# General Diagnostic Engine: GDE

## Consistency based diagnosis

1 Introduction

2 Computational approach: GDE

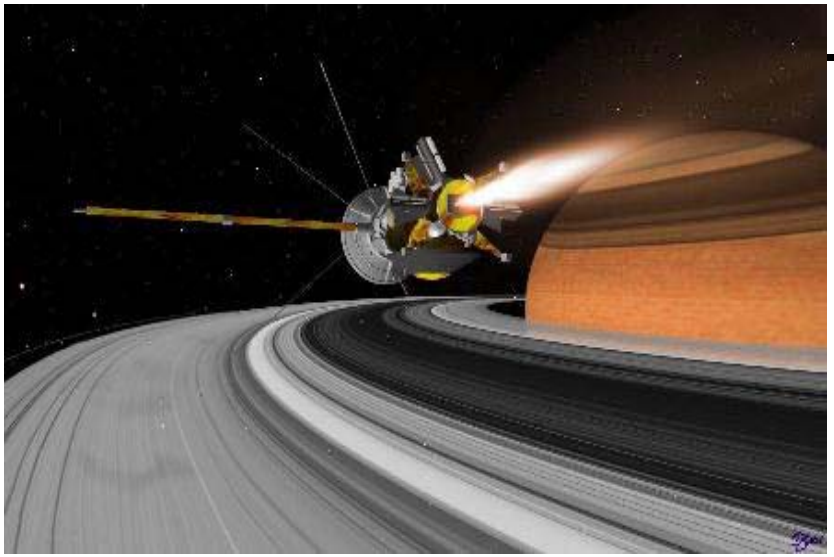
Carlos Alonso González – Belarmino Pulido Junquera

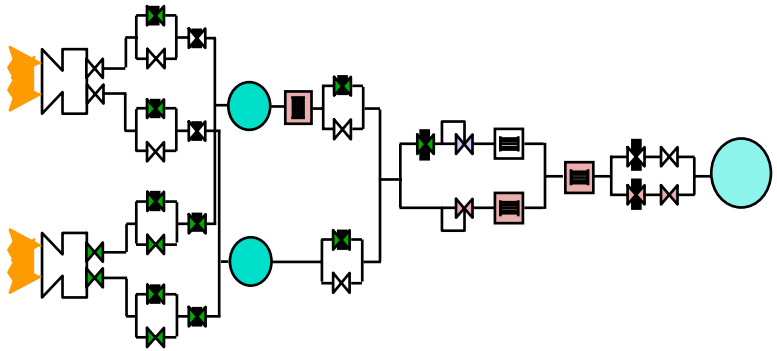
GSI. Dpto. Informática, Universidad de Valladolid

# Consistency Based Diagnosis

- Main Model Based Diagnosis framework from DX community
- Component oriented
  - May be extended to processes
- Knowledge: structure + behavioural (local) models of components
- Only models of correct behaviour

# Basic Assumptions (de Kleer 03)

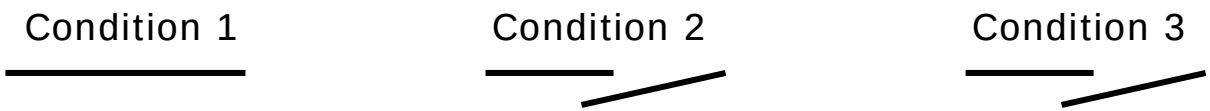




- Physical system
  - Set of interconnected components
  - Known desired function
  - Design achieves function
  - System is correct instance of design
- All malfunctions caused by faulty component(s)
- Behavioural information
  - Only indirect evidence

