	1	1

### 3.4 RunCdd

RunCdd [file]

*file* (String): File handed over to cdd as command parameter (automatically extended with “.ext.”-suffix).

**Description:** RunCdd [...] executes cdd (with file as parameter) and returns the corresponding H-representation, which can be used in various other functions like PlotInequalities [...] or GetViolInequalities [...].

Using this function one has to pay attention to the potentially long runtime in calculating the faces (i. e. the inequalities) of the correlation polytope. It can be more beneficial to use ToCddExtFile [file,...] to create a “.ext”-file, followed by executing cdd outside of Mathematica (eventually on a faster computer) to convert the data to H-representation stored in an “.ine”-file. Afterwards one can read in this file utilizing ReadInHRep [file]

**Example:**

```
In[]= RunCdd [ "test" ]
```

```
Out[]= {{H-representation},{begin},{684,16,real},{2,0,-2,...},...,{end}},
        whereas 2-particles 3-measurement configuration is taken into consideration
        here.
```

**3.5 ShowVRep**

```
ShowVRep [ particles, measurements ]
```

*particles (Integer):* Number of particles

*measurements (Integer):* Number of possible measurements to each particle (equivalent to number of detection angles)

**Description:** Shows the V-representation of a given configuration.

**Example:**

```
In[]= ShowVRep [ 2,2 ]
```

```
Out[]= {V-representation
        begin,
        {16, 9, integer},
        {1, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {1, 1, 0, 0, 0, 0, 0, 0, 0, 0},
        {1, 0, 1, 0, 0, 0, 0, 0, 0, 0}, {1, 1, 1, 0, 0, 0, 0, 0, 0, 0},
        {1, 0, 0, 1, 0, 0, 0, 0, 0, 0}, {1, 1, 0, 1, 0, 1, 0, 0, 0, 0},
        {1, 0, 1, 1, 0, 0, 0, 1, 0, 0}, {1, 1, 1, 1, 0, 1, 0, 1, 0, 0},
        {1, 0, 0, 0, 1, 0, 0, 0, 0, 0}, {1, 1, 0, 0, 1, 0, 1, 0, 0, 0},
        {1, 0, 1, 0, 1, 0, 0, 0, 1, 0}, {1, 1, 1, 0, 1, 0, 1, 0, 1, 0},
        {1, 0, 0, 1, 1, 0, 0, 0, 0, 0}, {1, 1, 0, 1, 1, 1, 1, 0, 0, 0},
        {1, 0, 1, 1, 1, 0, 0, 1, 1, 0}, {1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        end}
```

**3.6 ConvToHRep**

```
ConvToHRep [ particles, measurements, opt: file, opt: options ] or
```

```
ConvToHRep [ vertices, opt: file, opt: options ]
```

*particles (Integer):* Number of particles

*measurements (Integer):* Number of possible measurements to each particle (equivalent to number of detection angles)

*vertices (List):* List of m vertices in n dimensions of the form

$$\{\{x_{11}, x_{12}, \dots, x_{1n}\}, \{x_{21}, \dots, x_{2n}\}, \dots, \{x_{m1}, x_{m2}, \dots, x_{mn}\}\}$$

*file (String)*: Filename that is used for the conversion from a “.ext”-file to a “.ine”-file which is equivalent to a conversion from V-representation to H-representation. Default is “tmp”.

*options (List)*: Options to `cdd` (e. g. *adjacency*, *nondegenerate*, *minindex*,...) - see documentation to `cdd`(<http://www.ifor.math.ethz.ch/fukuda/cddman/cddman.html>)

**Description:** This function converts a given configuration (n particles, m measurements) or a given list of vertices from V-representation into a H-representation. As above in (3.4) the potentially long calculation time has to be taken into consideration, depending on the complexity of the problem.

**Example:**

```
In[] = ConvToHRep [2,3,"2_3"]
```

```
Out[] = {{H-representation},{begin},{684, 16,real},{2, 0,...}}...,{end}}
```

whereas in this case the files “2\_3.ext” (created by *Mathematica* containing the data for `cdd`) and “2\_3.ine” (created by `cdd` as result of the calculation) are generated in the current working directory.

### 3.7 ReadInHRep

```
ReadInHRep [file]
```

*file (String)*: “.ine” - file containing the H-representation which is to be read in.

**Description:** Reads the H-representation from a given “.ine”-file for further use in various functions like `GetViolInequalities [...]` or `PlotInequalities [...]`.

**Example:**

```
In[] = ReadInHRep ["2_3"]
```

```
Out[] = {{H-representation},{begin},{684,16,real},{2,0,...}}...,{end}}
```

### 3.8 GetInequFromHRep

```
GetInequFromHRep [hrep]
```

*hrep (List)*: H-representation yielded for example from the function `ReadInHRep [...]` or `ConvToHRep [...]`.

**Description:** Returns the inequalities from a given H-representation as a list. To make the list more readable one can apply `InequToRead [...]` on it.

**Example:**

```
In[]= GetInequFromHRep [ConvToHRep [2,1] ]
Out[]= {{a1 - a1b1 + b1, 1}, {-a1 + a1b1, 0}, {a1b1 - b1, 0},
        {-a1b1, 0}}
```

**3.9 InequToRead**

```
InequToRead [ inequalities ]
```

*inequities (List):* List of inequalities yielded from `GetInequFromHRep [...]`

**Description:** Makes the list of inequalities yielded from `GetInequFromHRep [...]` more readable.

**Example:**

```
In[]= InequToRead [GetInequFromHRep [ConvToHRep [2,1] ] ]
Out[]= a1 - a1b1 + b1 ≤ 1
        -a1 + a1b1 ≤ 0
        a1b1 - b1 ≤ 0
        -a1b1 ≤ 0
```

**3.10 GetViolInequalities**

```
GetViolInequalities [ hrep, angles, functions, inequ-nr, violation ] or
GetViolInequalities [ file, angles, functions, inequ-nr, violation, opt: options ]
```

*hrep (List):* H-representation yielded for example from the function `ReadInHRep [...]` or `ConvToHRep [...]`.

