

# **Selected methods of knowledge engineering for system diagnosis**

# Process of fault diagnosis

- Detection of faulty behavior of the system
- Classification of this behavior
- Search for and determination of causes of the observed misbehavior :
  - Generation of potential diagnoses
  - Verification of diagnoses
  - Selection of the correct one
- Repair phase

# Diagnostic process

- This problem is one inverse to simulation task.
  - one observes a faulty behavior of the simulated system (and thus apart from the knowledge about correct behavior also the one about faulty behavior should be accessible)
  - taking into account the observed state (output) the main goal is not reconstruction of the input (control) but rather the causes of the failure are searched for.

# Diagnostic process

- $D = \{d_1, d_2, \dots, d_n\}$  - the binary set of potential elementary causes to be considered
- $M = \{m_1, m_2, \dots, m_m\}$  - the binary set of failure symptoms
- In case of a failure can be observed:
  - $m_i \in M$
  - $M^+ \subseteq M$
- Diagnosis - a set  $D^+ \subseteq D$  explains the observed misbehavior.

# The diagnoses

- In general case, the result of the diagnostic process can consists of one or more potential diagnoses - subsets of the set  $D$  :
  - single-element sets (elementary diagnoses)
  - multi-element sets
    - minimal diagnoses

# Building a diagnostic system

- *Causal Relation:*

- $R_C \subseteq 2^D \times 2^M$

- $R_C : 2^D \longrightarrow 2^M$

- *Diagnostic function:*

- $f = R_C^{-1}$

# Example

The set of  $n$  bulbs for a Christmas Tree:

- An elementary diagnosis:  
 $d_i$  ( $i$ -th bulb being blown)
- The set of manifestations of the failure:  
 $M = \{m_1\}$  (the bulbs are not on)
- There exists  $n$  single-element elementary diagnoses :  $m_1$
- There exists  $(2^n - 1)$  multi-element diagnoses

# Knowledge engineering

- Symbolic representation of domain and expert knowledge
- Automated inference paradigms for knowledge processing



# Knowledge engineering in diagnostic

- Type (source) and the way of specification of diagnostic knowledge
- Applied knowledge representation methods
- Applied inference methods
- Inference control mechanism

# Expert methods

Methods based on use of numerical data:

- Pattern recognition methods in feature space
- Classifiers using the technology of artificial neural networks
- Simple rule-based classifiers, including fuzzy rule-based systems
- Hybrid systems

# Expert methods

Methods using symbolic data and knowledge:

- Diagnostic tests
- Fault dictionaries
- Decision trees
- Decision tables
- Logic based methods, mainly rule-based systems and expert systems
- Case-based systems

# Model-based methods

## Consistency-based methods:

- Consistency-based reasoning using purely logical models (Reiter's theory)
- Consistency-based reasoning using mathematical, causal models and qualitative models

# Model-based methods

## Causal methods:

- Diagnostic graphs and relations
- Fault trees
- Causal graphs
- Logical abductive reasoning
- Logical causal graphs

# Causal relationship in diagnosis

- $d$  - elementary diagnosis
  - *True* - the fault occurs
  - *False* - the fault is not observed
- $m$  - diagnostic signal (true or false)
- $d \models m$ , i.e.  $m$  is a logical consequence of  $d$
- $t_d < t_m$ , i.e. a cause precedes its result
- there exists a flow of a physical signal from symptom  $d$  to symptom  $m$

# Causal Relationships

- $V = \{v_1, v_2, \dots, v_k\}$  - set of symptoms (*True* or *False*)

- *OR* type

- $v_1 \vee v_2 \vee \dots \vee v_k \models v$

- $v_1 | v_2 | \dots | v_k \longrightarrow v$

- *AND* type

- $v_1 \wedge v_2 \wedge \dots \wedge v_k \models v$

- $[v_1, v_2, \dots, v_k] \longrightarrow v$

# Causal Relationships

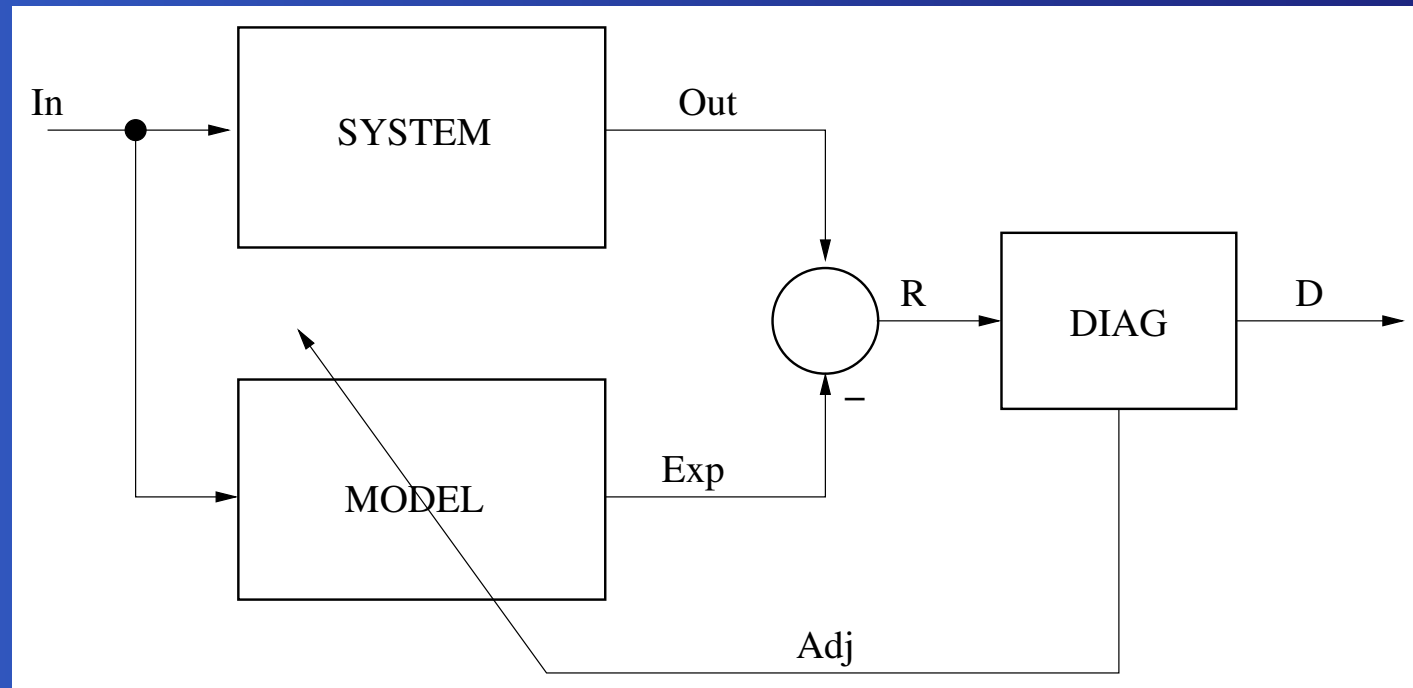
- *NOT* type

- $u \models \bar{v}$  and  $\bar{u} \models v$

- $u \xrightarrow{\bullet} v$



# Consistency-based diagnostic reasoning



■  $R$  - Residuum

# Logic and consistency-based reasoning

The formulae of predicate calculus are build of :

- terms
- relational symbols (predicates)
- logical connectors
- quantifiers
- auxiliary symbols
  - parentheses
  - comma

# Logic and consistency-based reasoning

Term :

- any constant, e.g.  $a, b, c$ , etc.
- any variable, e.g.  $X, Y, Z$ , etc.
- if  $f$  is an  $n$ -place functional symbol, and  $t_1, t_2, \dots, t_n$  are terms, then also any expression of the form  $f(t_1, t_2, \dots, t_n)$  is a term.

Nothing more is a term.

# Atomic Formulae

- $p$  is an  $n$ -place relational symbol
- $t_1, t_2, \dots, t_n$  are terms
- $p(t_1, t_2, \dots, t_n)$  is an atomic formula

Atomic formulae constitute some simple statements which can be interpreted as follows:

- "  $n$ -place relation  $p$  holds for objects  $t_1, t_2, \dots, t_n$  "

# Atomic Formulae

Logical connectives:

- conjunction -  $\wedge$
- disjunction -  $\vee$
- negation -  $\neg$
- implication -  $\Rightarrow$

# Atomic Formulae

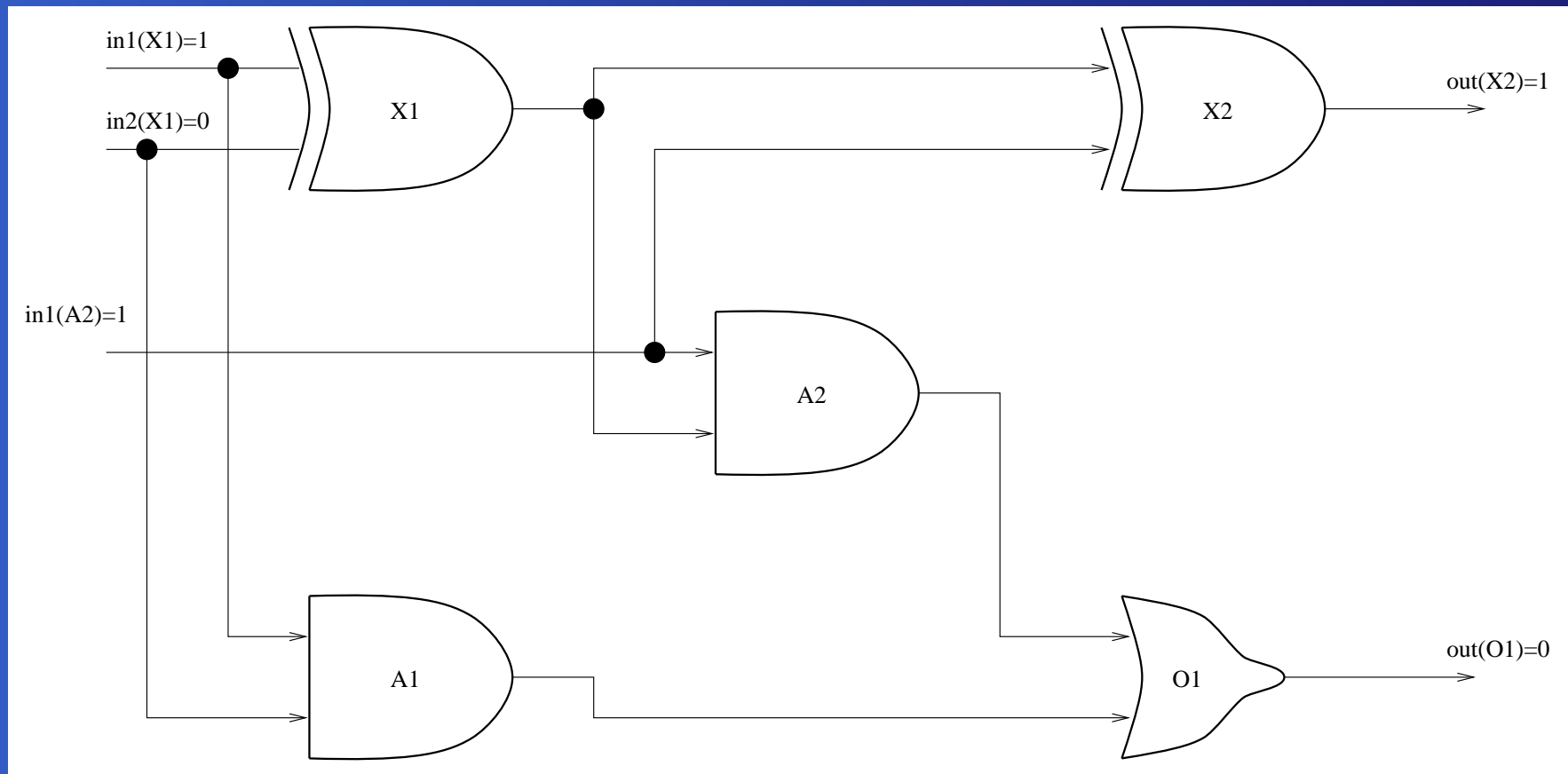
Variables should appear within the scope of some quantifier:

- universal quantifier -  $\forall$
- existential quantifier -  $\exists$

# Logical formulae

- $\models \Psi$  - tautology
- $\Psi \models \Phi$  - logical consequence
- $\not\models \Psi$  - unsatisfiable formula  
(always faulty or inconsistent)

# Full adder





# Full adder

## Components:

- $COMP = \{A1, A2, X1, X2, O1\}$

## SD - System Description:

- $and(0, 0) = 0, \quad and(0, 1) = 0, \quad and(1, 0) = 0,$   
 $and(1, 1) = 1$

- $or(0, 0) = 0, \quad or(0, 1) = 1, \quad or(1, 0) = 1,$   
 $or(1, 1) = 1$

- $xor(0, 0) = 0, \quad xor(0, 1) = 1, \quad xor(1, 0) = 1,$   
 $xor(1, 1) = 0$

# Full adder

## SD - System Description:

- $ANDG(X) \wedge \neg AB(X) \Rightarrow$   
 $out(X) = and(in1(X), in2(X))$
- $XORG(X) \wedge \neg AB(X) \Rightarrow$   
 $out(X) = xor(in1(X), in2(X))$
- $ORG(X) \wedge \neg AB(X) \Rightarrow$   
 $out(X) = or(in1(X), in2(X))$
- $ANDG(A1), ANDG(A2), XORG(X1),$   
 $XORG(X2), ORG(O1)$

# Full adder

## SD - *System Description:*

- $out(X1) = in2(A2)$
- $out(X1) = in1(X2)$
- $out(A2) = in1(O1)$
- $in1(A2) = in2(X2)$
- $in1(X1) = in1(A1)$
- $in2(X1) = in2(A1)$
- $out(A1) = in2(O1)$

# Full adder

## SD - *System Description*:

- $in1(X) = 0 \vee in1(X) = 1$   
 $in2(X) = 0 \vee in2(X) = 1$   
 $out(X) = 0 \vee out(X) = 1$

- axioms concerning equality
- axioms of Boolean algebra

# Full adder

OBS - *Observations*:

■  $in1(X1) = 1$

■  $in2(X1) = 0$

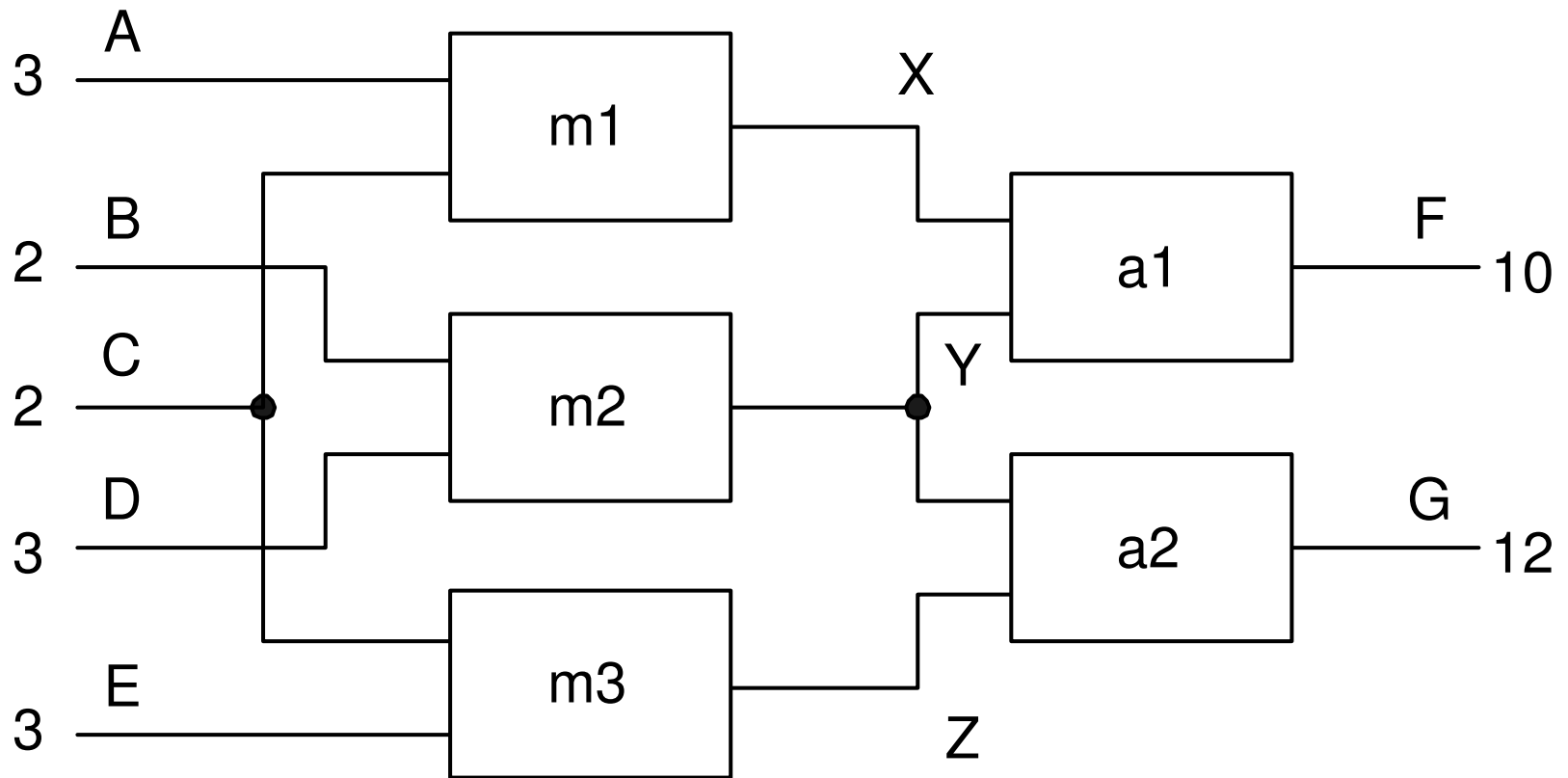
■  $in1(A2) = 1$

■  $out(X2) = 1$

■  $out(O1) = 0$

SD becomes inconsistent with OBS

# Arithmetic system



# Arithmetic system

Components:

- $COMP = \{m1, m2, m3, a1, a2\}$

SD - *System Description*:

- $ADD(x) \wedge \neg AB(x) \Rightarrow$   
 $Output(x) = Input1(x) + Input2(x)$

- $MULT(x) \wedge \neg AB(x) \Rightarrow$   
 $Output(x) = Input1(x) * Input2(x)$

- $ADD(a1), ADD(a2), MULT(m1), MULT(m2),$   
 $MULT(m3)$

# Arithmetic system

SD - *System Description*:

- $\text{Output}(m1) = \text{Input1}(a1)$
- $\text{Output}(m2) = \text{Input2}(a1)$
- $\text{Output}(m2) = \text{Input1}(a2)$
- $\text{Output}(m3) = \text{Input2}(a2)$
- $\text{Input2}(m1) = \text{Input1}(m3)$



# Arithmetic system

SD - *System Description*:

- $X = A * C, Y = B * D, Z = C * E$

- $F = X + Y, G = Y + Z$

- definitions and properties of operations of multiplication and addition

- properties of the equality relation

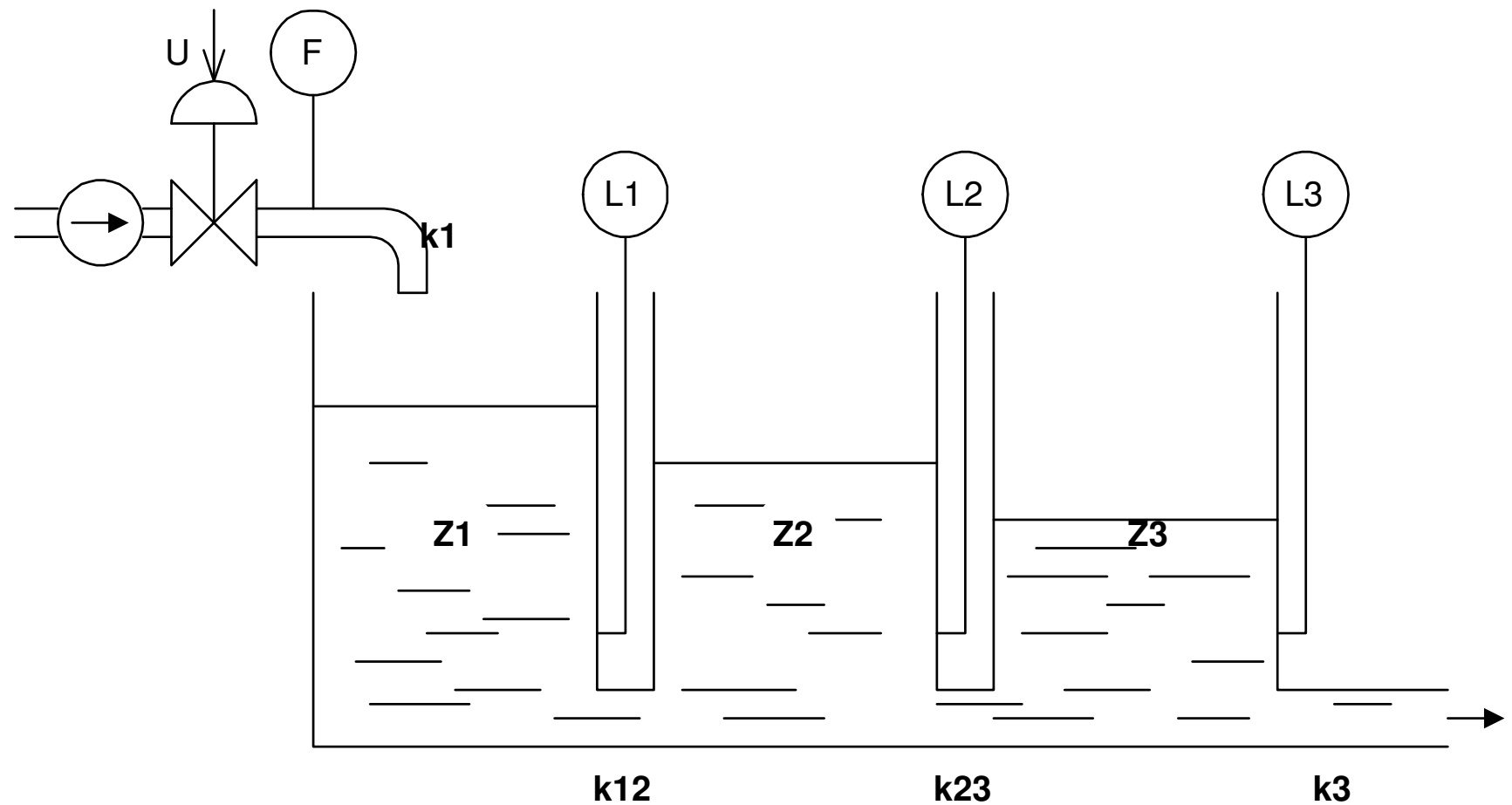
# Arithmetic system

OBS - *Observations*:

- $\text{OBS} = \{A = 3, B = 2, C = 2, D = 3, E = 3, F = 10, G = 12\}$

SD becomes inconsistent with OBS

# Three-tank system



# Three-tank system

Components:

$$\blacksquare \text{ COMP} = \{k1, k12, k23, k3, z1, z2, z3\}$$

SD - *System Description*:

$$f(U) = F \quad (1)$$

$$A_1 \frac{dL_1}{dt} = F - F_{12} \quad (2)$$

# Three-tank system

SD - *System Description:*

$$A_2 \frac{dL_2}{dt} = F_{12} - F_{23} \quad (3)$$

$$A_3 \frac{dL_3}{dt} = F_{23} - F_3 \quad (4)$$

$$F_{ij} = \alpha_{ij} C_{ij} \sqrt{2g(L_i - L_j)},$$

$$F_3 = \alpha_3 C_3 \sqrt{2gL_3}$$

# Three-tank system

SD - *System Description:*

$A_i$  - cross-sectional areas of the tanks for  
 $i = 1, 2, 3$

$C_{ij}, C_3$  - cross-sectional areas of the channels  
connecting the tanks for  $ij = 12, 23$

# Conflict sets

*Conflict set (conflict)* - any subset of the distinguished system elements, i.e.

*COMPONENTS*, such that all items belonging to such set cannot be claimed to work correctly (i.e. at least one of them must be faulty) – it is just the assumption about their correct work which leads to inconsistency.

# Conflict sets

- *SD* - System Description
- *COMPONENTS* =  $\{c_1, c_2, \dots, c_n\}$
- $\neg AB(c_1) \wedge \neg AB(c_2) \wedge \dots \wedge \neg AB(c_n)$

In the case of failure the set of formulae:

$$SD \cup \{\neg AB(c_1), \neg AB(c_2), \dots, \neg AB(c_n)\}$$

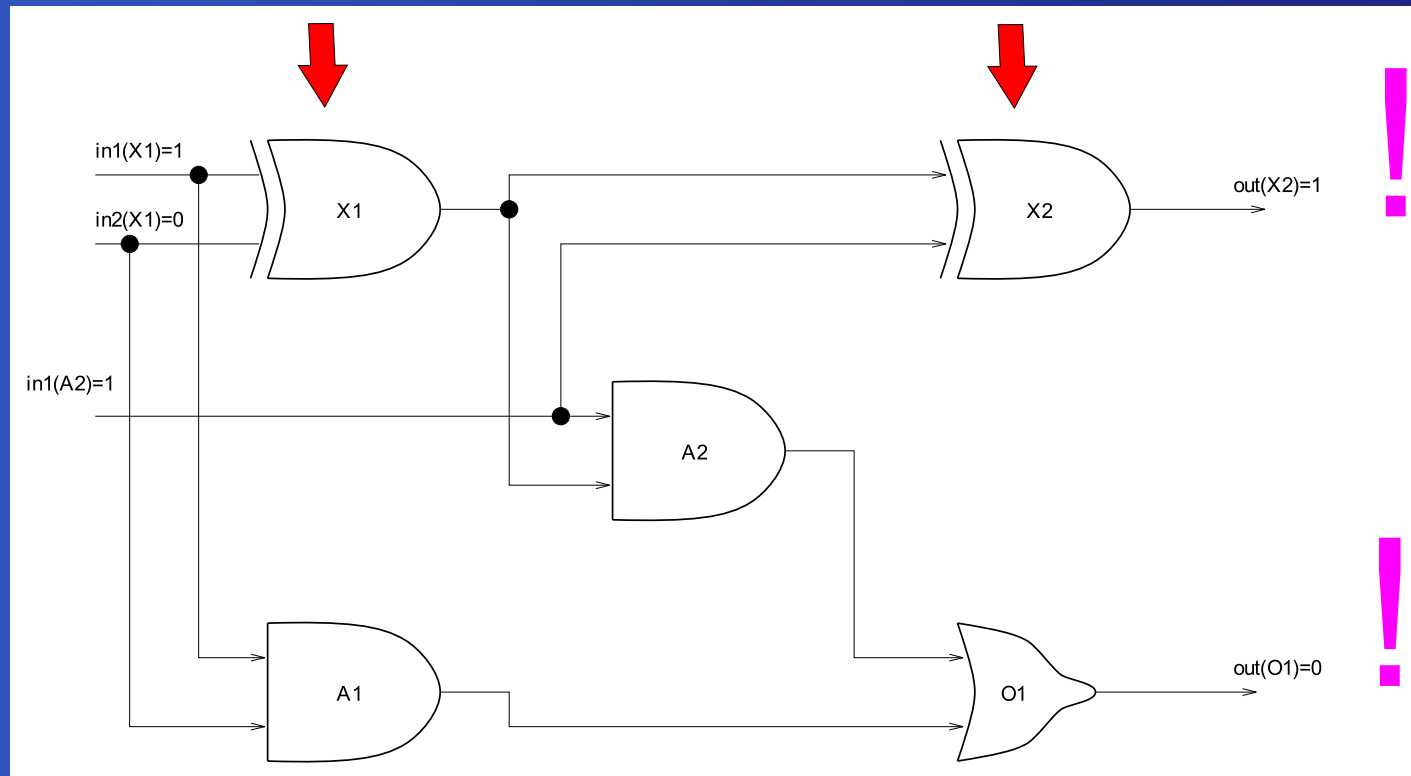
turns out to be inconsistent



# Conflict sets

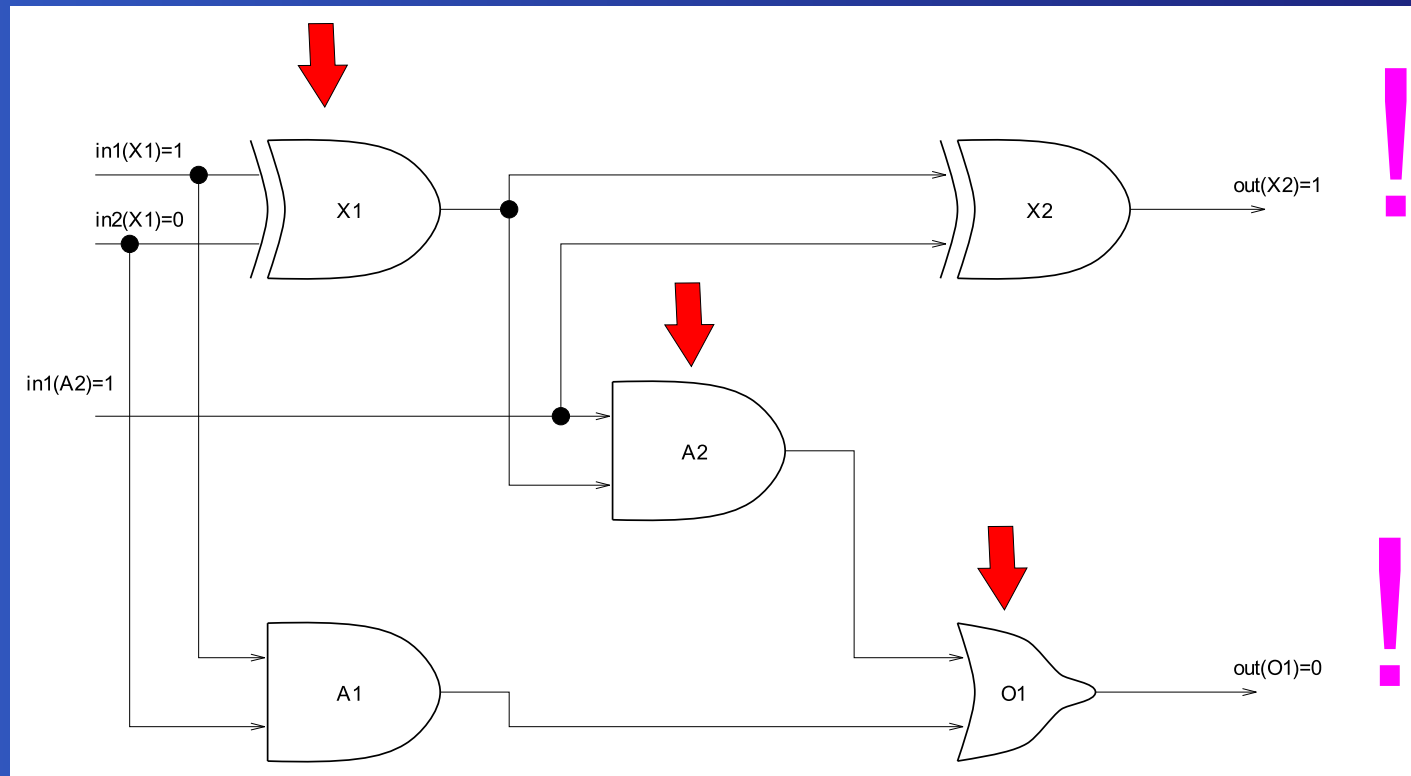
- $\{c^1, c^2, \dots, c^k\} \subseteq \text{COMPONENTS}$  - one or several sets of components such that at least one of them must have become faulty
- $AB(c^1) \vee AB(c^2) \vee \dots \vee AB(c^k)$