# Behavioural information: Behavioural models

- Components are in some physical condition
  - e.g. a wire




- Different physical conditions result in different behaviours

Behaviour 1

| v | 0 | + | - |
|---|---|---|---|
| i | 0 | + | - |

Behaviour 2

| v | 0 | + | - |
|---|---|---|---|
| i | 0 | 0 | 0 |

Behaviour 3

| v | 0 | + | + | - | - |
|---|---|---|---|---|---|
| i | 0 | 0 | + | 0 | - |

# Behavioural information: Ruling out behaviours

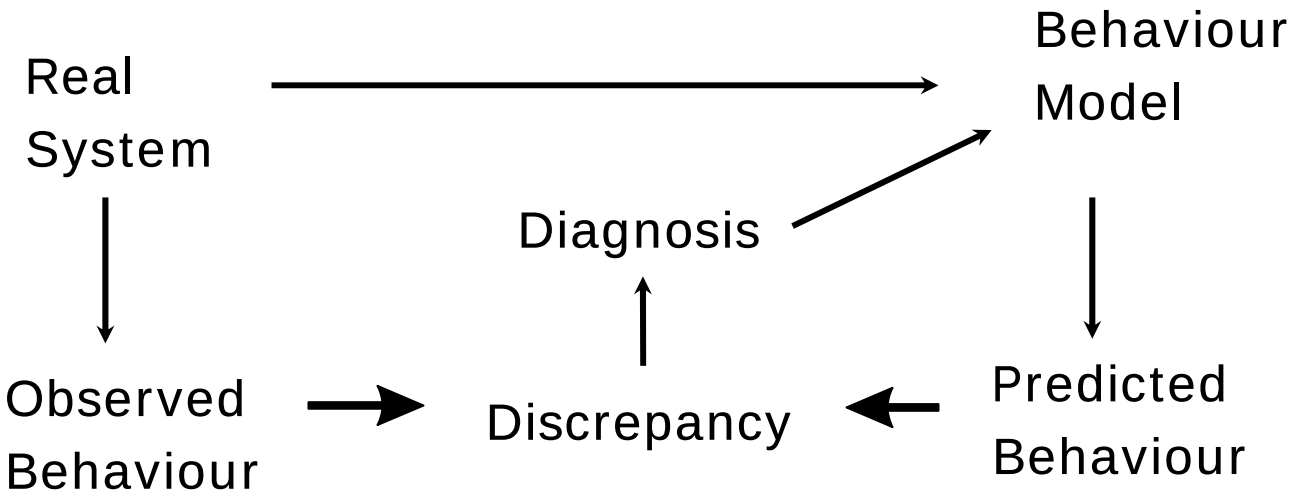
- We cannot verify the presence of behaviours, but we can falsify them
  - After observing

| v | 0 | + | - |
|---|---|---|---|
| i | 0 | 0 | 0 |

  - We cannot infer *behaviour 2*, but we can **reject *behaviour 1***

# Consistency Based Diagnosis Intuition

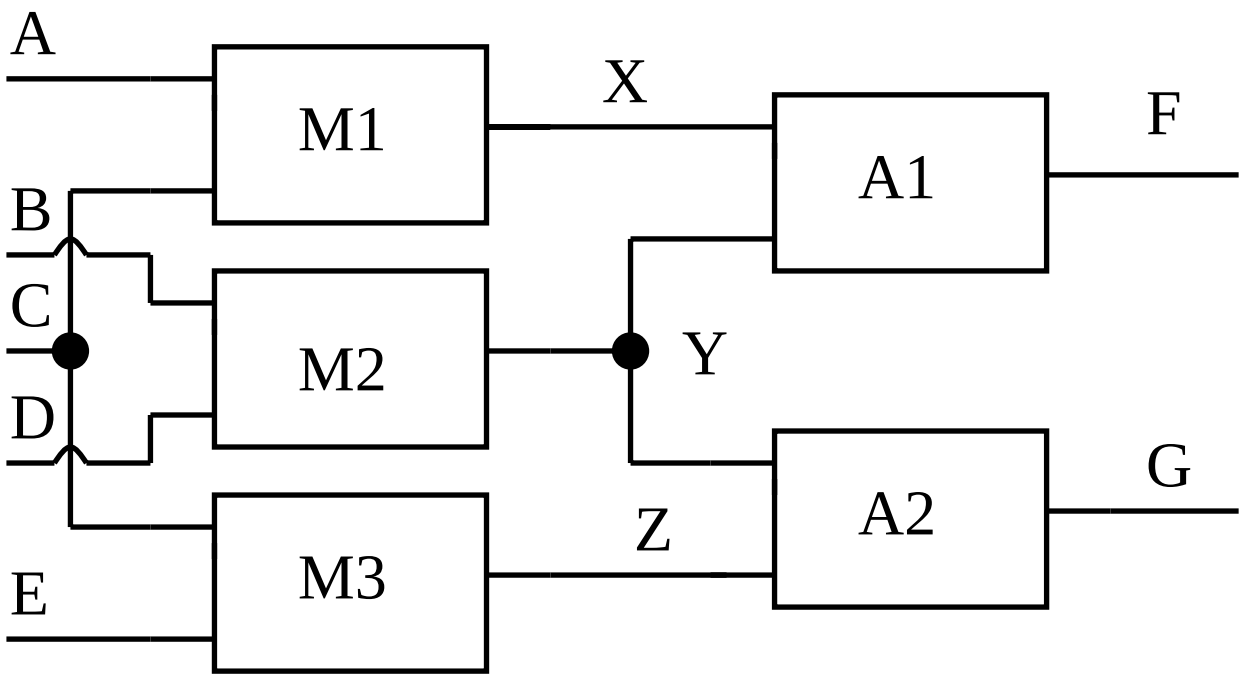
- Search for the model that is “compliant” with the observations