*file (String):* File containing the demanded H-representation in `cdd` format.

*angles (List):* List of measurement angles for each particle, whereas the dimension of the list must represent the configuration. If one chooses the configuration “2 particles - 2 measurements” the list must have the dimension `[2,2]`, in this case the particles `a` and `b` are measured each along two axis given by the angles  $a_1$ ,  $a_2$ ,  $b_1$  and  $b_2$ , so this parameter has the form  $\{\{a_1, a_2\}, \{b_1, b_2\}\}$ .

*functions (Symbol):* Functions to calculate the quantum mechanical probability of the events. Considering for example two spin- $\frac{1}{2}$  particles in a singlett state, the probability to find the particles both either in spin “up” or both in spin “down” states is given by  $\mathbf{P}^{\uparrow\uparrow}(\theta, \phi) = \mathbf{P}^{\downarrow\downarrow}(\theta, \phi) = \frac{1}{2} \sin^2[\frac{\theta-\phi}{2}]$ , where  $\theta$  and  $\phi$  are the measurement angles of the particles.

In defining these functions one has to notice, that for all possible events (single events, two-particle-events,...) an appropriate function definition has to exist, each taking a list as parameter (e.g.  $\text{Prob}[\{x-, y-\}] = \frac{1}{2} \text{Sin}[(x - y) / 2]^2$ ).

*inequ-nr (List)*: Range of rows in H-representation used for checking violated inequalities. Specifying this can be useful, if many inequalities have to be evaluated. The form of the parameter is  $\{\text{min}, \text{max}\}$  oder All.

*violation (Real)*: Only inequalities are printed out, that are violated more than this parameter. Default value is 0.

*opt: options*: Options can be `PrintOut`  $\rightarrow$  `True/False`, which specifies, if the inequalities are printed out during evaluation or not.

**Description:** `GetViolInequalities [...]` calculates the discrepancy of inequalities using the given probability functions and therefore the quantum mechanical violation for a distinct adjustment (i. e. special angles) of the detectors and returns all violated inequalities.

**Example:**

```
In[]= GetViolInequalities [ConvToHRep [2,2], {{-π/6,0},{0,π/6}}, Prob]
```

```
Out[]= (-a1b1+a1b2+a2-a2b1-a2b2+b1 ≤ 1  $\frac{9}{8}$ )
```

The probability for the single event of measuring one particle in spin “up” or spin “down” at any angle is given by  $\text{Prob}[\{x_{-}\}] = \frac{1}{2}$  and the probability of the joint event has been calculated by  $\text{Prob}[\{x_{-}, y_{-}\}] = \frac{1}{2} \text{Sin}[(x - y)/2]^2$ .

### 3.11 PlotInequalities

`PlotInequalities [hrep, range, angles, functions, opt: options]` or

`PlotInequalities [file, range, angles, functions, inequ-nr, violation, opt: options]`

*hrep (List)*: H-representation yielded for example from the function `ReadInHRep [...]` or `ConvToHRep [...]`.

*file (String)*: File containing the demanded H-representation in `cdd` format.

*range (List)*: Parameter specifying the range for the variable x, which is plotted on the horizontal axis. It has the form  $\{x, x_{min}, x_{max}\}$  (see *Mathematica* function `Plot [...]`)

*angles (List)*: List of measurement angles for each particle, whereas the dimension of the list must represent the configuration. If one chooses the configuration “2 particles - 2 measurements” the list must have the dimension  $[2, 2]$ , in this case the particles a and b are measured each along two axis given by the angles  $a_1, a_2, b_1$  and  $b_2$ , so this parameter has the form  $\{\{a_1, a_2\}, \{b_1, b_2\}\}$ .

*functions (Symbol)*: Functions to calculate the quantum mechanical probability of the events. Considering for example two spin- $\frac{1}{2}$  particles in a singlett state, the probability to find the particles both either in spin “up” or both in spin “down” states

is given by  $\mathbf{P}^{\uparrow\uparrow}(\theta, \phi) = \mathbf{P}^{\downarrow\downarrow}(\theta, \phi) = \frac{1}{2} \sin^2[\frac{(\theta-\phi)}{2}]$ , where  $\theta$  and  $\phi$  are the measurement angles of the particles.

In defining these functions one has to notice, that for all possible events (single events, two-particle-events, ...) an appropriate function definition has to exist, each taking a list as parameter (e.g.  $\text{Prob}[\{x_-, y_-\}] = \frac{1}{2} \text{Sin}[(x-y)/2]^2$ ).

*inequ-nr (List)*: Range of rows in H-representation used for checking violated inequalities. Specifying this can be useful, if many inequalities have to be evaluated. The form of the parameter is  $\{\text{min}, \text{max}\}$  oder `All`.

*violation (Real)*: Only inequalities are printed out, that are violated more than this parameter. Default value is 0.

*opt: options*: Options for the *Mathematica*-function `Plot [...]` can be handed over.

**Description:** This function yields a plot of (violated) inequalities, whereas the function plotted is  $f(x) = p(x) - b$  derived from the inequalities of the form  $p(x) \leq b$  ( $p(x)$  is a linear combination of functions to calculate the probabilities of (joint) events, dependent on one variable). Consequently the degree of violation is represented as a positive value of  $f(x)$ .

