STK3405 – Week 34a

A. B. Huseby & K. R. Dahl

Department of Mathematics
University of Oslo, Norway

Overview

- System analysis:
  - Binary monotone systems
  - Coherent systems
  - Reliability of binary monotone systems
- Basic reliability calculation methods
  - Pivotal decompositions (conditioning)
  - Path and cut sets
  - Modules of monotone systems
- Exact computation of reliability
  - State space enumeration
  - The multiplication method
  - The inclusion-exclusion methods
  - Network reliability

Overview

- System analysis:
  - Binary monotone systems
  - Coherent systems
  - Reliability of binary monotone systems
- Basic reliability calculation methods
  - Pivotal decompositions (conditioning)
  - Path and cut sets
  - Modules of monotone systems
- Exact computation of reliability
  - State space enumeration
  - The multiplication method
  - The inclusion-exclusion methods
  - Network reliability

Overview

- System analysis:
  - Binary monotone systems
  - Coherent systems
  - Reliability of binary monotone systems
- Basic reliability calculation methods
  - Pivotal decompositions (conditioning)
  - Path and cut sets
  - Modules of monotone systems
- Exact computation of reliability
  - State space enumeration
  - The multiplication method
  - The inclusion-exclusion methods
  - Network reliability

Overview (cont.)

- Structural and reliability importance:
  - Structural importance of a component
  - Reliability importance of a component
  - Time-independent measures
- Association and reliability bounds
  - Associated random variables
  - Bounds on the system reliability
- Conditional Monte Carlo methods
  - Monte Carlo simulation and conditioning
  - Conditioning on the sum
  - Identical component reliabilities

Overview (cont.)

- Structural and reliability importance:
  - Structural importance of a component
  - Reliability importance of a component
  - Time-independent measures
- Association and reliability bounds
  - Associated random variables
  - Bounds on the system reliability
- Conditional Monte Carlo methods
  - Monte Carlo simulation and conditioning
  - Conditioning on the sum
  - Identical component reliabilities

Overview (cont.)

- Structural and reliability importance:
  - Structural importance of a component
  - Reliability importance of a component
  - Time-independent measures
- Association and reliability bounds
  - Associated random variables
  - Bounds on the system reliability
- Conditional Monte Carlo methods
  - Monte Carlo simulation and conditioning
  - Conditioning on the sum
  - Identical component reliabilities

Overview (cont.)

- Discrete event simulation:
  - Pure jump processes
  - Binary monotone systems of repairable components
  - Simulating repairable systems
  - Estimating availability and importance
- Applications
  - Case study: Fishing boat engines
  - Case study: Transmission of electronic pulses

Overview (cont.)

- Discrete event simulation:
  - Pure jump processes
  - Binary monotone systems of repairable components
  - Simulating repairable systems
  - Estimating availability and importance
- Applications
  - Case study: Fishing boat engines
  - Case study: Transmission of electronic pulses

Section 2.1

Binary monotone systems

Binary monotone systems

- A *system* is some technological unit consisting of a finite set of *components* which are operating together
- A *binary system* has only two possible states: *functioning* or *failed*. Moreover, each component is either functioning or failed as well
- A *binary monotone system* is a binary system such that repairing a component does not make the system worse, and breaking a component does not make system better

Binary monotone systems (cont.)

We consider a binary system of n components, $1, \ldots, n$, and introduce the component state variable X_i denoting the *state* of component i, $i = 1, \ldots, n$, defined as:

$$X_i := \begin{cases} 1 & \text{if the } i\text{th component is functioning} \\ 0 & \text{otherwise.} \end{cases}$$

Furtermore, we introduce the variable ϕ representing the *state* of the system, defined as:

$$\phi := \begin{cases} 1 & \text{if the system is functioning} \\ 0 & \text{otherwise.} \end{cases}$$

Binary monotone systems (cont.)

The variables X_i, $i = 1, \ldots, n$ and ϕ are said to be *binary*, since they only have two possible values, 0 and 1.

The state of the system is assumed to be uniquely determined by the state of the components. Thus, we may write ϕ as:

$$\phi = \phi(\boldsymbol{X}) = \phi(X_1, X_2, \ldots, X_n)$$

The function $\phi : \{0, 1\}^n \to \{0, 1\}$ is called the *structure function* of the system.