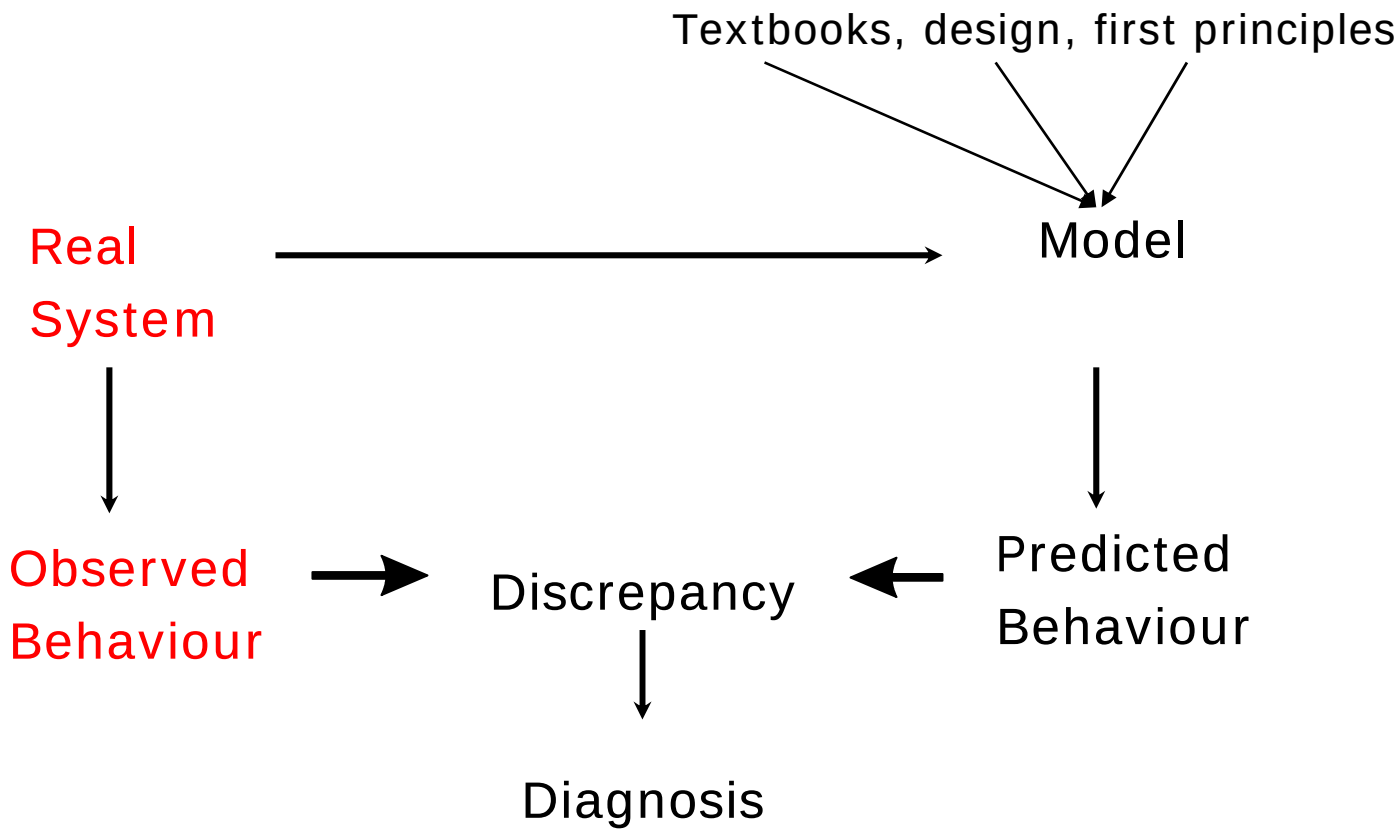
# General Diagnostic Engine

- GDE, de Kleer and Williams, 87
- First model based computational system for multiple faults
- Main computational paradigm
  - Still in use!
- Still a reference to compare any model base proposal on DX community