Take for example the case “2 particles and 2 measurement directions”, where the inequality

$$-p_{a_1 b_1} + p_{a_1 b_2} + p_{a_2} - p_{a_2 b_1} - p_{a_2 b_2} + p_{b_1} \leq 1$$

appears. The probability for the single event of measuring one particle in spin “up” or spin “down” at any angle is given by

$$p_{a_1}(x) = p_{a_2}(x) = p_{b_1}(x) = p_{b_2}(x) = \frac{1}{2}$$

and the probability of the joint event can be calculated by

$$p_{a_1 b_1}(x, y) = p_{a_2 b_1}(x, y) = p_{a_1 b_2}(x, y) = p_{a_2 b_2}(x, y) = \frac{1}{2} \sin^2[(x-y)/2].$$

If we define the measurement angles by

$$a_1 = -\frac{\pi}{3} + x \quad a_2 = b_1 = 0 \quad b_2 = 2 * \pi$$

the inequality can be written as

$$1 + \frac{1}{2} \sin \frac{1}{2}(-\frac{\pi}{3} - x) - \frac{\sin x}{2} - \frac{1}{2} \sin \frac{1}{2}(-\frac{\pi}{3} + x) \leq 1$$

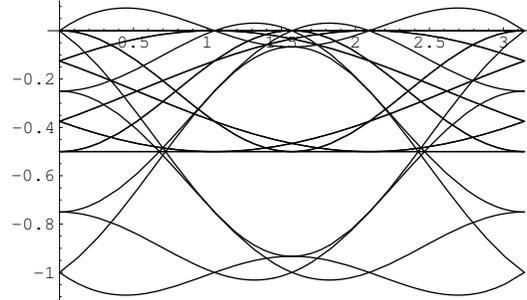
The left side is dependent on  $x$  ( $p(x) = 1 + \frac{1}{2} \sin \frac{1}{2}(-\frac{\pi}{3} - x) - \frac{\sin x}{2} - \frac{1}{2} \sin \frac{1}{2}(-\frac{\pi}{3} + x)$ ) and  $b = 1$ . The function to be plotted is  $f(x) = p(x) - b$ , thus

$$f(x) = 1 + \frac{1}{2} \sin \frac{1}{2}(-\frac{\pi}{3} - x) - \frac{\sin x}{2} - \frac{1}{2} \sin \frac{1}{2}(-\frac{\pi}{3} + x) - 1$$

**Example:**

```
In[]= PlotInequalities [ConvToHRep [2,2] ,{x,0,π},{{-π/3,+x,0},{0,2x}},Prob]
```

```
Out []=
```



The functions to calculate the probabilities have been defined as  $\text{Prob}[\{x_{-}\}] := \frac{1}{2}$  for a single event and  $\text{Prob}[\{x_{-}, y_{-}\}] := \frac{1}{2} \text{Sin}[(x - y)/2]^2$  for two-particle events.

### 3.12 ContPlotInequalities

`ContPlotInequalities [hrep, rangex, rangey, angles, func, ineqnr, violation, opt: options]` or

`ContPlotInequalities [file, rangex, rangey, angles, func, ineqnr, violation, opt: options]`

*hrep (List):* H-representation yielded for example from the function `ReadInHRep [...]` or `ConvToHRep [...]`.

*rangex (List):* Parameter specifying the range for the variable x, which is plotted on the horizontal axis. It has the form  $\{x, x_{min}, x_{max}\}$  (see *Mathematica* function `Plot [...]`)

*rangey (List):* Parameter specifying the range for the variable y, which is plotted on the vertical axis. It has the form  $\{y, y_{min}, y_{max}\}$  (see *Mathematica* function `Plot [...]`)

*angles (List):* List of measurement angles for each particle, whereas the dimension of the list must represent the configuration. If one chooses the configuration “2 particles - 2 measurements” the list must have the dimension  $[2, 2]$ , in this case the particles a and b are measured each along two axis given by the angles  $a_1, a_2, b_1$  and  $b_2$ , so this parameter has the form  $\{\{a_1, a_2\}, \{b_1, b_2\}\}$ .

*functions (Symbol):* Functions to calculate the quantum mechanical probability of the events. Considering for example two spin- $\frac{1}{2}$  particles in a singlet state, the probability to find the particles both either in spin “up” or both in spin “down” states is given by  $\mathbf{P}^{\uparrow\uparrow}(\theta, \phi) = \mathbf{P}^{\downarrow\downarrow}(\theta, \phi) = \frac{1}{2} \sin^2\left[\frac{\theta-\phi}{2}\right]$ , where  $\theta$  and  $\phi$  are the measurement angles of the particles.

In defining these functions one has to notice, that for all possible events (single events, two-particle-events, ...) an appropriate function definition has to exist, each taking a list as parameter (e.g.  $\text{Prob}[\{x_-, y_-\}] = \frac{1}{2} \text{Sin}[(x - y)/2]^2$ ).

*inequ-nr (List):* Range of rows in H-representation used for checking violated inequalities. Specifying this can be useful, if many inequalities have to be evaluated. The form of the parameter is  $\{\text{min}, \text{max}\}$  oder **All**.

*violation (Real):* Only inequalities are printed out, that are violated more than this parameter. Default value is 0.

*opt: options:* Options for the *Mathematica*-function `Plot [...]` can be handed over.

**Description:** Like the function `PlotInequalities [...]` `ContPlotInequalities [...]` yields a graphical representation of the violation of Boole-Bell type inequalities, but in this case the functions are dependant on two variables:  $f(x,y) = p(x,y) - b$  derived from  $p(x,y) \leq b$ , where  $p(x,y)$  is a linear combination of functions to calculate the probability for single or joint events. Like in the description of the `PlotInequalities [...]` -function we take the “2 particles - 2 measurement directions”, the only difference is the selection of the measurement angles:

$$a_1 = x \quad a_2 = b_1 = 0 \quad b_2 = y$$

Thus the function  $f(x,y)$  is now given by

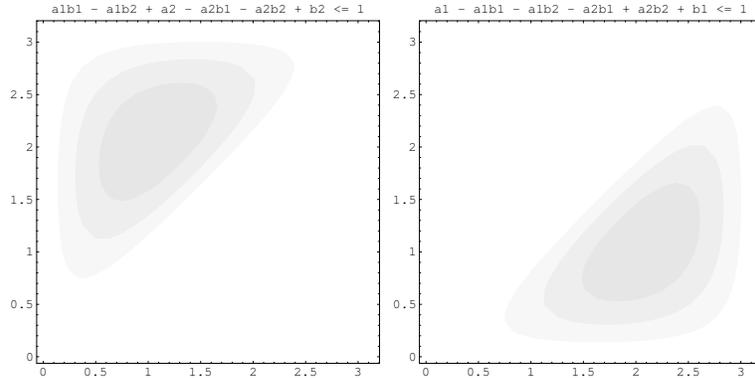
$$f(x,y) = 1 - \frac{1}{2} \sin \frac{x}{2} + \frac{1}{2} \sin \frac{x-y}{2} - \frac{1}{2} \sin \frac{y}{2} - 1$$

and can be plotted as contour plots. A higher level of violation is represented by a darker contour layer.