# Conflict sets - Full adder



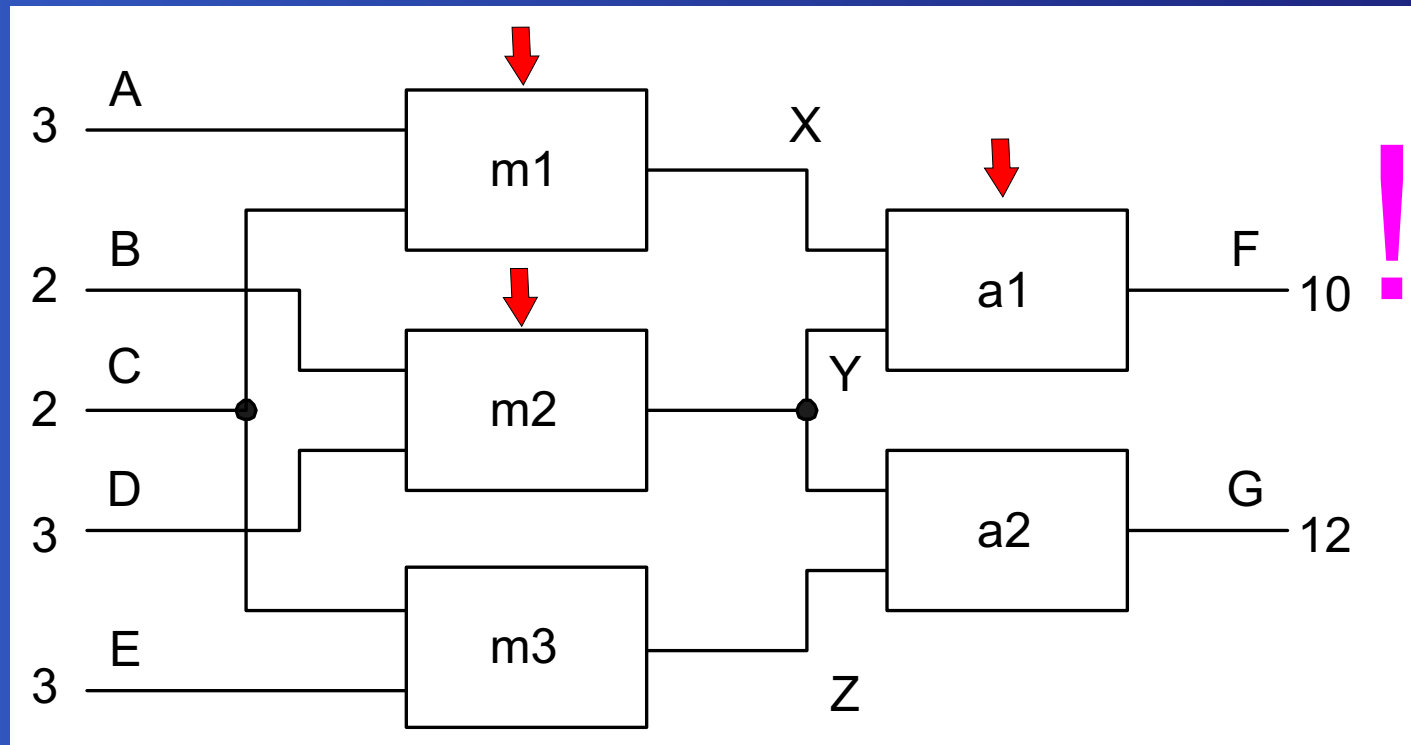
$\{X1, X2\}$

# Conflict sets - Full adder



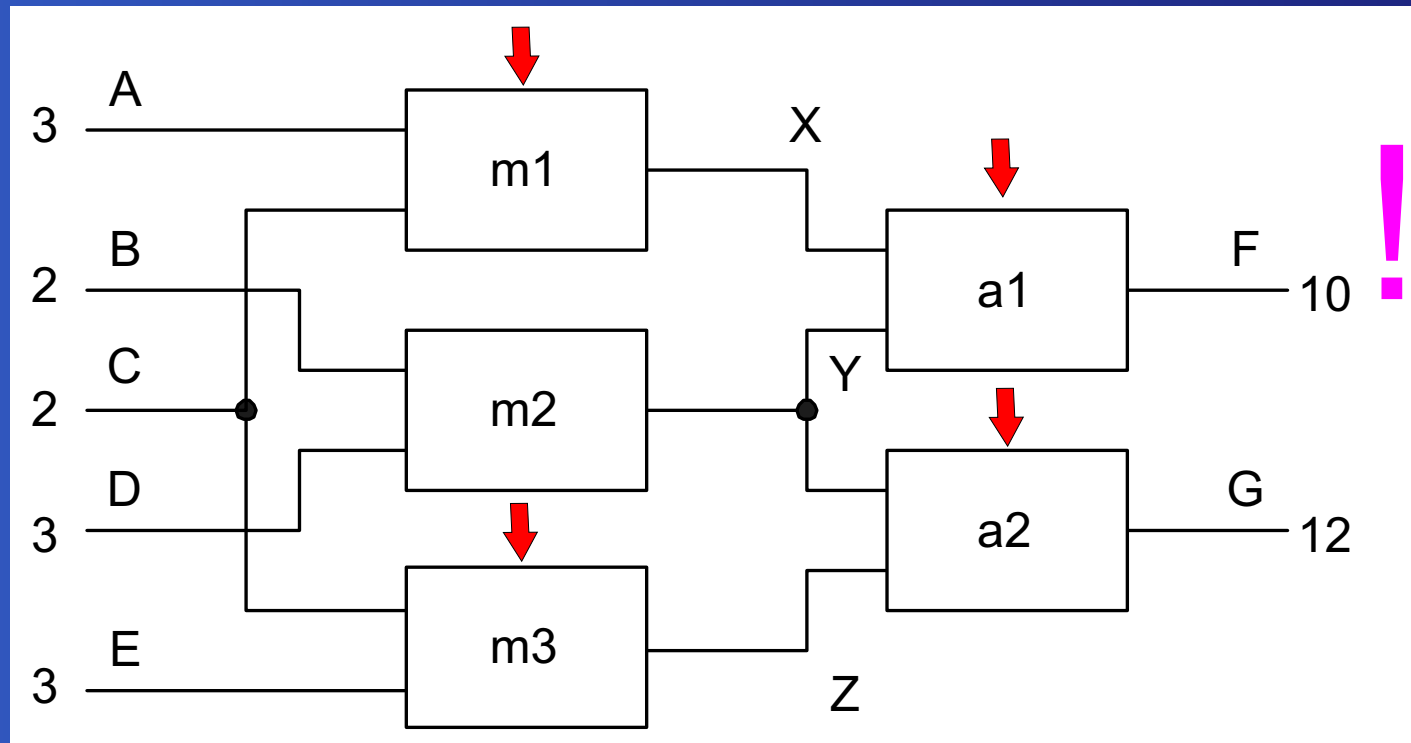
$\{X1, A2, O1\}$

# Conflict sets - Arithmetic system



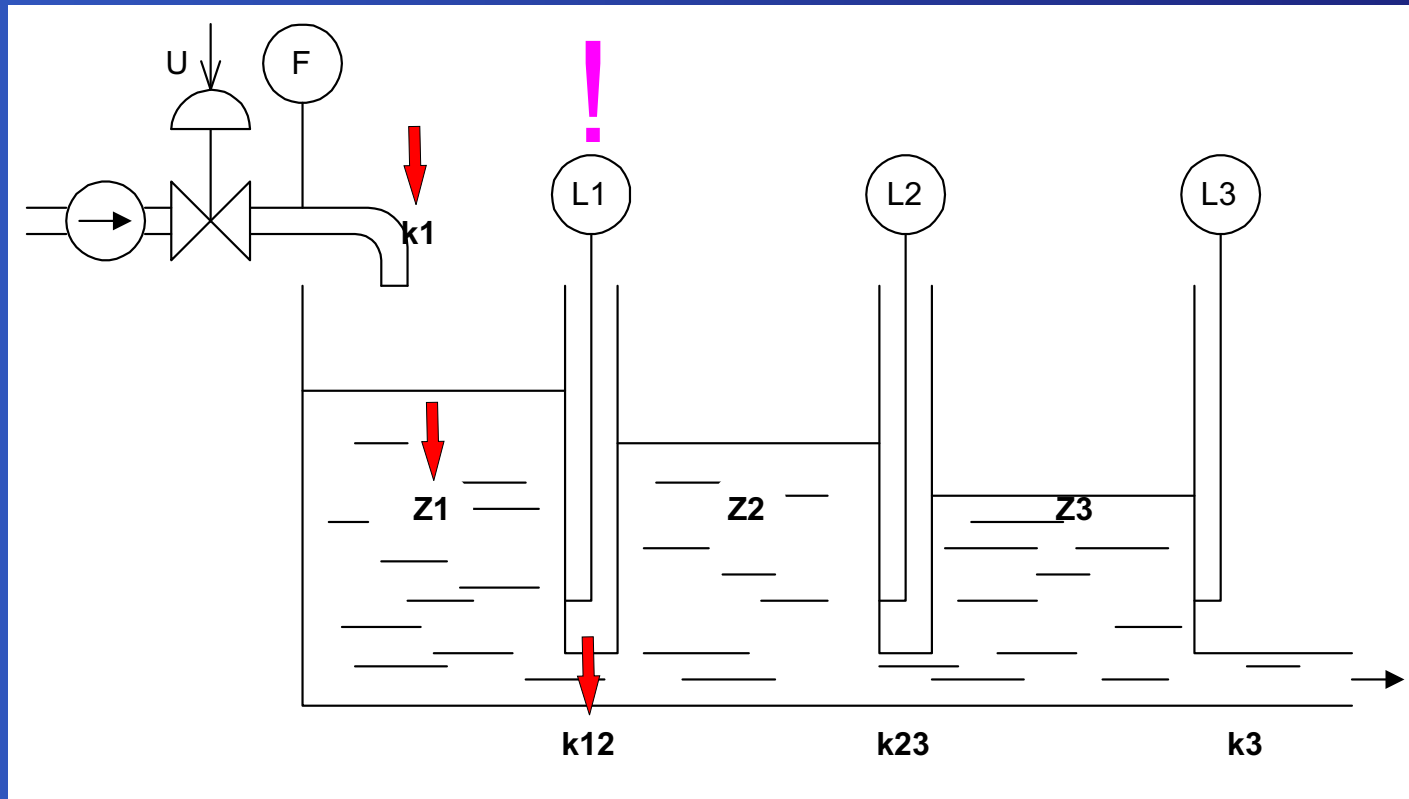
■  $\{a1, m1, m2\}$

# Conflict sets - Arithmetic system



■  $\{a1, a2, m1, m3\}$

# Conflict sets - Three-tank system



  $\{k_1, z_1, k_{12}\}$

# Reiter's theory

**Definition 1** A system is a pair (*SD*, *COMPONENTS*) where:

1. *SD* is a set of first-order predicate calculus formulae defining the system, i.e. the System Description,
2. *COMPONENTS* is a set of constants representing distinguished elements of the system.

# Reiter's theory

**Definition 2** A diagnosis for the system with observations specified by  $(SD, COMPONENTS, OBS)$  is any minimal set  $\Delta \subseteq COMPONENTS$ , such that the set

$$SD \cup OBS \cup \{AB(c) \mid c \in \Delta\} \cup$$

$$\{\neg AB(c) \mid c \in COMPONENTS - \Delta\}$$

is consistent.



# Conflict set

**Definition 3** A conflict set  $(SD, COMPONENTS, OBS)$  is any set  $\{c_1, \dots, c_k\} \subseteq COMPONENTS$ , such that

$$SD \cup OBS \cup \{\neg AB(c_1), \dots, \neg AB(c_k)\}$$

*is inconsistent.*

A conflict set is *minimal* if any of its proper subsets is not a conflict set.

# Hitting set

**Definition 4** Let  $C$  be any family of sets.  
A hitting set for  $C$  is any set

$$H \subseteq \bigcup_{S \in C} S$$

such that

$H \cap S \neq \emptyset$  for any set  $S \in C$ .

A hitting set is minimal if and only if any of its proper subsets is not a hitting set for  $C$ .

# Reiter's theory

**Thesis 1**  $\Delta \subseteq \text{COMPONENTS}$  is a diagnosis for  $(SD, \text{COMPONENTS}, OBS)$  if and only if  $\Delta$  is a minimal hitting set for the family of conflict sets for  $(SD, \text{COMPONENTS}, OBS)$ .

# Reiter's theory

**Conclusion 1**  $\Delta \subseteq \text{COMPONENTS}$  is a diagnosis for  $(SD, \text{COMPONENTS}, \text{OBS})$  if and only if  $\Delta$  is a minimal hitting set for the family of conflict sets for  $(SD, \text{COMPONENTS}, \text{OBS})$ .