Binary monotone systems (cont.)

Definition

A binary monotone system is an ordered pair (C, ϕ), where:

$C = \{1, \ldots, n\}$ is the component set and ϕ is the structure function.

The structure function ϕ is non-decreasing in each argument.

The number of components, n, is referred to as the *order* of the system.

Trivial systems

NOTE: According to the definition binary monotone systems includes systems where the structure function is constant, i.e., systems where:

$$\phi(\boldsymbol{x}) = 1 \text{ for all } \boldsymbol{x} \in \{0, 1\}^n$$

or

$$\phi(\boldsymbol{x}) = 0 \text{ for all } \boldsymbol{x} \in \{0, 1\}^n$$

We will refer to such systems as *trivial* systems.

Some textbooks exclude trivial systems from the class of monotone systems. We have chosen not to do so in order to have a class which is closed with respect to conditioning.

Trivial systems (cont.)

Theorem

Let (C, ϕ) *be a non-trivial binary monotone system.*

Then $\phi(\mathbf{0}) = 0$ *and* $\phi(\mathbf{1}) = 1$*.*

Proof: Since (C, ϕ) is assumed to be non-trivial, there exists at least one vector $\boldsymbol{x}_0$ such that $\phi(\boldsymbol{x}_0) = 0$ and another vector $\boldsymbol{x}_1$ such that $\phi(\boldsymbol{x}_1) = 1$.

Since (C, ϕ) is monotone, it follows that:

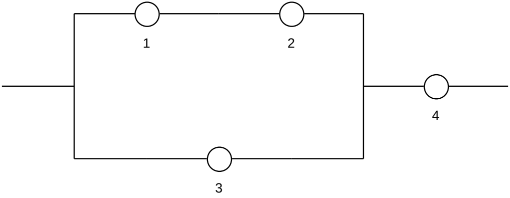
$$0 \leq \phi(\mathbf{0}) \leq \phi(\boldsymbol{x}_0) = 0$$

$$1 = \phi(\boldsymbol{x}_1) \leq \phi(\mathbf{1}) \leq 1$$

Hence, we conclude that $\phi(\mathbf{0}) = 0$ and $\phi(\mathbf{1}) = 1$

Reliability block diagrams

In a *reliability block diagram* components are drawn as circles and connected by lines. The system is functioning if and only it is possible to find a way through the diagram passing only functioning components.



Note: In order to represent arbitrarily binary monotone systems, we allow components to occur in multiple places in the block diagram. Thus, a reliability block diagram should not necessarily be interpreted as a picture of a physical system.

A series system

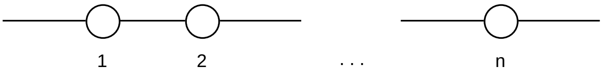


Figure: A reliability block diagram of a series system.

A parallel system

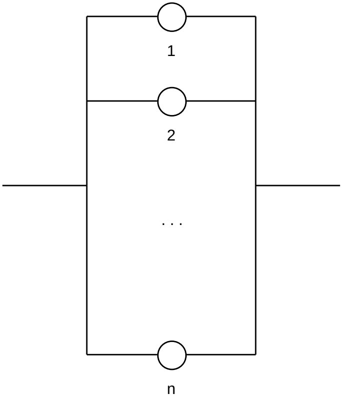


Figure: A reliability block diagram of a parallel system.

A 2-out-of-3 system

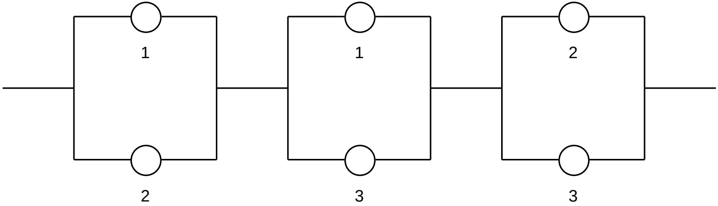


Figure: A reliability block diagram of a 2-out-of-3 system.

For a 2-out-of-3 system to fail 2 out of 3 components must fail. There are 3 possible subsets of components which contains 2 components: $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$.