**Example:**

```
In[]= ContPlotInequalities [ConvToHRep [2,2], {x,0,π}, {y,0,π}, {{x,0}, {0,y}}, Prob, All]
```

```
Out[]=
```



The functions to calculate the probabilities have been defined as  $\text{Prob}[\{x_{-}\}] := \frac{1}{2}$  for a single event and  $\text{Prob}[\{x_{-}, y_{-}\}] := \frac{1}{2} \text{Sin}[(x - y)/2]^2$  for two-particle events.

### 3.13 Cdd

`Cdd [ values, opt: command ]`

*values (List)*: Data handed over to `cdd` in an input-file. Each element of the list must be a string and represents a row in the output-file.

*opt: command (String)*: Command to be executed on the generated output-file (default is “`cdd`” - the default value can be changed utilizing the function `SetCddCmd [ newcommand_String ]`). Here one can specify for example “`cddf`” or “`cddr`”.

**Description:** Simple Interface to run `cdd` in *Mathematica*. The current version cannot distinguish, whether `cdd` has produced correct output or not, so one has to pay attention while using this function.

#### Example:

```
In[] = Cdd [ { "H-Representation", "begin", "6 4 real", "2 -1 0 0", "2 0 -1 0", "-1 1 0 0", "-1 0 1 0", "-1 0 0 1", "4 -1 -1 0", "end" } ]
```

```
Out[] = { { "*", "cdd:", "Double", "Description", "Method", "C-Code:Version", "0.61b", "(November", 29, "1997)", { "*", "Copyright", "(C)", 1996, "Komei", "Fukuda,", "fukuda@ifor.math.ethz.ch" }, { "*Input", "File:tmp.ine", "(", 6, "x", "4)" }, { "*HyperplaneOrder:", "LexMin" }, { "*Degeneracy", "preknowledge", "for", "computation:", "None", "(possible", "degeneracy)" }, { "*Vertex/Ray", "enumeration", "is", "chosen." }, { "*Computation", "completed", "at", "Iteration", 6. }, { "*Computation", "starts", "at", "Thu", "Mar", 22, "18:48:36", 2001 }, { "*", "terminates", "at", "Thu", "Mar", 22, "18:48:36", 2001 },
```

```
{
  {"*Total", "processor", "time", "=", 0, "seconds"}, {"*", "=", 0, "hour", 0,
  "min", 0, "sec"}, {"*FINAL", "RESULT:"}, {"*Number", "of", "Vertices",
  "=", 4, "Rays", "=", 1}, {"V-representation"}, {"begin"}, {5, 4, "real"}, {1, 2,
  1, 1}, {1, 1, 1, 1}, {1, 1, 2, 1}, {1, 2, 2, 1}, {0, 0, 0, 1}, {"end"}, {"hull"}
}
```

## 4 Examples

These following two examples are originally solved in a paper by Pitowsky and Svozil [15]. The associated *Mathematica* - notebooks are "3\_2.nb" (three particles - 2 measurement directions) and "2\_3.nb" (two particles and three measurement directions)

### 4.1 Three particles and two measurement directions

In this configuration three particles ( $a$ ,  $b$  and  $c$ ) are measured in detectors which can be switched between two angles each. Consequently there are six different propositions for single particle events:  $a_1, a_2, b_1, b_2, c_1, c_2$ , supposing that  $a_1$  is the detection (i. e. the click in a counter) of the particle  $a$  in the detector set along the axis specified by the first angle for particle  $a$ ,  $b_2$  the detection of particle  $b$  at the second angle for this particle, and so on ... (cf. Figure 3). If we also take two and three particle events into

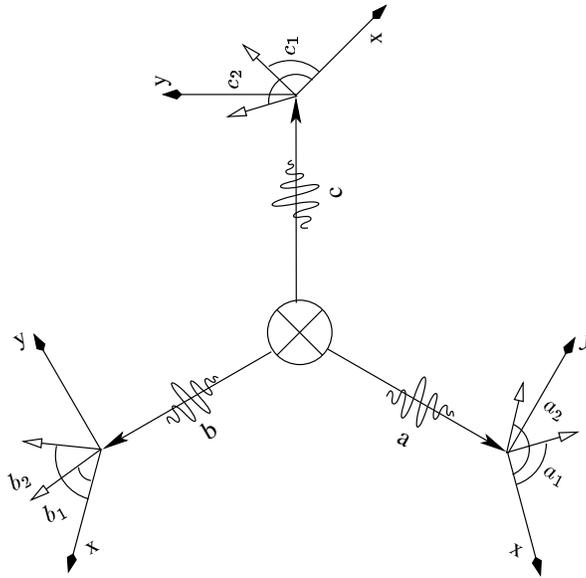


Figure 3: Setting for "2 particles - 3 angles"

account (for example the event  $a_2c_1$  means a click in the counter for particle  $a$  at the second angle AND a click in the counter for particle  $c$  at the first angle), there are 26 different events:

$a_1, a_2, b_1, b_2, c_1, c_2, a_1b_1, a_1b_2, a_2b_1, a_2b_2, a_1c_1, a_1c_2, a_2c_1, a_2c_2, b_1c_1, b_1c_2, b_2c_1, b_2c_2, a_1b_1c_1, a_1b_1c_2, a_1b_2c_1, a_1b_2c_2, a_2b_1c_1, a_2b_1c_2, a_2b_2c_1, a_2b_2c_2$