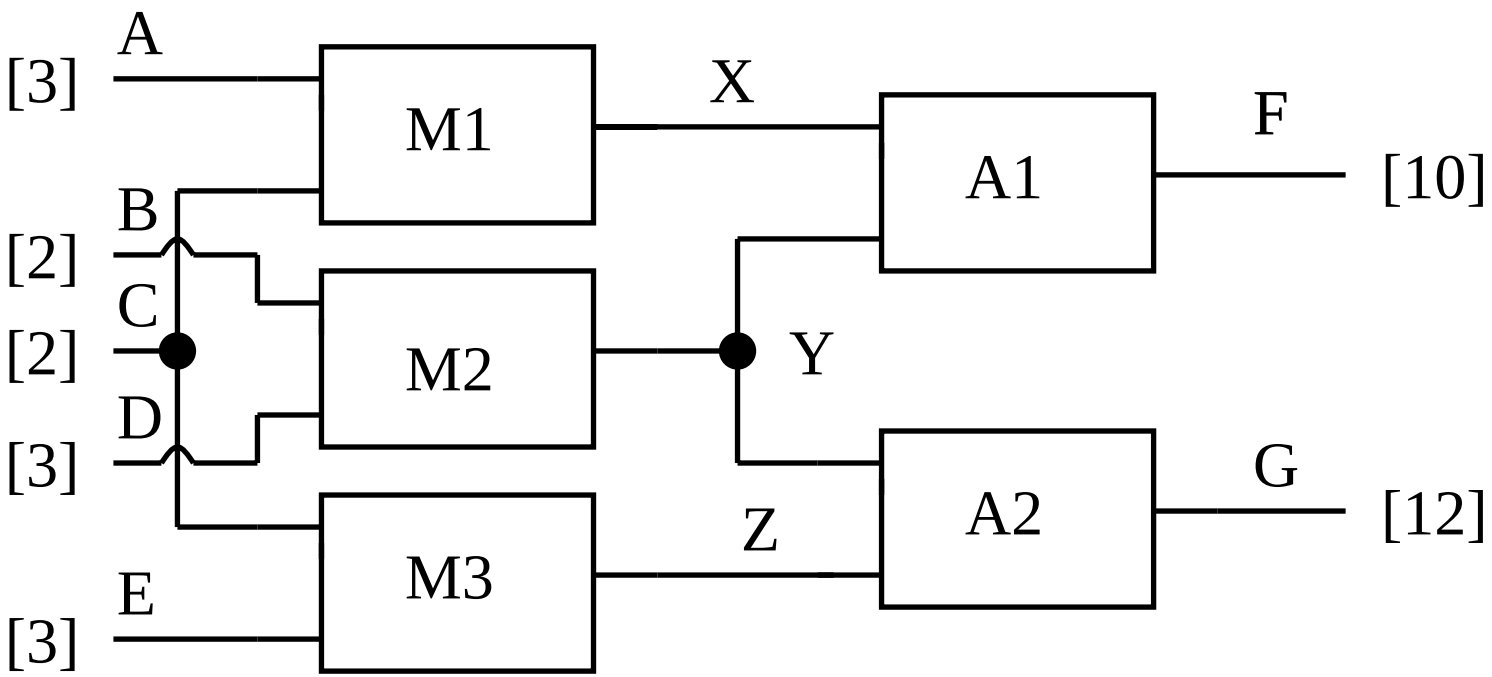
# A classic expository example: the polybox (de Kleer 87, 03)