A 2-out-of-4 system

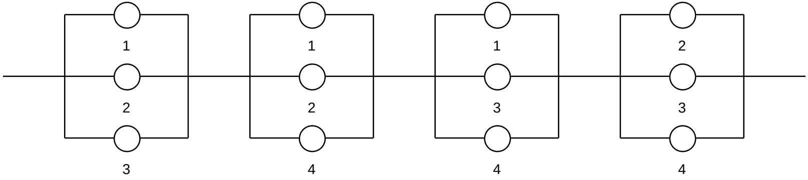


Figure: A reliability block diagram of a 2-out-of-4 system.

For a 2-out-of-4 system to fail 3 out of 4 components must fail. There are 4 possible subsets of components which contains 3 components: $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, $\{2, 3, 4\}$.

The structure function of a series system

Figure: A reliability block diagram of a series system.

The structure function of a series system (cont.)

From the reliability block diagram we see that the series system is functioning if and only if all of its components are functioning, i.e., if and only if $X_1 = 1$ and $\cdots$ and $X_n = 1$.

Hence, the state of the system can be expressed as a function of the component state variables as follows:

$$\phi(\boldsymbol{X}) = X_1 \cdot X_2 \cdots X_n = \prod_{i=1}^{n} X_i$$

Alternatively, the structure function of a series system can be written as:

$$\phi(\boldsymbol{X}) = \min\{X_1, \ldots, X_n\}$$

since $\min\{X_1, \ldots, X_n\} = 1$ if and only if $X_1 = 1$ and $\cdots$ and $X_n = 1$.

The coproduct operator

For any $a_1, a_2, \ldots, a_n$ we define:

$$a_1 \amalg a_2 = 1 - (1 - a_1)(1 - a_2),$$

$$\coprod_{i=1}^{n} a_i = 1 - \prod_{i=1}^{n}(1 - a_i).$$

Note: If $a_1, a_2, \ldots, a_n \in \{0, 1\}$, we have:

$$a_1 \amalg a_2 = 1 - (1 - a_1)(1 - a_2) = \max\{a_1, a_2\},$$

$$\coprod_{i=1}^{n} a_i = 1 - \prod_{i=1}^{n}(1 - a_i) = \max\{a_1, \ldots, a_n\}.$$

The coproduct operator (cont.)

Proof:

$$a_1 \amalg a_2 = 1 - (1 - a_1)(1 - a_2) = 0$$

if and only if

$$(1 - a_1)(1 - a_2) = 1$$

Hence, $a_1 \amalg a_2 = 0$ if and only if $a_1 = 0$ and $a_2 = 0$.

Equivalently, $a_1 \amalg a_2 = 1$ if and only if $a_1 = 1$ or $a_2 = 1$.

Hence, $a_1 \amalg a_2 = \max\{a_1, a_2\}$.

The coproduct operator (cont.)

$$\coprod_{i=1}^{n} a_i = 1 - \prod_{i=1}^{n}(1 - a_i) = 0$$

if and only if

$$\prod_{i=1}^{n}(1 - a_i) = 1$$

Hence, $\coprod_{i=1}^{n} a_i = 0$ if and only if $a_1 = 0$ and $\cdots$ and $a_n = 0$.

Equivalently, $\coprod_{i=1}^{n} a_i = 1$ if and only if $a_1 = 1$ or $\cdots$ or $a_n = 1$.

Hence, $\coprod_{i=1}^{n} a_i = \max\{a_1, \ldots, a_n\}$.

The structure function of a parallel system

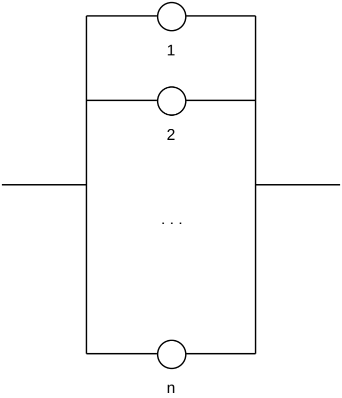


Figure: A reliability block diagram of a parallel system.

The structure function of a parallel system (cont.)

From the reliability block diagram we see that the parallel system is functioning if and only if at least one of its components are functioning, i.e., if and only if $X_1 = 1$ or $\cdots$ or $X_n = 1$.