# Causal graph

**Definition 5** *By a causal graph we shall understand a set of nodes representing system variables and a set of vertices describing mutual influences among the variables. The vertices of the graph are assigned equations describing the influences in a quantitative way, and the variables are assigned some domains.*

# Causal graph

- $A, B, C, \dots$  – measurable variables,
- $[U], [V], [W]$  – unmeasurable variables,
- $X^*$  – conflicting variable, i.e. one taking the value inconsistent with model-based prediction,

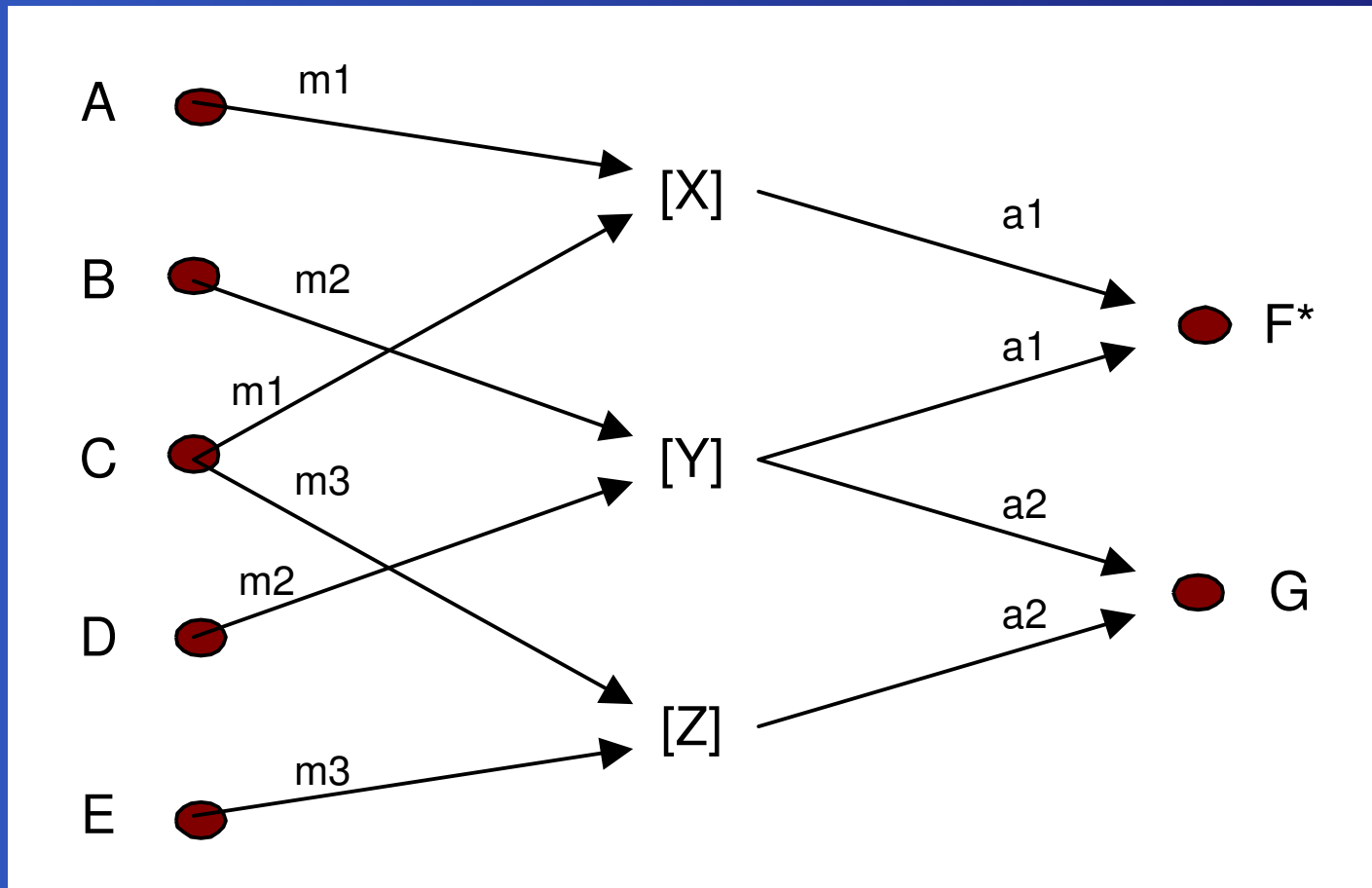
and let ( $\longrightarrow$ ) denote the existence of causal influence between two variables. Any such influence is assigned an expression of the form:

$$i = \left( [X_1, X_2, \dots, X_k], f, Y, [c_1, c_2, \dots, c_k, c_Y] \right)$$

# Causal graph

- $X_1, X_2, \dots, X_k$  – input variables
- $f$  – function
- $Y$  – output variable
- $c_1, c_2, \dots, c_k$  – the system components responsible for correct work of the subsystems generating the output values
- $c_Y$  – the component responsible for the value of the output variable  $Y$

# Causal graph - Arithmetic system





# Causal graph

- existence of all the conflicts is indicated by misbehavior of some variables (behavior different from the predicted one),
- in order to state that a conflict exists the current value of it (observed or measured) must be different from the one predicted with use of the model,
- the conflict set will be composed of the components responsible for the correct value of the misbehaving variable.

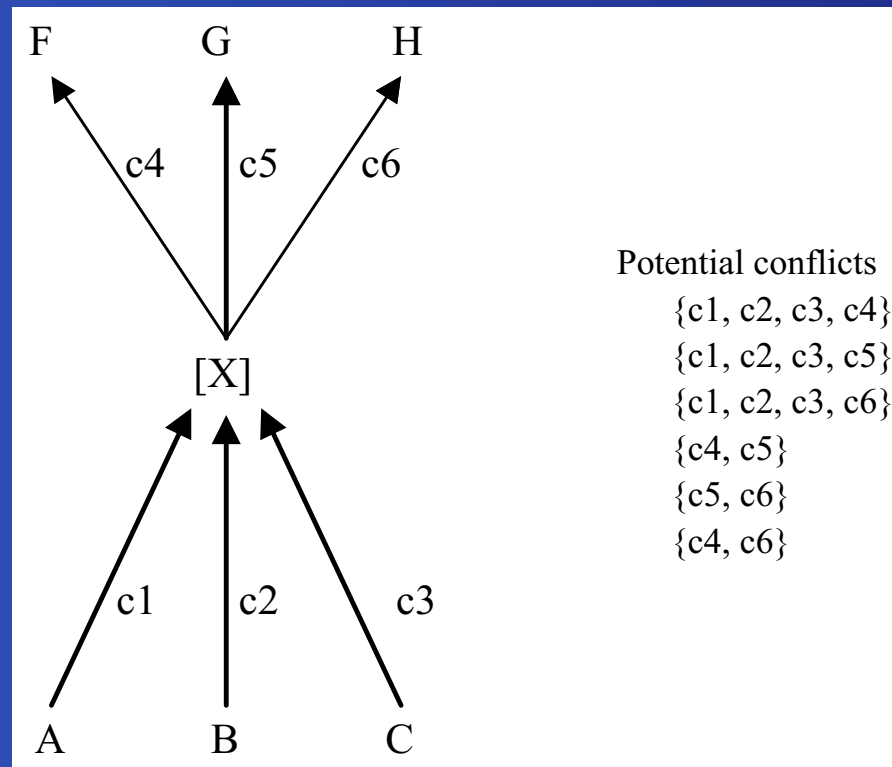
# Causal graph

**Definition 6** *A PCS structure defined for variable  $X$  on  $m$  variables is any subgraph of the causal graph, such that:*

- *it contains exactly  $m$  variables (including  $X$ ),*
- *the values of all the variables are measured or calculated (they are well-defined),*
- *the value of variable  $X$  is double-defined,*
- *in the considered PCS all the values of the  $m$  variables are necessary for  $X$  in order to be double-defined.*