## 4.2 Violations of inequalities

The truth table for this configuration can be obtained utilizing the function `TruthTable[3,2]`<sup>2</sup>, executing `ConvToHRep[3,2]` yielded the appropriate H-representation, but this would last quite long, due to the complexity of the correlation polytope for this setting (there are 53856 hyper-planes limiting the polytope). Because of this fact trying to read in the H-representation created by `cdd` (using `ReadInHRep[...]`) could also result in memory resource problems by *Mathematica*. To avoid this symptoms it is suggested to export the list of vertices and apply `cdd` outside of *Mathematica* to the file containing the list of vertices (V-representation). This can be done by invoking `ToCddExtFile["3.2",3,2]`, which creates a file "3.2.ext". This file can be handed over to `cdd` as parameter to get the file "3.2.in" comprising the H-representation of the correlation polytope (Command: "`cdd 3.2.ext`").

Now the search for violated inequalities can begin using the function `GetViolInequalities[...]`:

May be accepted that the functions to calculate the quantum probabilities of the (joint) events (Prob) have been defined by `Prob[{x_}] := 1/2`, `Prob[{x_, y_}] := 1/4` and `Prob[{x_, y_, z_}] := 1/8 * (1 - Sin[x + y + z])`, where x, y and z are the angles used for detection of each particle,

```
GetViolInequalities["3.2.in",{{0, Pi/2}, {0, Pi/2}, {0, Pi/2}}, Prob, All, 0.4]
```

yields:

```
{{-3 a1+2 a1b1+a1b1c1-4 a1b1c2+3 a1b2-3 a1b2c1-
a1b2c2+a1c1+3 a1c2+2 a2b1-2 a2b1c1-a2b1c2-
2 a2b2+a2b2c1+3 a2b2c2+a2c1-a2c2-2 b1+
b1c1+2 b1c2+b2c1-2 b2c2-c1 <= 0,0.5},
{-2 a1+2 a1b1+a1b1c1-4 a1b1c2+2 a1b2-2 a1b2c1-
a1b2c2+a1c1+2 a1c2+3 a2b1-3 a2b1c1-a2b1c2-
2 a2b2+a2b2c1+3 a2b2c2+a2c1-2 a2c2-3 b1+
b1c1+3 b1c2+b2c1-b2c2-c1 <= 0,0.5},
{-2 a1+a1b1+a1b1c1-4 a1b1c2+2 a1b2-3 a1b2c1-
a1b2c2+2 a1c1+2 a1c2+2 a2b1-3 a2b1c1-
a2b1c2-a2b2+a2b2c1+2 a2b2c2+a2c1-a2c2-
2 b1+2 b1c1+2 b1c2+b2c1-b2c2-2 c1 <= 0,0.5},
{-2 a1+2 a1b1+a1b1c1-4 a1b1c2+2 a1b2-3 a1b2c1-
a1b2c2+3 a1c2+2 a2b1-3 a2b1c1-a2b1c2-
2 a2b2+a2b2c1+2 a2b2c2+a2c1-2 b1+2 b1c1+
```

---

<sup>2</sup>due to lack of space not listed here, but it can be found in the *Mathematica*- notebook 3.2.nb

$$\begin{aligned}
& 2 b_1 c_2 + b_2 c_1 - b_2 c_2 - c_1 - c_2 \leq 0, 0.5\}, \\
\{-a_1 + a_1 b_1 + a_1 b_1 c_1 - 4 a_1 b_1 c_2 + a_1 b_2 - 3 a_1 b_2 c_1 - \\
& a_1 b_2 c_2 + a_1 c_1 + 2 a_1 c_2 + a_2 + a_2 b_1 - 2 a_2 b_1 c_1 - \\
& a_2 b_1 c_2 - 2 a_2 b_2 + a_2 b_2 c_1 + 3 a_2 b_2 c_2 - a_2 c_2 - b_1 + \\
& b_1 c_1 + 2 b_1 c_2 + b_2 + b_2 c_1 - 2 b_2 c_2 - c_1 \leq 1, 1.5\}, \\
\{-a_1 + 2 a_1 b_1 - a_1 b_1 c_1 - 3 a_1 b_1 c_2 + a_1 b_2 - 4 a_1 b_2 c_1 + \\
& a_1 b_2 c_2 + 3 a_1 c_1 + a_2 + a_2 b_1 - 2 a_2 b_1 c_1 - a_2 b_1 c_2 - \\
& 2 a_2 b_2 + a_2 b_2 c_1 + 3 a_2 b_2 c_2 - a_2 c_2 - 2 b_1 + 2 b_1 c_1 + \\
& 2 b_1 c_2 + b_2 + 2 b_2 c_1 - 3 b_2 c_2 - 2 c_1 + c_2 \leq 1, 1.5\}, \\
\{-2 a_1 + 2 a_1 b_1 - a_1 b_1 c_1 - 3 a_1 b_1 c_2 + a_1 b_2 - 2 a_1 b_2 c_1 - \\
& a_1 b_2 c_2 + 2 a_1 c_1 + 2 a_1 c_2 + a_2 + a_2 b_1 - 4 a_2 b_1 c_1 + \\
& a_2 b_1 c_2 - 2 a_2 b_2 + a_2 b_2 c_1 + 3 a_2 b_2 c_2 + 2 a_2 c_1 - 3 a_2 c_2 - \\
& b_1 + 3 b_1 c_1 + b_2 - b_2 c_2 - 2 c_1 + c_2 \leq 1, 1.5\} \\
\{\dots\}\dots\}
\end{aligned}$$