Hence, the state of the system can be expressed as a function of the component state variables as follows:

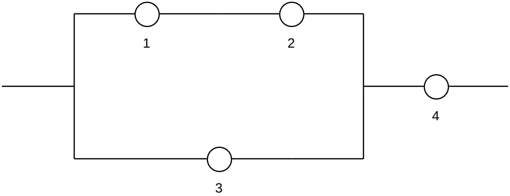
$$\phi(\boldsymbol{X}) = X_1 \amalg X_2 \cdots \amalg X_n = \coprod_{i=1}^{n} X_i.$$

Alternatively, the structure function of a parallel system can be written as:

$$\phi(\boldsymbol{X}) = \max\{X_1, \ldots, X_n\}$$

since $\max\{X_1, \ldots, X_n\} = 1$ if and only if $X_1 = 1$ or $\cdots$ or $X_n = 1$.

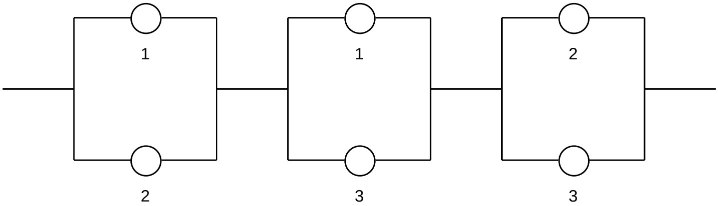
The structure function of a mixed system



It is easy to verify that the structure function of this system is:

$$\phi(\boldsymbol{X}) = [(X_1 \cdot X_2) \amalg X_3] \cdot X_4$$

The structure function of a 2-out-of-3 system



It is easy to verify that the structure function of this system is:

$$\phi(\boldsymbol{X}) = (X_1 \amalg X_2)(X_1 \amalg X_3)(X_2 \amalg X_3)$$

Coherent systems

Coherent systems

Every component of a binary monotone system should have some impact on the system state. More precisely, if i is a component in a system, there should ideally exist at least some state of the rest of the system where the system state depends on the state of component i.

Notation:

$$(1_i, \boldsymbol{x}) = (x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$$

$$(0_i, \boldsymbol{x}) = (x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)$$

$$(\cdot_i, \boldsymbol{x}) = (x_1, \ldots, x_{i-1}, \cdot, x_{i+1}, \ldots, x_n).$$

Coherent systems (cont.)

Note: If X is a binary variable, then $X^n = X$ for $n = 1, 2, \ldots$.

The coproduct of two variables:

$$\begin{aligned} X_1 \amalg X_2 &= 1 - (1 - X_1)(1 - X_2) \\ &= 1 - (1 - X_1 - X_2 + X_1 X_2) \\ &= 1 - 1 + X_1 + X_2 - X_1 X_2 \\ &= X_1 + X_2 - X_1 X_2 \end{aligned}$$

Coherent systems (cont.)

Definition

Let (C, ϕ) be a binary monotone system, and let $i \in C$. The component i is said to be *relevant* for the system (C, ϕ) if:

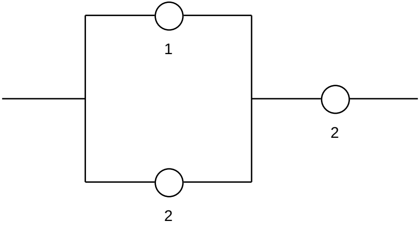
$$0 = \phi(0_i, \boldsymbol{x}) < \phi(1_i, \boldsymbol{x}) = 1 \text{ for some } (\cdot_i, \boldsymbol{x}).$$

If this is not the case, component i is said to be *irrelevant* for the system.

A binary monotone system (C, ϕ) is *coherent* if all its components are relevant.

Note that a coherent system is obviously non-trivial as well, since in a trivial system *all* components are irrelevant.