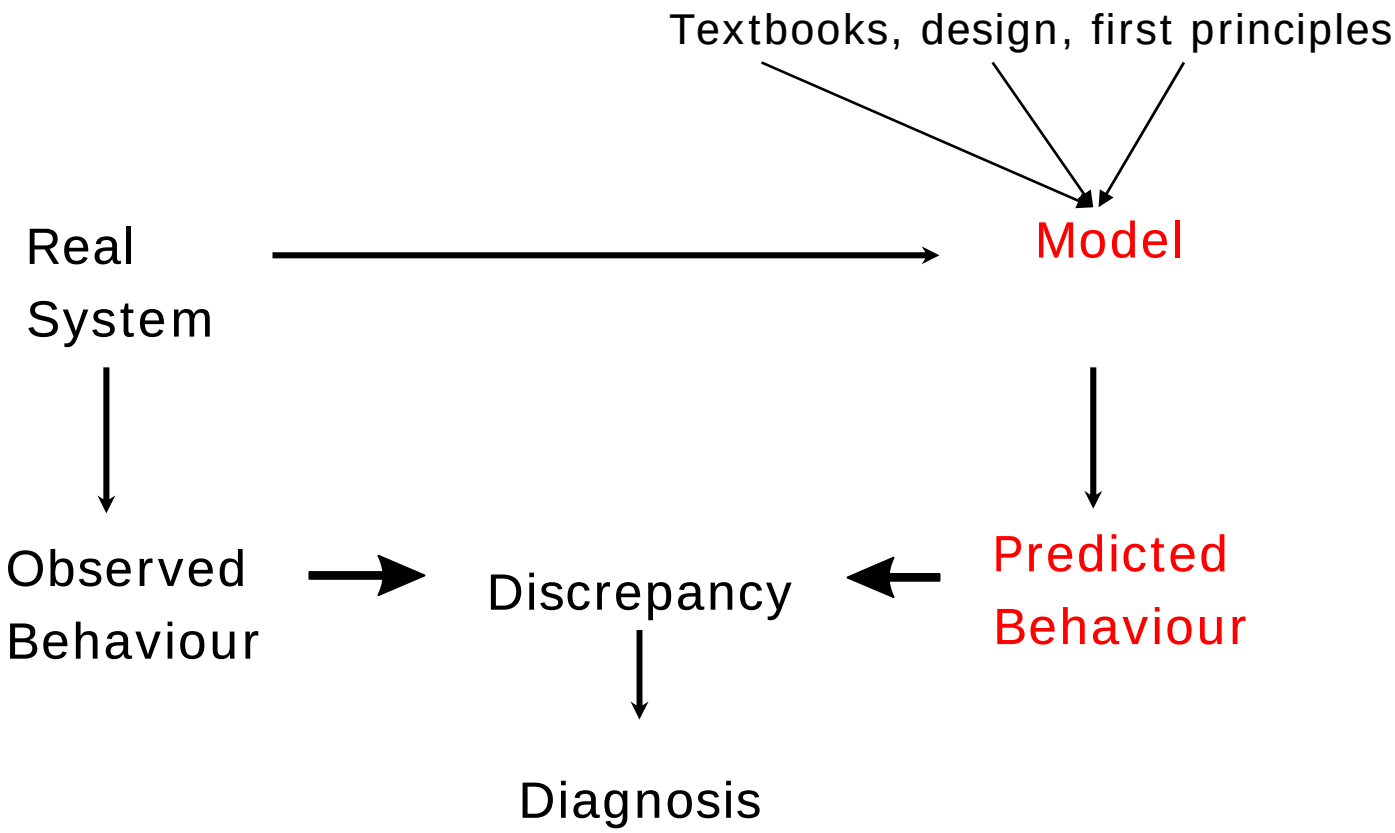
# Model based approach to diagnosis