### 4.3 Graphical representation

Using `PlotInequalities [...]` a graph can be created showing the violation of inequalities dependent on one variable. Defining the probability functions as above, executing

```
PlotInequalities ["3_2.ine", {x, 0, π}, {{0, x}, {0, x}, {0, x}}, Prob, {10000, 20000}, 0.4]
```

yields the following plot (cf. Figure 4), whereas the corresponding H-representation has to be stored in the file "3\_2.ine", {10000, 20000} indicates the range of row numbers taken for calculating the graph and 0.4 is the minimal degree of violation to include the inequality in the graph:

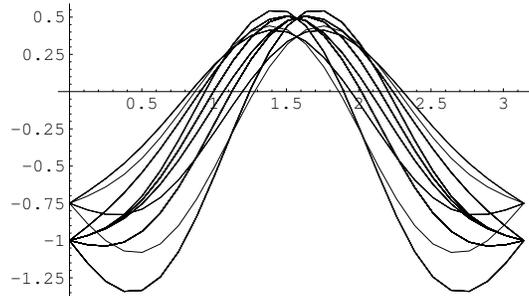


Figure 4: `PlotInequalities["3_2.ine", {x, 0, π}, {{0, x}, {0, x}, {0, x}}, Prob, {10000, 20000}, 0.4]`

To display inequalities dependent on two variables the function `ContPlotInequalities [...]` is provided. This function shows the violation

as a contour plot, a more violated set of detection angles results in a darker region in the plot.

`ContPlotInequalities [ "3_2.ine", {x,0,π}, {x,0,π}, {{0,x},{0,y},{x,y}}, Prob, {10000,20000}, 0.4]`  
returns `ContourGraphics`-objects, which can be displayed for example by executing  
`Show [GraphicsArray [Partition [cont, 3][[ {1,2}, All]]]] // TableForm`  
(see figure 5).

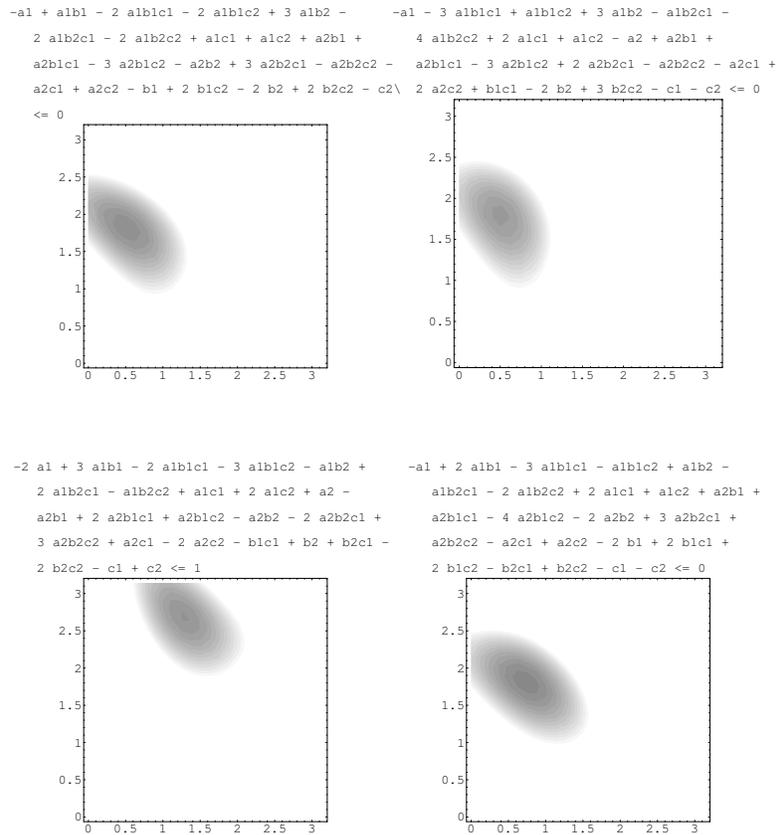


Figure 5: `ContPlotInequalities["3_2.ine", {x,0,π}, {x,0,π}, {{0,x},{0,y},{x,y}}, Prob, {10000,20000}, 0.4]`  
`Show[GraphicsArray[Partition[cont,3][[ {1,2}, All]]] // TableForm`

#### 4.4 Two particles and three measurement directions

In the case of two particles ( $a$  and  $b$ ) with three properties (whereas the properties are three different angles of the detectors for each particle denoted by  $a_1, a_2, a_3, b_1, b_2, b_3$  - see figure 6) 15 different events can be found:

$\{a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3, a_1b_1, a_1b_2, a_1b_3, a_2b_1, a_2b_2, a_2b_3, a_3b_1, a_3b_2, a_3b_3\}$

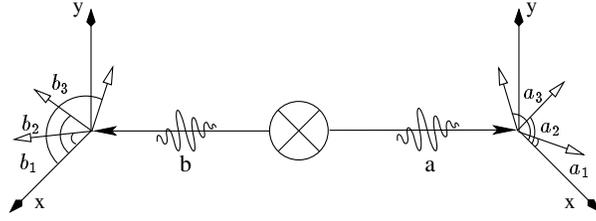


Figure 6: Setting for "2 particles - 3 angles"

#### 4.5 Violations of inequalities

Using TruthTable [2,3] all vertices of the corresponding correlation polytope can be found - we get a dimension of 15 and 64 vertices as result (table 3).