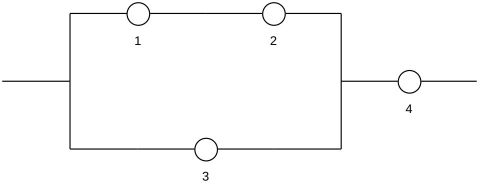
An incoherent system



The structure function of this system is:

$$\phi(\boldsymbol{X}) = (X_1 \amalg X_2) \cdot X_2$$
$$= (X_1 + X_2 - X_1X_2) \cdot X_2$$
$$= X_1X_2 + X_2^2 - X_1X_2^2$$
$$= X_1X_2 + X_2 - X_1X_2 = X_2$$

Conditionally irrelevant components



The structure function of this system is:

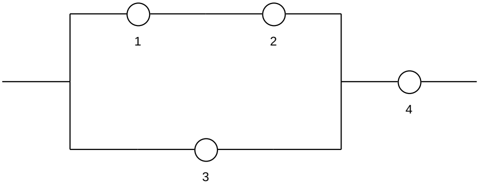
$$\phi(\boldsymbol{X}) = [(X_1 \cdot X_2) \amalg X_3] \cdot X_4$$
$$= (X_1 X_2 + X_3 - X_1 X_2 X_3) \cdot X_4$$

Given that component 2 is failed, i.e., $X_2 = 0$, we get:

$$\phi(0_2, \boldsymbol{X}) = X_3 \cdot X_4$$

Thus, component 1 is *conditionally irrelevant* given that component 2 is failed.

Conditionally irrelevant components (cont.)



The structure function of this system is:

$$\phi(\boldsymbol{X}) = [(X_1 \cdot X_2) \amalg X_3] \cdot X_4$$

$$= (X_1X_2 + X_3 - X_1X_2X_3) \cdot X_4$$

Given that component 3 is functioning, i.e., $X_3 = 1$, we get:

$$\phi(1_3, \boldsymbol{X}) = (X_1X_2 + 1 - X_1X_2) \cdot X_4 = X_4$$

Thus, components 1 and 2 are *conditionally irrelevant* given that component 3 is functioning.

The best and worst systems

Theorem

Let (C, ϕ) be a non-trivial binary monotone system of order n. Then for all $\boldsymbol{x} \in \{0,1\}^n$ we have:

$$\prod_{i=1}^{n} x_i \leq \phi(\boldsymbol{x}) \leq \coprod_{i=1}^{n} x_i.$$

Proof:
If $\prod_{i=1}^{n} x_i = 0$, then $\prod_{i=1}^{n} x_i \leq \phi(\boldsymbol{x})$ since $\phi(\boldsymbol{x}) \in \{0,1\}$ for all $\boldsymbol{x} \in \{0,1\}^n$.

If $\prod_{i=1}^{n} x_i = 1$, we have $\boldsymbol{x} = \mathbf{1}$. Hence, $\phi(\boldsymbol{x}) = \phi(\mathbf{1}) = 1$ since (C, ϕ) is non-trivial

Thus, the inequality is valid in this case as well. This completes the proof of the left-hand inequality.

The right-hand inequality is proved in a similar way.

The product and coproduct operators for vectors

Consider two vectors $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$.

For our purpose the *product* and *coproduct* operators for vectors are defined as follows:

$$\boldsymbol{x} \cdot \boldsymbol{y} = (x_1 \cdot y_1, x_2 \cdot y_2, \ldots, x_n \cdot y_n),$$

$$\boldsymbol{x} \amalg \boldsymbol{y} = (x_1 \amalg y_1, x_2 \amalg y_2, \ldots, x_n \amalg y_n).$$

Component level changes vs. system level changes

Theorem

Let (C, ϕ) be a binary monotone system of order n. Then for all binary vectors $\boldsymbol{x}, \boldsymbol{y}$ we have:

(i) $\phi(\boldsymbol{x} \amalg \boldsymbol{y}) \geq \phi(\boldsymbol{x}) \amalg \phi(\boldsymbol{y})$,

(ii) $\phi(\boldsymbol{x} \cdot \boldsymbol{y}) \leq \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y})$.

Moreover, assume that (C, ϕ) is coherent. Then equality holds in (i) for all $\boldsymbol{x}, \boldsymbol{y} \in \{0, 1\}^n$ if and only if (C, ϕ) is a parallel system. Similarly, equality holds in (ii) for all $\boldsymbol{x}, \boldsymbol{y} \in \{0, 1\}^n$ if and only if (C, ϕ) is a series system.

Interpretation: Components in parallel are better than systems in parallel. Components in series are worse than systems in series.

Component level vs. system level (cont.)