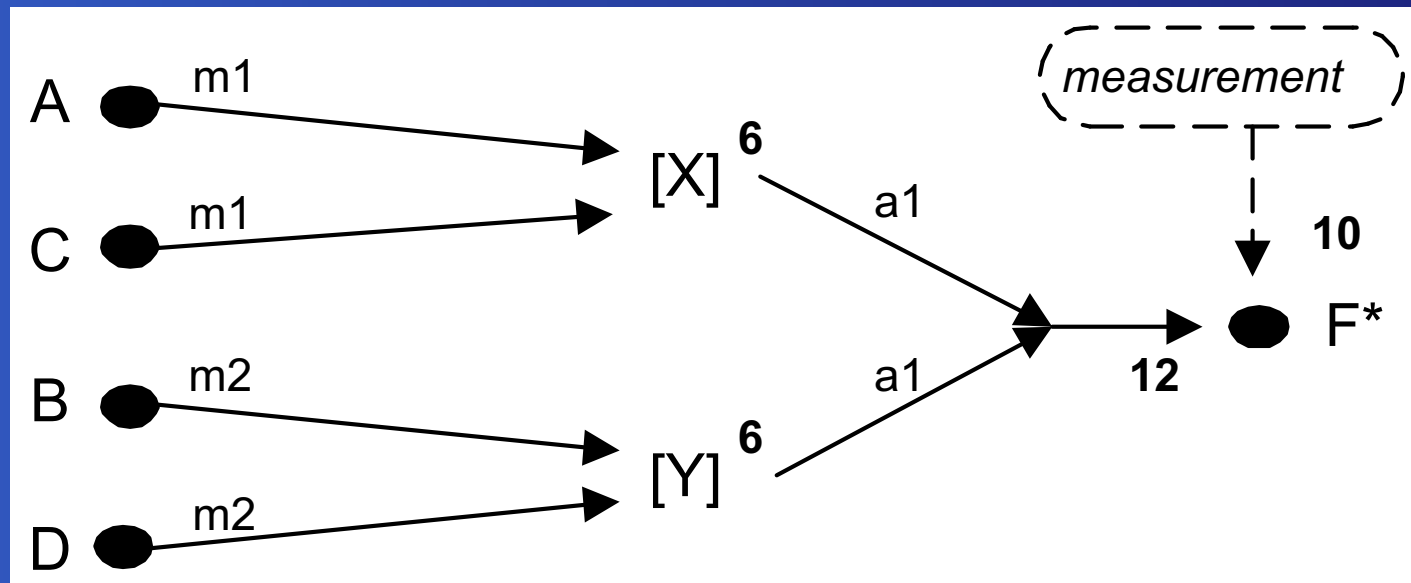
# Example PCS

## Arithmetic system



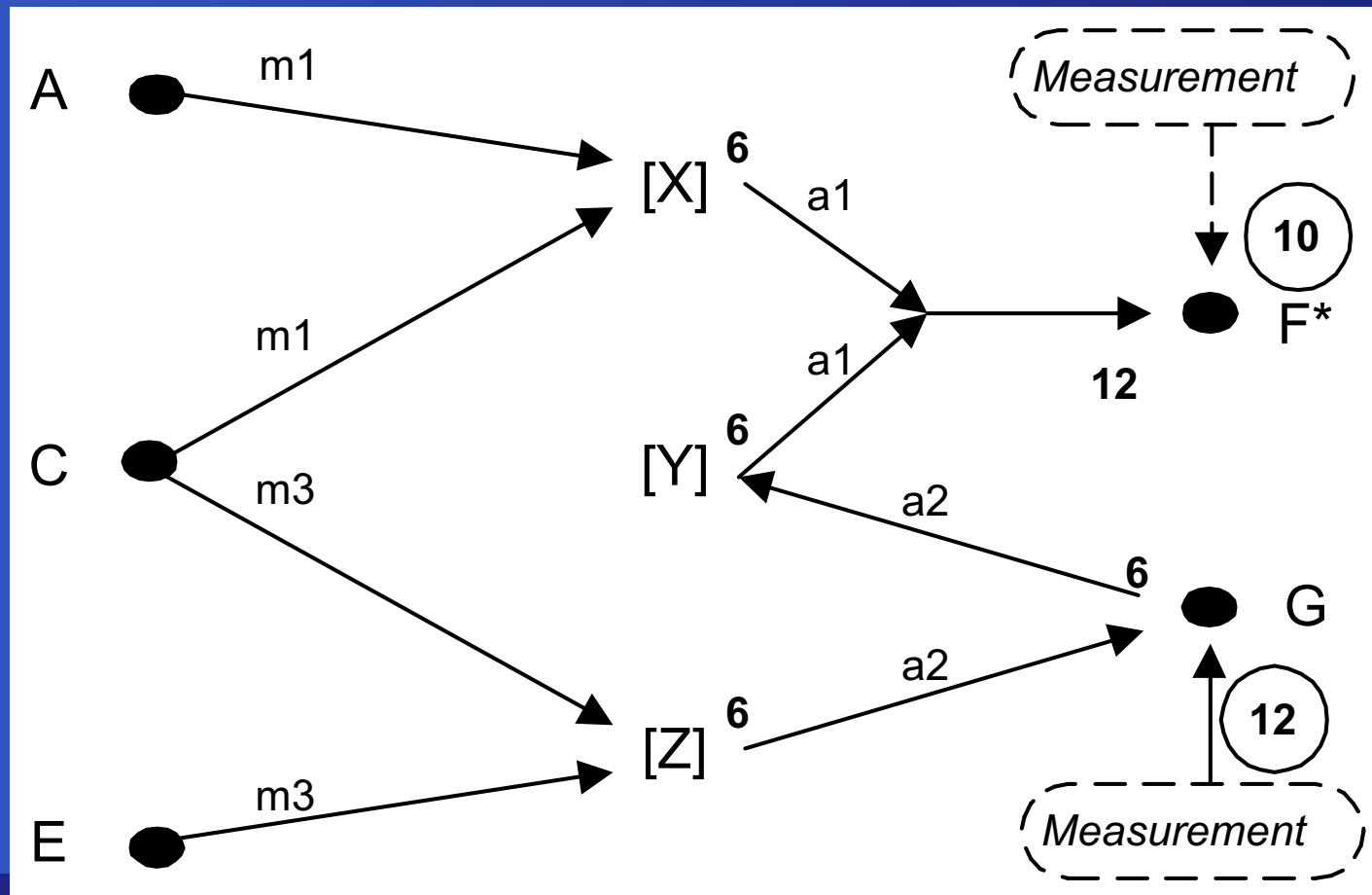
# Example PCS

## Arithmetic system

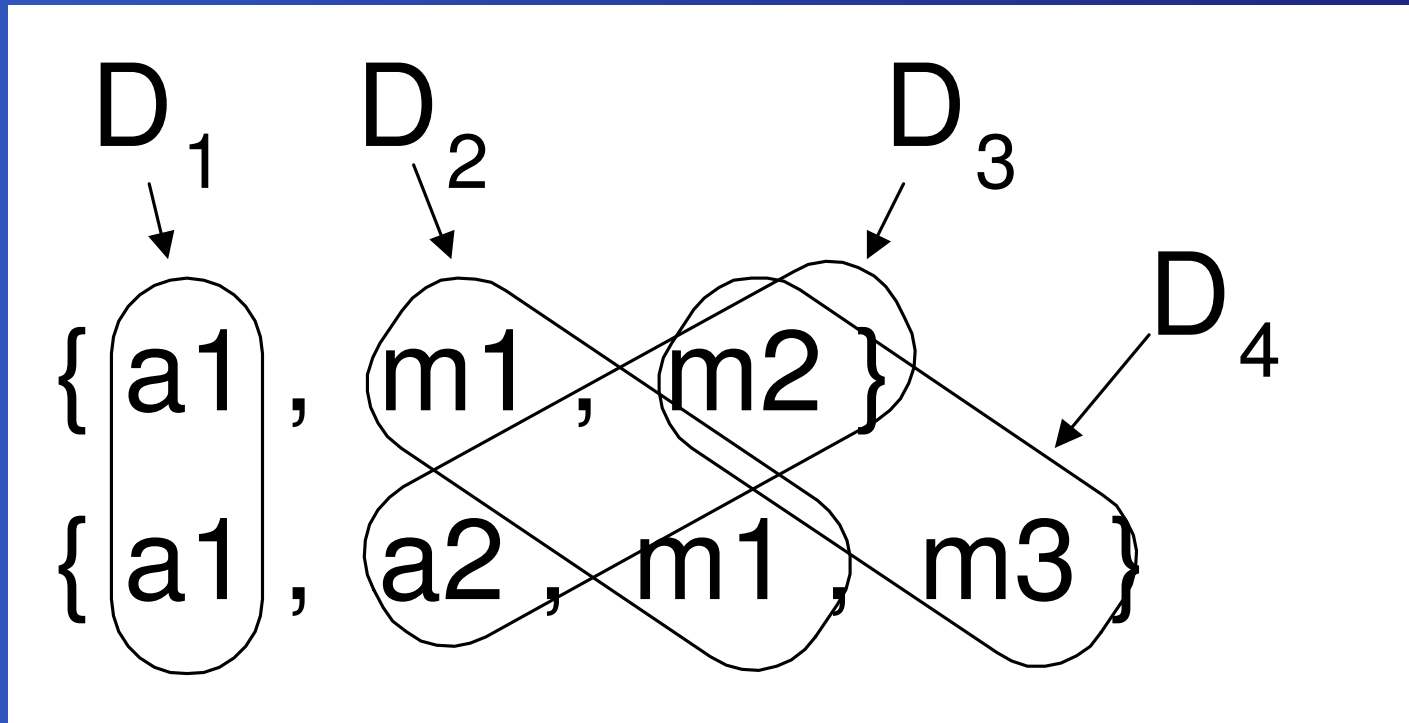


# Example PCS

## Arithmetic system



# Minimal hitting sets - Arithmetic system



# Logical causal graphs

- Nodes - symptoms
- Vertices - causal relationships

# Logical causal graphs

Let  $\mathbf{N}$  denote a finite set of symptoms describing behavior of the analyzed system. It is further assumed that this set is composed of three disjoint subsets, namely  $\mathbf{M}$ ,  $\mathbf{V}$  and  $\mathbf{D}$ ,  $\mathbf{N} = \mathbf{D} \cup \mathbf{V} \cup \mathbf{M}$ , where:

- $\mathbf{D}$  is a set of *primary symptoms*
- $\mathbf{M}$  is a set of *terminal symptoms*
- $\mathbf{V}$  is a set of *intermediate symptoms*

Let  $\Psi$  denote the set of all the causal dependencies among the symptoms of  $\mathbf{N}$



# Logical causal graphs

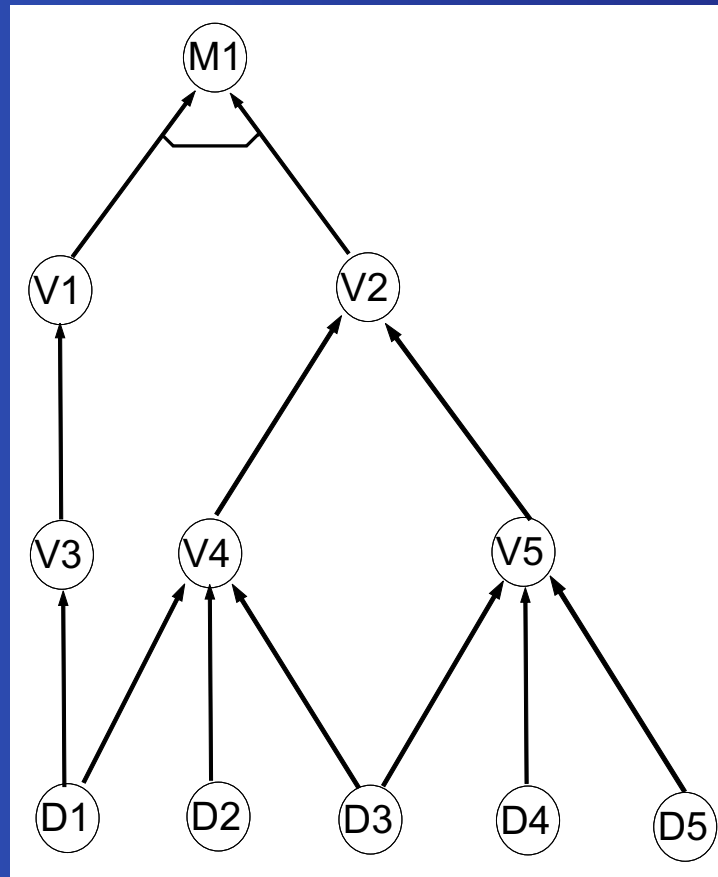
**Definition 7** *A logical causal graph is a pair*  
 $G = (N, \Psi)$ .

# The AND/OR/NOT graphs

**Definition 8** *A logical causal graph of the AND/OR/NOT type is a pair  $G = (\mathbf{N}, \Psi)$ , where  $\Psi = \{AND, OR, NOT\}$ ; the logical functors allow any finite number of arguments.*

- The graph does not contain cycles
- The nodes of  $\mathbf{D}$  do not have ancestors
- The nodes of  $\mathbf{M}$  do not have successors

# The AND/OR/NOT graph -Example



# The AND/OR/NOT graphs

- $X^+$  – any set of symptoms assigned *true*
- $X^-$  – any set of symptoms assigned *false*

# The AND/OR/NOT graphs

- $G$  is an *AND/OR/NOT* causal graph
- Distinguished sets of terminal symptoms (manifestations):
  - $M^+ \subseteq M$  – true
  - $M^- \subseteq M$  – false
- Sets of some other symptoms (observations):
  - $N^+ \subseteq N$  – true
  - $N^- \subseteq N$  – false

# The AND/OR/NOT graphs

**Definition 9** *Let  $G$  be an AND/OR/NOT causal logical graph. A diagnostic problem is any five-tuple of the form:*

$$(G, M^+, M^-, N^+, N^-)$$

# The AND/OR/NOT graphs - nodes

- *AND* node –  $[n_1, n_2, \dots, n_i] \longrightarrow n$
- *OR* node –  $n_1 | n_2 | \dots | n_i \longrightarrow n$
- *NOT* node –  $n \longrightarrow \bullet n'$

# Backward inference rules

- **OR node, false:** if a node  $n$  of type **OR** is false, then all its predecessors  $n_1, n_2, \dots, n_i$  should be set to *false*
- **AND node, true:** if a node  $n$  of type **AND** is true, then all its predecessors  $n_1, n_2, \dots, n_i$  should be set to *true*



# Backward inference rules

- *NOT* node, true: if a node  $n'$  of type *NOT* is true, then its predecessor  $n$  should be set to *false*
- *NOT* node, false: if a node  $n'$  of type *NOT* is false, then its predecessor  $n$  should be set to *true*

# Forward inference rules

- *OR* node, true: if at least one of the predecessors  $n_k \in \{n_1, n_2, \dots, n_i\}$  of an *OR* is *true*, then node  $n$  should be set to *true*,
- *AND* node, false: if at least one of the predecessors  $n_k \in \{n_1, n_2, \dots, n_i\}$  of an *AND* node is false, then node  $n$  should be set to *false*

# Forward inference rules

- *NOT* node, true: if the predecessor  $n$  of a *NOT* node is false, then node  $n'$  should be set to *true*
- *NOT* node, false: if the predecessor  $n$  of a *NOT* node is true, then node  $n'$  should be set to *false*

# Forward inference rules

- **OR** node, false: if all the predecessors  $n_1, n_2, \dots, n_i$  of an **OR** node are *false*, then node  $n$  should be set to *false*
- **AND** node, true: if all the predecessors  $n_1, n_2, \dots, n_i$  of an **AND** node are *true*, then node  $n$  should be set to *true*.