a1	a2	a3	b1	b2	b3	a1b1	a1b2	a1b3	a2b1	a2b2	a2b3	a3b1	a3b2	a3b3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
1	1	0	1	0	0	1	0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	1	0	0	0	0	0	1	0	0
0	1	1	1	0	0	1	0	0	1	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	1	0	0	0	0
1	1	0	0	1	0	0	1	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	0	0	1	0
1	0	1	0	1	0	0	1	0	0	0	0	0	1	0
0	1	1	0	1	0	0	0	0	1	0	0	0	1	0
0	0	1	1	1	0	0	0	0	0	0	1	1	0	0
1	0	1	1	1	0	1	1	0	0	0	0	1	1	0
0	1	1	1	1	0	0	0	0	1	1	0	1	1	0
1	1	1	1	1	0	1	1	0	1	1	0	1	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	1	0	0	0
1	1	0	0	0	1	0	0	1	0	0	0	1	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
1	0	1	0	0	1	0	0	1	0	0	0	0	0	1
0	1	1	0	0	1	0	0	0	0	0	1	0	0	1
0	1	1	0	0	1	0	0	0	0	0	1	0	0	1

a1	a2	a3	b1	b2	b3	a1b1	a1b2	a1b3	a2b1	a2b2	a2b3	a3b1	a3b2	a3b3
1	1	1	0	0	1	0	0	1	0	0	1	0	0	1
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	1	0	1	0	0	0	0	0	0
0	1	0	1	0	1	0	0	0	1	0	1	0	0	0
1	1	0	1	0	1	1	0	1	1	0	1	0	0	0
0	0	1	1	0	1	0	0	0	0	0	0	1	0	1
1	0	1	1	0	1	1	0	1	0	0	0	1	0	1
0	1	1	1	0	1	0	0	0	1	0	1	1	0	1
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	1	1	0	0	0	0	0	0
0	1	0	0	1	1	0	0	0	0	1	1	0	0	0
1	1	0	0	1	1	0	1	1	0	1	1	0	0	0
0	0	1	0	1	1	0	0	0	0	0	0	0	1	1
1	0	1	0	1	1	0	1	1	0	0	0	0	1	1
0	1	1	0	1	1	0	0	0	1	1	1	0	1	1
1	1	1	0	1	1	0	1	1	0	1	1	0	1	1
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	0	0	0	0	0	0
0	1	0	1	1	1	0	0	0	1	1	1	0	0	0
1	1	0	1	1	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	0	0	0	0	0	0	1	1	1
1	0	1	1	1	1	1	1	1	0	0	0	1	1	1
0	1	1	1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 3: Truth table for 6 propositions

Executing `hrep=ConvToHRep [2,3]` a H-representation of the polytope described by the truth table above can be created. This results in 684 hyper-planes respectively 684 inequalities from the 64 vertices limiting the polytope:

```
{H-representation}, {begin},{684, 16, real},
{2, 0, -2, 1, -1, 0, -1, 1, -1, 0, 1, 1, 1, -1, -1, 1},
{2, -2, 0, 1, -1, 0, -1, 1, 1, 1, 1, -1, 0, -1, -1, 1},
{...},
{1, -1, 0, 0, -1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0},
{end},{Konfiguration,2,3}}
```

All inequalities can be displayed by

```
GetInequFromHRep [hrep] // InequToRead
```

The result of this operation is

```
-a1b1 + a1b2 + 2 a2 - a2b1 - a2b2 - a2b3 - a3 + a3b1 + a3b2 - a3b3 + b1 + b3 ≤ 2
2a1 - a1b1 - a1b2 - a1b3 - a2b1 + a2b2 - a3 + a3b1 + a3b2 - a3b3 + b1 + b3 ≤ 2
a1 - a1b1 - a1b3 - a2b1 + a2b2 + a2b3 - a3 + a3b1 + a3b2 - a3b3 + b1 - b2 ≤ 1
...
...
a1- a1b1 + b1 ≤ 1
```

Using

GetViolInequalities [hrep,{{0, $\frac{2\pi}{3}$ , $\frac{4\pi}{3}$ }},{{0, $\frac{2\pi}{3}$ , $\frac{4\pi}{3}$ }},Prob,All] // TableForm

all inequalities can be displayed that are violated at the specific angles  $a_1 = b_1 = 0$ ,  $a_2 = b_2 = \frac{2\pi}{3}$  and  $a_3 = b_3 = \frac{4\pi}{3}$  taking the functions  $\text{Prob}[\{x_{-}\}] := \frac{1}{2}$  and  $\text{Prob}[\{x_{-}, y_{-}\}] := \text{Sin}[\frac{x_{-}y_{-}}{2}]/2$  to calculate the quantum probabilities, which is equivalent to the probability to find two spin- $\frac{1}{2}$  particles in a singlet state ( $|\psi\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$ ) both either in spin “up” or both in spin “down”.

$$\begin{array}{r}
 -a_1 - a_1b_1 + a_1b_2 + a_1b_3 - a_2 + a_2b_1 + a_2b_3 + a_3b_1 + a_3b_2 - a_3b_3 - b_1 - b_2 \leq 0 \\
 -a_1 - a_1b_1 + a_1b_2 + a_1b_3 + a_2b_1 - a_2b_2 + a_2b_3 - a_3 + a_3b_1 + a_3b_2 - b_1 - b_3 \leq 0 \\
 -a_1 + a_1b_2 + a_1b_3 - a_2 + a_2b_1 - a_2b_2 + a_2b_3 + a_3b_1 + a_3b_2 - a_3b_3 - b_1 - b_2 \leq 0 \\
 -a_1b_1 + a_1b_2 + a_1b_3 - a_2 + a_2b_1 - a_2b_2 + a_2b_3 - a_3 + a_3b_1 + a_3b_2 - b_2 - b_3 \leq 0 \\
 -a_1 + a_1b_2 + a_1b_3 + a_2b_1 - a_2b_2 + a_2b_3 - a_3 + a_3b_1 + a_3b_2 - a_3b_3 - b_1 - b_3 \leq 0 \\
 -a_1b_1 + a_1b_2 + a_1b_3 - a_2 + a_2b_1 + a_2b_3 - a_3 + a_3b_1 + a_3b_2 - a_3b_3 - b_2 - b_3 \leq 0 \\
 -a_1 + a_1b_2 + a_1b_3 - a_2b_2 + a_2b_3 - b_3 \leq 0 \\
 -a_1b_1 + a_1b_3 - a_2 + a_2b_1 + a_2b_3 - b_3 \leq 0 \\
 -a_1b_1 + a_1b_2 - a_3 + a_3b_1 + a_3b_2 - b_2 \leq 0 \\
 -a_1 + a_1b_2 + a_1b_3 + a_3b_2 - a_3b_3 - b_2 \leq 0 \\
 a_2b_1 - a_2b_2 - a_3 + a_3b_1 + a_3b_2 - b_1 \leq 0 \\
 -a_2 + a_2b_1 + a_2b_3 + a_3b_1 - a_3b_3 - b_1 \leq 0
 \end{array}$$