Proof: Since ϕ is non-decreasing and $x_i \amalg y_i \geq x_i$ for $i = 1, \ldots, n$, it follows that:

$$\phi(\boldsymbol{x} \amalg \boldsymbol{y}) \geq \phi(\boldsymbol{x}).$$

Similarly, we see that

$$\phi(\boldsymbol{x} \amalg \boldsymbol{y}) \geq \phi(\boldsymbol{y}).$$

Hence,

$$\phi(\boldsymbol{x} \amalg \boldsymbol{y}) \geq \max\{\phi(\boldsymbol{x}), \phi(\boldsymbol{y})\} = \phi(\boldsymbol{x}) \amalg \phi(\boldsymbol{y}).$$

This proves (i).

The proof of (ii) is similar.

Component level vs. system level (cont.)

We then prove that equality holds in (i) for all binary vectors $\boldsymbol{x}, \boldsymbol{y}$ if and only if (C, ϕ) is a parallel system.

If (C, ϕ) is a parallel system, it follows that:

$$\begin{aligned}\phi(\boldsymbol{x} \amalg \boldsymbol{y}) &= \amalg_{i=1}^{n}(x_i \amalg y_i) = \max_{1 \leq i \leq n} \{ \max\{x_i, y_i\}\} \\ &= \max\{ \max_{1 \leq i \leq n} x_i, \max_{1 \leq i \leq n} y_i \} = \phi(\boldsymbol{x}) \amalg \phi(\boldsymbol{y}),\end{aligned}$$

which proves the "if"-part of the equivalence.

Component level vs. system level (cont.)

Assume conversely that $\phi(\boldsymbol{x} \amalg \boldsymbol{y}) = \phi(\boldsymbol{x}) \amalg \phi(\boldsymbol{y})$ for all binary vectors $\boldsymbol{x}, \boldsymbol{y}$.

Since (C, ϕ) is coherent it follows that for any $i \in C$ there exists a vector $(\cdot_i, \boldsymbol{x})$ such that:

$$\phi(1_i, \boldsymbol{x}) = 1 \text{ and } \phi(0_i, \boldsymbol{x}) = 0.$$

For this particular vector $(\cdot_i, \boldsymbol{x})$ we have:

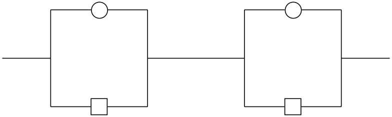
$$\begin{aligned} 1 &= \phi(1_i, \boldsymbol{x}) = \phi\big((1_i, \mathbf{0}) \amalg (0_i, \boldsymbol{x})\big) \\ &= \phi(1_i, \mathbf{0}) \amalg \phi(0_i, \boldsymbol{x}) = \phi(1_i, \mathbf{0}) \amalg 0 \\ &= \phi(1_i, \mathbf{0}). \end{aligned}$$

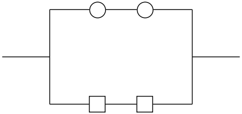
This implies that component i is in parallel with the rest of the system. Since this holds for any $i \in C$, we conclude that the system is a parallel system. This proves the "only if"-part of the equivalence.

The other equivalence is proved similarly.

Component level vs. system level (cont.)

Example: A series system of two components.





For all binary vectors $\boldsymbol{x} = (x_1, x_2)$ and $\boldsymbol{y} = (y_1, y_2)$ we have:

$$(x_1 \amalg y_1) \cdot (x_2 \amalg y_2) \geq (x_1 \cdot x_2) \amalg (y_1 \cdot y_2).$$

Component level vs. system level (cont.)

We have:

$$(x_1 \amalg y_1) \cdot (x_2 \amalg y_2)$$
$$= (x_1 + y_1 - x_1 y_1) \cdot (x_2 + y_2 - x_2 y_2) :$$

$$(x_1 \cdot x_2) \amalg (y_1 \cdot y_2) = x_1 x_2 + y_1 y_2 - x_1 x_2 y_1 y_2.$$

From this it can be shown that:

$$(x_1 \amalg y_1) \cdot (x_2 \amalg y_2) - (x_1 \cdot x_2) \amalg (y_1 \cdot y_2) = \cdots$$
$$= x_1(1 - x_2)(1 - y_1)y_2 + (1 - x_1)x_2 y_1(1 - y_2) \geq 0.$$