# The AND/OR/NOT graphs - state

The current state of the graph is defined by the pair of sets  $(S^+, S^-)$  containing all the true and false symptoms defined in the graph, and being the fixed point of the operation of state propagation.

# Diagnostic reasoning

Abduction:

$$\frac{\beta, \alpha \Rightarrow \beta}{\alpha}.$$

# Abductive inference rules

- *OR* node, true: if an *OR*  $n$  is true, then *at least one* of its predecessors  $n_k \in \{n_1, n_2, \dots, n_i\}$  must be true; if not, one of them must be selected and set to *true*:

$$\frac{n, \quad n_1 \vee n_2 \vee \dots \vee n_i \Rightarrow n}{n_k}.$$

# Abductive inference rules

- *AND* node, false: if an *AND* node is false, then at least one of its predecessors  $n_k \in \{n_1, n_2, \dots, n_i\}$  must be false; if not, one of them must be selected and set to *false*:

$$\frac{\neg n, \quad n_1 \wedge n_2 \wedge \dots \wedge n_i \Rightarrow n}{\neg n_k}.$$



# Diagnostic reasoning

Let  $D = (D^+, D^-)$  denote some assignment of logical values to elementary diagnoses. If state  $S = (S^+, S^-)$  can be obtained from  $D$  with use of the propagation rules, this fact is denoted as  $D \vdash S$ . Since all the information propagation rules represent valid logical inference, if  $D \vdash S$  then also  $S$  is a logical consequence of  $D$  ( $D \models S$ ). Hence,  $D$  constitutes a correct explanation of state  $S$ .

# Diagnostic reasoning

**Definition 10** *Let  $(G, M^+, M^-, N^+, N^-)$  be a diagnostic problem. A diagnosis  $D$  (a solution of the diagnostic problem) is any pair of the form  $(D^+, D^-)$  of the sets of input symptoms true and false (of elementary diagnoses assigned true or false, respectively), satisfying the following conditions:*

- $(D^+, D^-) \vdash (M^+, M^-)$ , i.e. the diagnosis explains all the symptoms indicating fault,

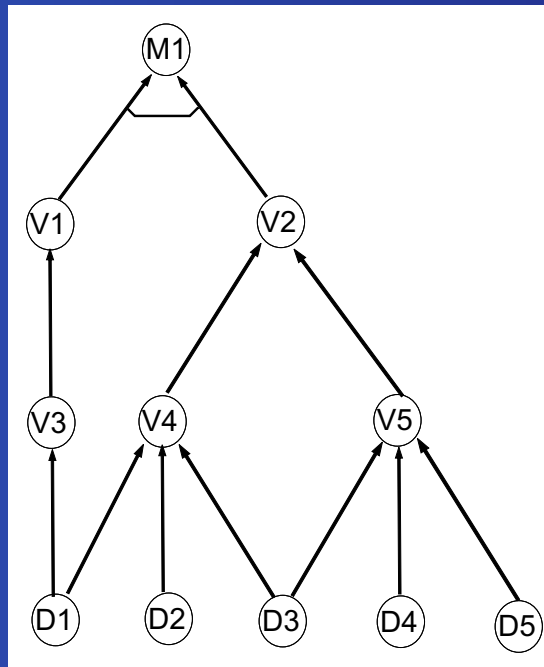
# Diagnostic reasoning

- if  $(S^+, S^-)$  is a state implied by diagnosis  $(D^+, D^-)$ , then such state must be consistent, i.e.  $S^+ \cap S^- = \emptyset$ ,
- each such state is consistent with the observations, i.e.  $N^+ \cap S^- = \emptyset$  and  $N^- \cap S^+ = \emptyset$ .

Moreover, most frequently the analysis is restricted to *minimal diagnoses*, i.e. such that any pair of sets  $(D_0^+, D_0^-) \neq (D^+, D^-)$ , where  $D_0^+ \subseteq D^+$  and  $D_0^- \subseteq D^-$  cannot be a diagnosis.

# The AND/OR/NOT graph

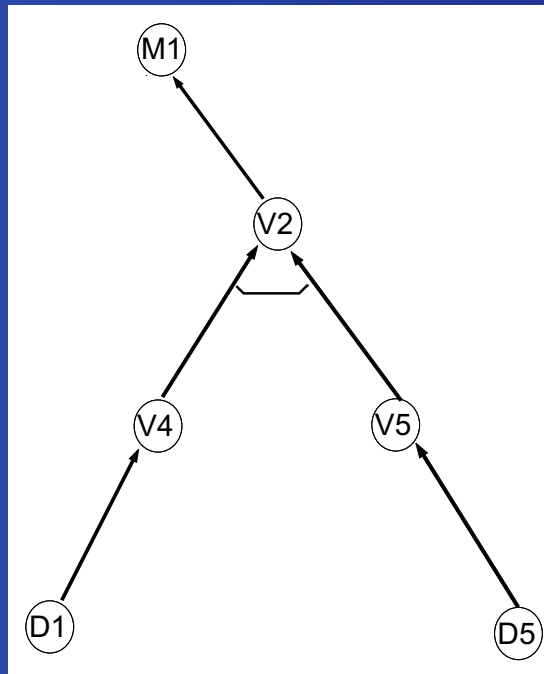
Example problem:



■  $M^+ = \{m_1\}, M^- = \emptyset, N^+ = \emptyset, N^- = \emptyset$

# The AND/OR/NOT graph

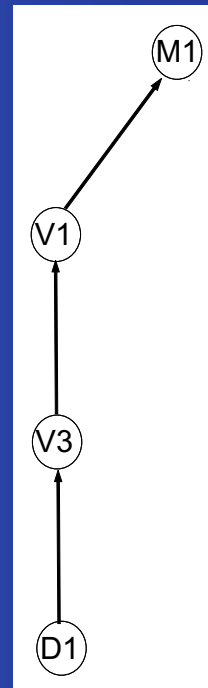
Example solution:



■  $D_1 = (D^+ = \{d_1, d_5\}, D^- = \emptyset)$

# The AND/OR/NOT graph

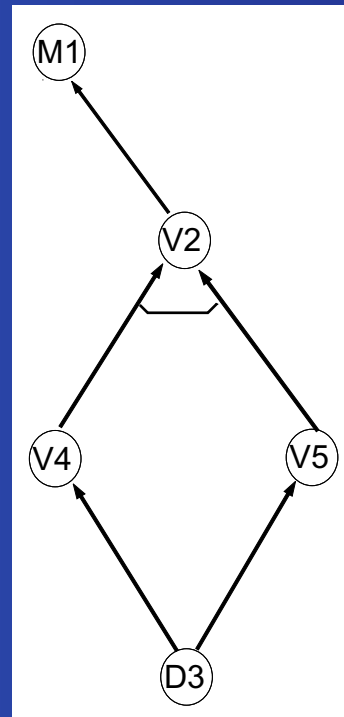
Example solution:



■  $D_2 = (D^+ = \{d_1\}, D^- = \emptyset)$

# The AND/OR/NOT graph

Example solution:



■  $D_3 = (D^+ = \{d_3\}, D^- = \emptyset)$

# Analysis and verification

- Typically, it is assumed that single fault diagnoses representing faults of a single system component are more likely than the multiple fault ones. The first step of analysis of potential diagnoses can consist then in selection of the singular diagnoses and verification of them.



# Analysis and verification

If we have  $V^+$  and  $V^-$ , then:

- elimination of certain diagnoses;  
if  $D$  is a diagnosis  $S = (S^+, S^-)$  denotes the state of the graph implied by this diagnosis ( $D \vdash S$ ), then diagnosis  $D$  may be eliminated if it is inconsistent with the auxiliary data, i.e. there is  $S^+ \cap V^- \neq \emptyset$  or  $S^- \cap V^+ \neq \emptyset$ ,

# Analysis and verification

- confirmation of certain diagnoses; the degree of confirmation can be calculated as total number of elements in the sets  $S^+ \cap V^+$  and  $S^- \cap V^-$ , obviously if the diagnosis is not eliminated due to inconsistency as described above.

# Analysis and verification

$D_1, D_2, \dots, D_l$  – the generated diagnoses,  
 $D_i = (D_i^+, D_i^-)$ , where the sets  $D_i^+$  and  $D_i^-$   
contain some number of elements. It is also  
assumed that only minimal diagnoses are  
considered and that the diagnoses are consistent  
( $D_i^+ \cap D_i^- = \emptyset$ )

# Analysis and verification

- Two diagnoses  $D_i = (D_i^+, D_i^-)$  and  $D_j = (D_j^+, D_j^-)$  are inconsistent if  $D_i^+ \cap D_j^- \neq \emptyset$  or  $D_i^- \cap D_j^+ \neq \emptyset$
- $d$  *conflict element* (in one of the diagnoses it is true, and in the other one it is false)

# Analysis and verification

- $n^+(d)$  – the number of diagnoses in which  $d$  occurs taking the value true (i.e. it belongs to  $D_i^+$ )
- $n^-(d)$  – the number of diagnoses in which it is false (i.e. it belongs to  $D_i^-$ )

# Analysis and verification

- $r(d) = \min(n^+(d), n^-(d))$  – The smallest number of eliminated diagnoses
- $d^*$  – the elementary diagnosis selected for verification
- $r(d^*) = \max_{d \in D_1, D_2, \dots, D_l}(r(d))$

# Extensions of the basic formalism

- Diagnostic tests
- Probabilities of symptoms:
  - classical probabilities
  - qualitative probabilities
- Functional causal graphs

# Diagnostic tests

The result of test  $t$  made in state  $S = (S^+, S^-)$  and under the assumption of existence of unknown fault is a new state  $S_t = (S_t^+, S_t^-)$ . Since the goal of such a test consists in obtaining new information, at least one of the following relations should hold:

- $S^+ \subset S_t^+$

- $S^- \subset S_t^-$

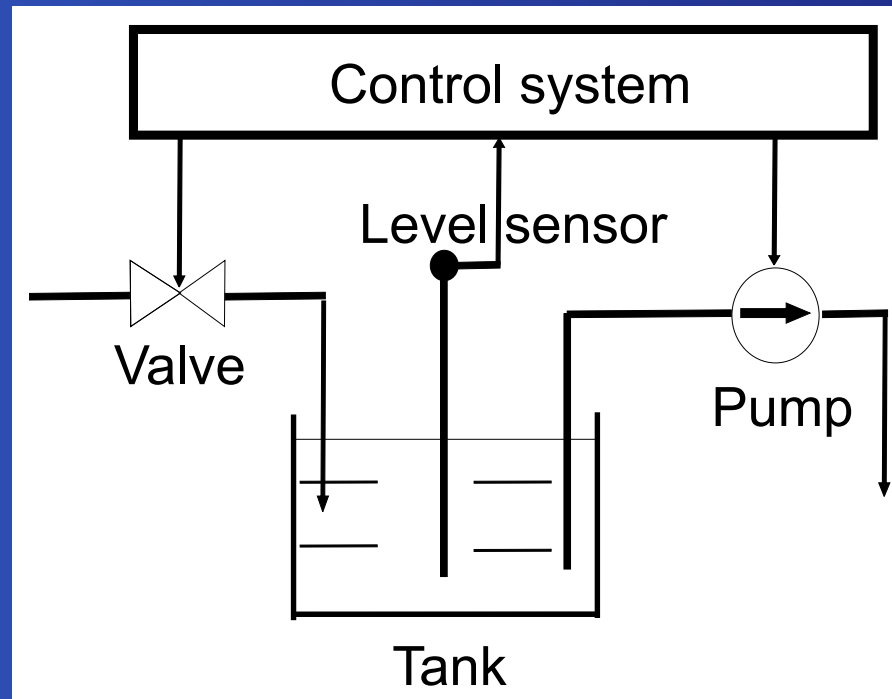


# Functional causal graphs

Nodes of such graphs are no longer restricted to represent logical variables; in fact, they can take more than two values. Example:

- A variable taking three qualitative values, e.g.  $\{-, 0, +\}$ 
  - 0 denotes nominal state
  - $-$  denotes deviation below the nominal value
  - $+$  denotes deviation above the nominal value

# An example system



Manifestations:  
 $m$  – tank overflow

# An example system

Intermediate symptoms:

*v1* – valve\_open,

*v2* – pump\_off,

*v3* – valve\_stuck\_in\_open\_position,

*v4* – valve\_open\_by\_control\_signal,

*v5* – pump\_off\_by\_power\_off,

*v6* – pump\_off\_control\_signal,

*v7* – pump\_blocked,

*v8* – pump\_on\_by\_control\_signal,

*v9* – valve\_open\_signal,

*v10* – pump\_on\_signal\_from\_level\_sensor,

# An example system

Intermediate symptoms:

*v11* –

pump\_on\_signal\_from\_control\_system,

*v12* – power\_off,

*v13* –

valve\_open\_signal\_from\_level\_sensor

# An example system

Input symptoms - elementary diagnoses:

*d1* - valve\_stuck\_in\_open\_position\_fault,

*d2* - valve\_open\_control\_signal\_on,

*d3* -

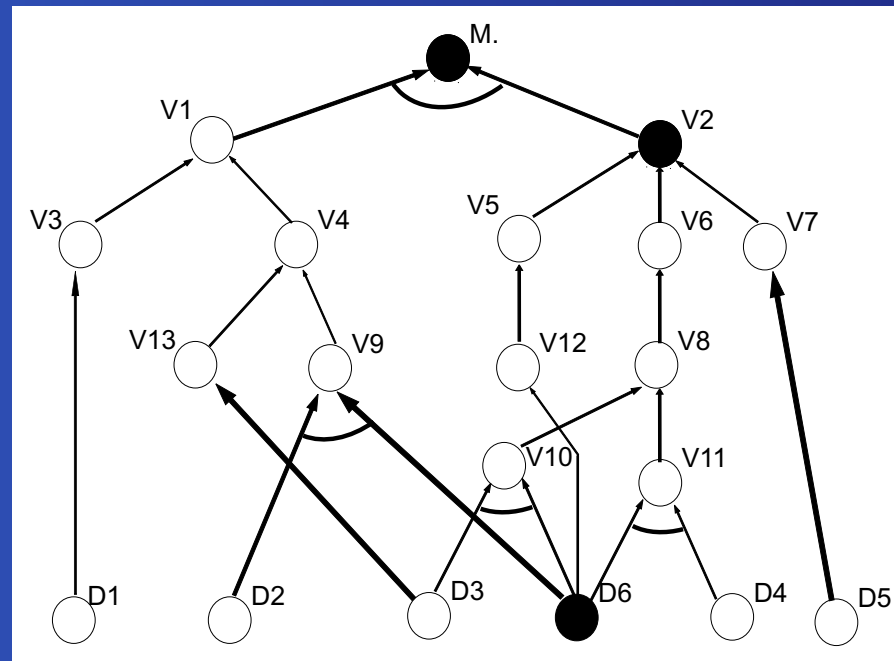
level\_sensor\_on\_when\_level\_too\_high,

*d4* - pump\_on\_by\_control,

*d5* - pump\_broken\_down\_fault,

*d6* - power\_on.

# Logical causal graph



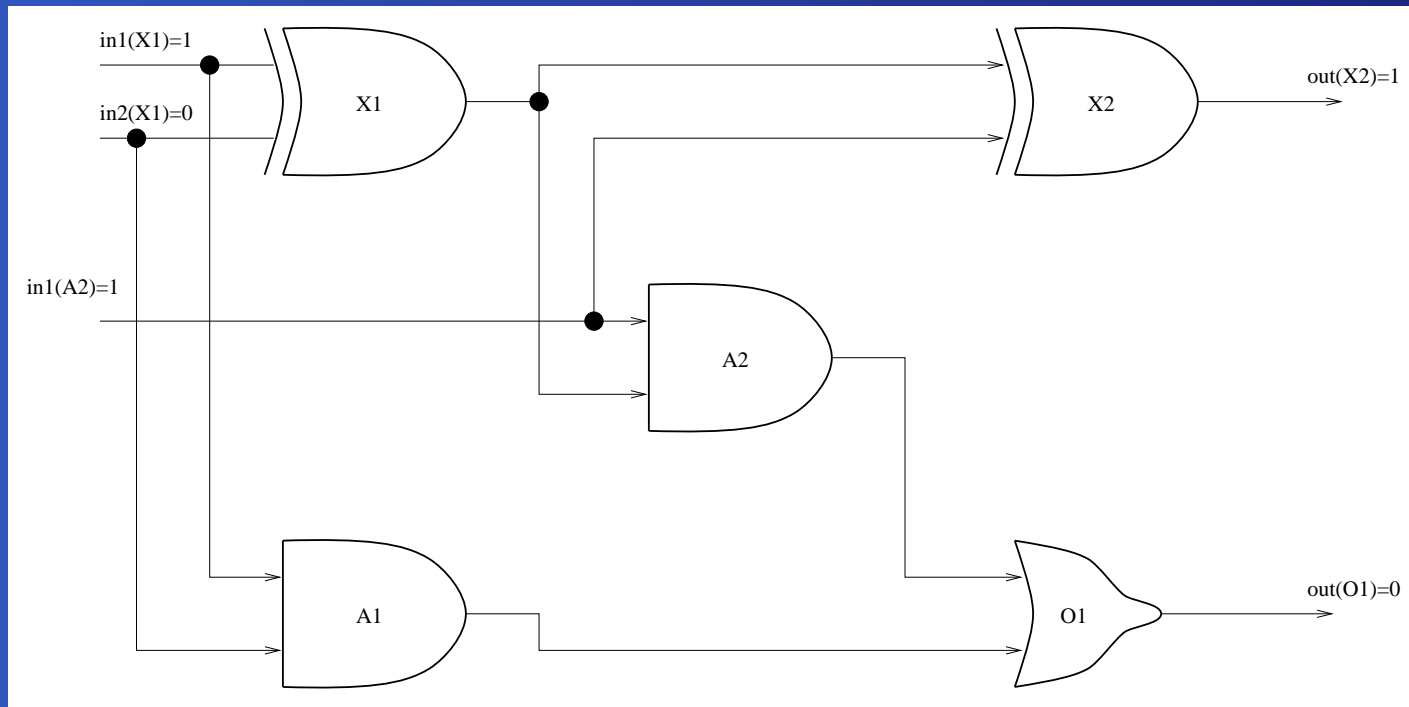
■  $M^+ = \{m\}$

■  $N^+ = \{v2, d6\}$

# Diagnoses

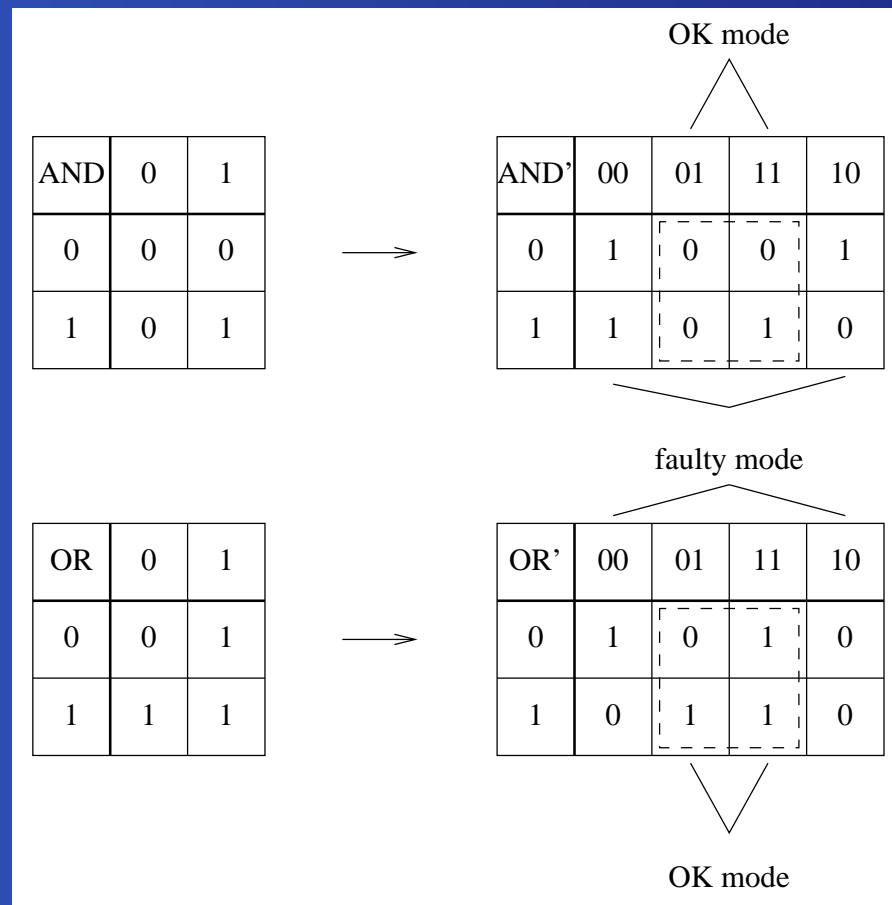
1.  $D_1 = (\{d1\}, \{d3, d4\})$
2.  $D_2 = (\{d1, d5\}, \{\})$  – minimal diagnose
3.  $D_3 = (\{\}, \{d3, d4\})$  – minimal diagnose
4.  $D_4 = (\{d5\}, \{d3\})$  – minimal diagnose
5.  $D_5 = (\{d2, d6\}, \{d3, d4\})$
6.  $D_6 = (\{d2, d5, d6\}, \{\})$  – minimal diagnose

# Full adder





# Full adder



# Full adder - diagnoses

$$\begin{aligned} D_1^+ &= \{O1, X2\}, & D_1^- &= \{X1, A2\} \\ D_2^+ &= \{O1, A1, X2\}, & D_2^- &= \{X1\} \\ D_3^+ &= \{A2, X2\}, & D_3^- &= \{X1, A1, O1\} \\ D_4^+ &= \{X1, O1, A2\}, & D_4^- &= \{X2\} \\ D_5^+ &= \{X1, X1, A1\}, & D_5^+ &= \{X2\} \\ D_6^+ &= \{X1\}, & D_6^- &= \{A1, A2, O1, X2\} \end{aligned}$$