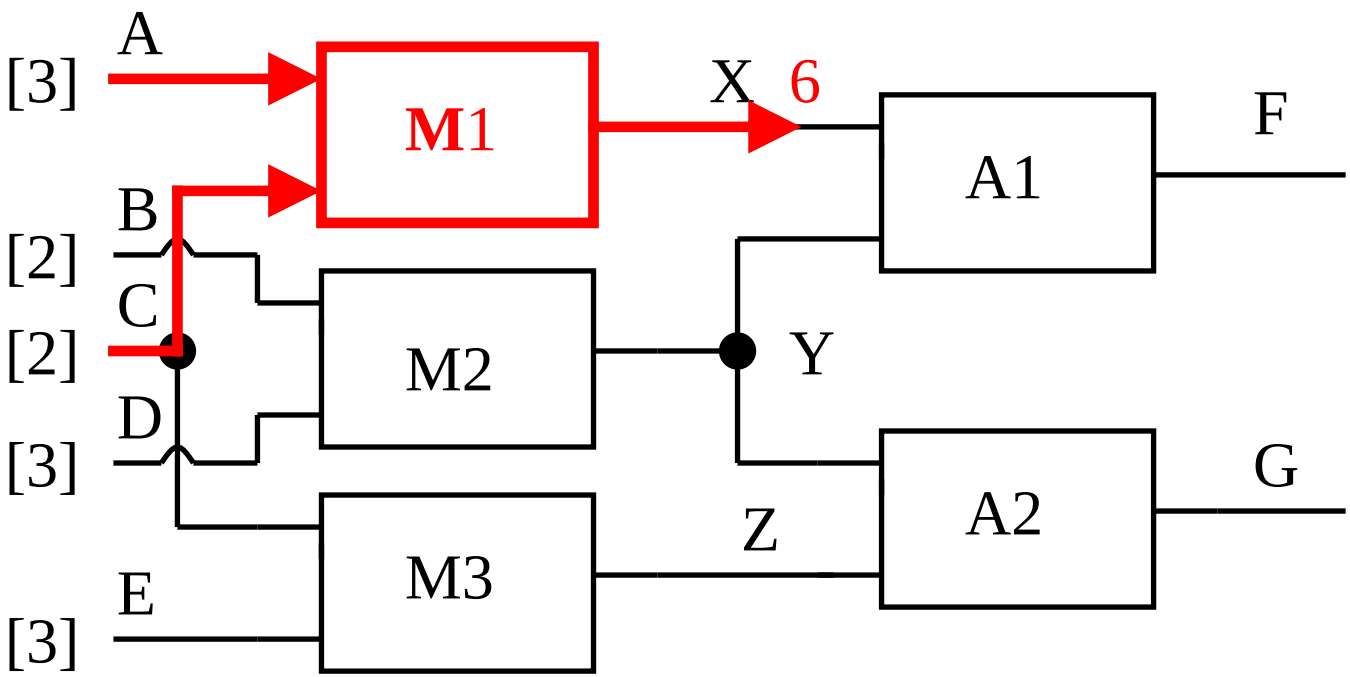
# Observed Behaviour



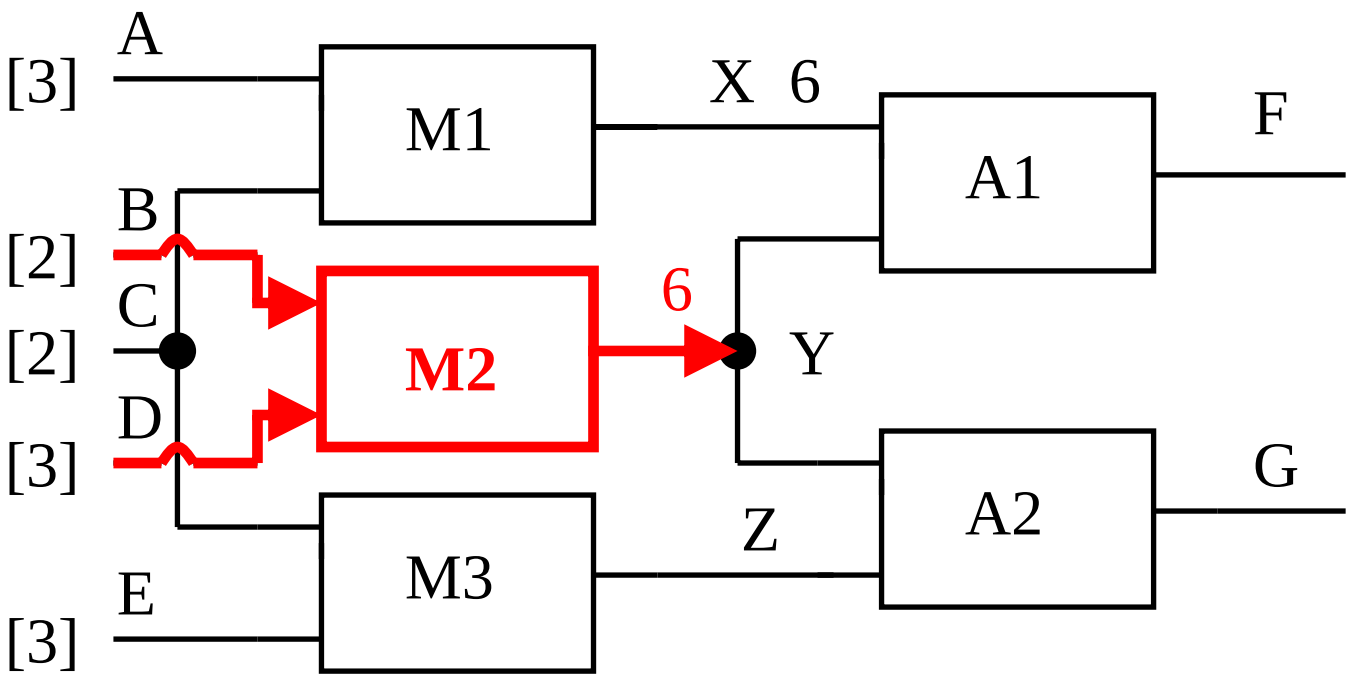