## 4.6 Graphical representation

Like in the configuration “three particles and two angles” described above a graphical representation can be generated either dependent on one or dependent on two variables. On the one hand side by using

PlotInequalities [hrep,{x,0, $\pi$ },{{0,x, $\frac{4\pi}{3}$ }},{{0,x, $\frac{4\pi}{3}$ }},Prob]

we get a plot of all violated inequalities (cf. Figure 7),

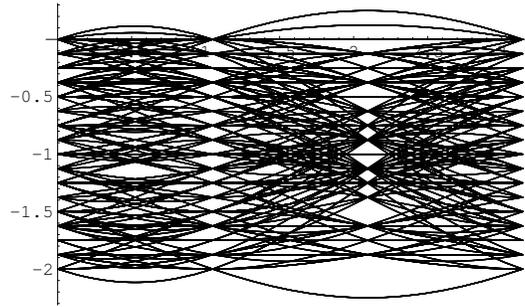


Figure 7: PlotInequalities[hrep,{x,0, $\pi$ },{{0,x, $\frac{4\pi}{3}$ }},{{0,x, $\frac{4\pi}{3}$ }},Prob]

on the other hand side contour plots of all inequalities violated for example more than 0.2 can be generated by executing

```
cont=ContPlotInequalities [hrep,{x,0,π},{y,0,π},{0,x,y},{0,x,y},Prob,All,0.2]
```

To display the outcome of the calculation entering `Show /@ cont` results in Contour-Plots of the following form (cf. Figure 8):

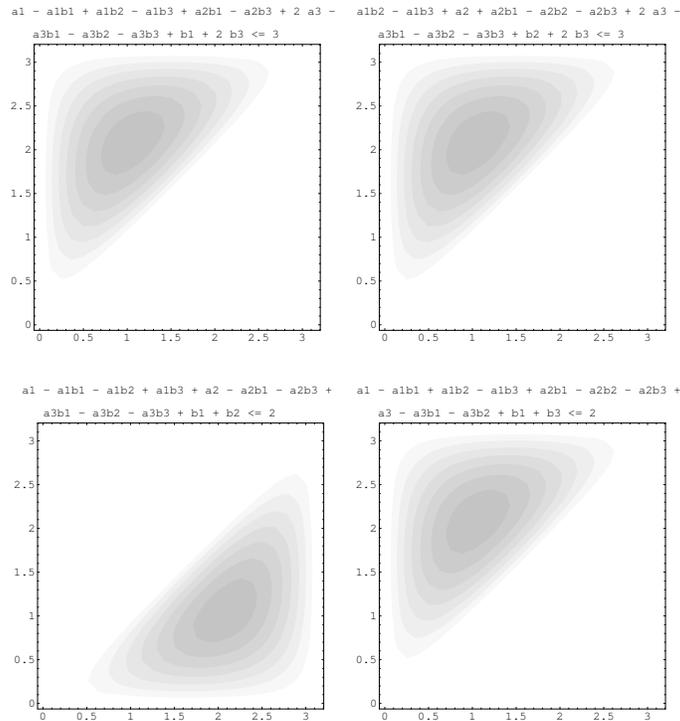


Figure 8: `ContPlotInequalities[hrep,{x,0,π},{y,0,π},{0,x,y},{0,x,y},Prob,All,0.2]`  
`Show /@ %`

## References

- [1] George Boole. *An investigation of the laws of thought*. Dover edition, New York, 1958.
- [2] George Boole. On the theory of probabilities. *Philosophical Transactions of the Royal Society of London*, 152:225–252, 1862.
- [3] Theodore Hailperin. *Boole's logic and probability (Studies in logic and the foundations of mathematics ; 85)*. North-Holland, Amsterdam, 1976.
- [4] Itamar Pitowsky. *Quantum Probability—Quantum Logic*. Springer, Berlin, 1989.

- [5] Itamar Pitowsky. George Boole's 'conditions of possible experience' and the quantum puzzle. *Brit. J. Phil. Sci.*, 45:95–125, 1994.
- [6] J. F. Clauser and M. A. Horne. *Physical Review*, D10:526, 1974.
- [7] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. *Physical Review Letters*, 23:880–884, 1969.
- [8] J. F. Clauser and A. Shimony. Bell's theorem: experimental tests and implications. *Rep. Prog. Phys.*, 41:1881–1926, 1978.
- [9] Itamar Pitowsky. From George Boole to John Bell: The origin of Bell's inequality. In M. Kafatos, editor, *Bell's Theorem, Quantum Theory and the Conceptions of the Universe*. Kluwer, Dordrecht, 1989.
- [10] Itamar Pitowsky. Correlation polytopes their geometry and complexity. *Mathematical Programming*, 50:395–414, 1991.
- [11] Günter M. Ziegler. *Lectures on Polytopes*. Springer, New York, 1994.
- [12] T.S. Motzkin, H. Raiffa, G.L. Thompson, and R.M. Thrall. The double description method. In *Contributions to theory of games, Vol. 2*. Princeton University Press, New Jersey, Princeton, 1953.
- [13] A. N. Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Springer, Berlin, 1933. English translation in [16].
- [14] Komei Fukuda. cdd program. 2000.
- [15] Itamar Pitowsky and Karl Svozil. New optimal tests of quantum nonlocality. *Physical Review*, A, to appear, 2001. [arXiv:quant-ph/0011060](https://arxiv.org/abs/quant-ph/0011060).
- [16] A. N. Kolmogorov. *Foundations of the Theory of Probability*. Chelsea, New York, 1956.