# Model based approach to diagnosis



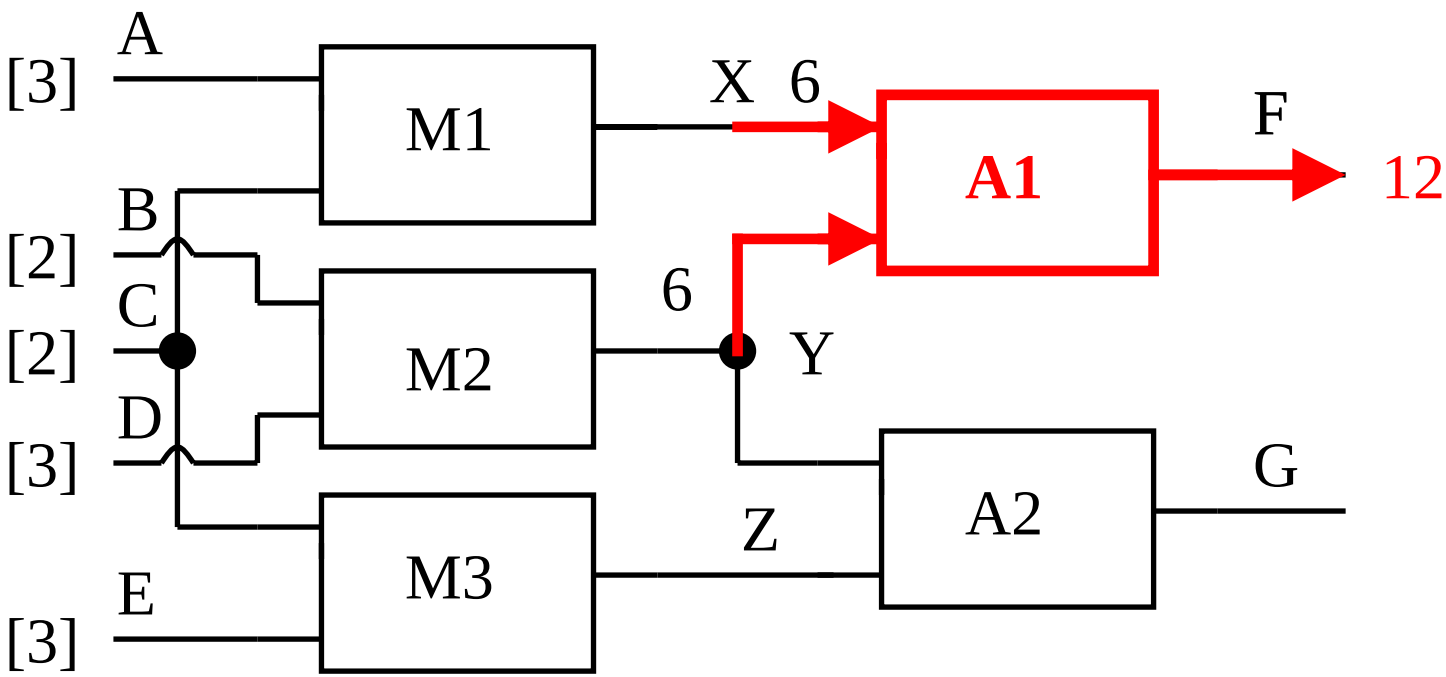
# Local propagation (I)



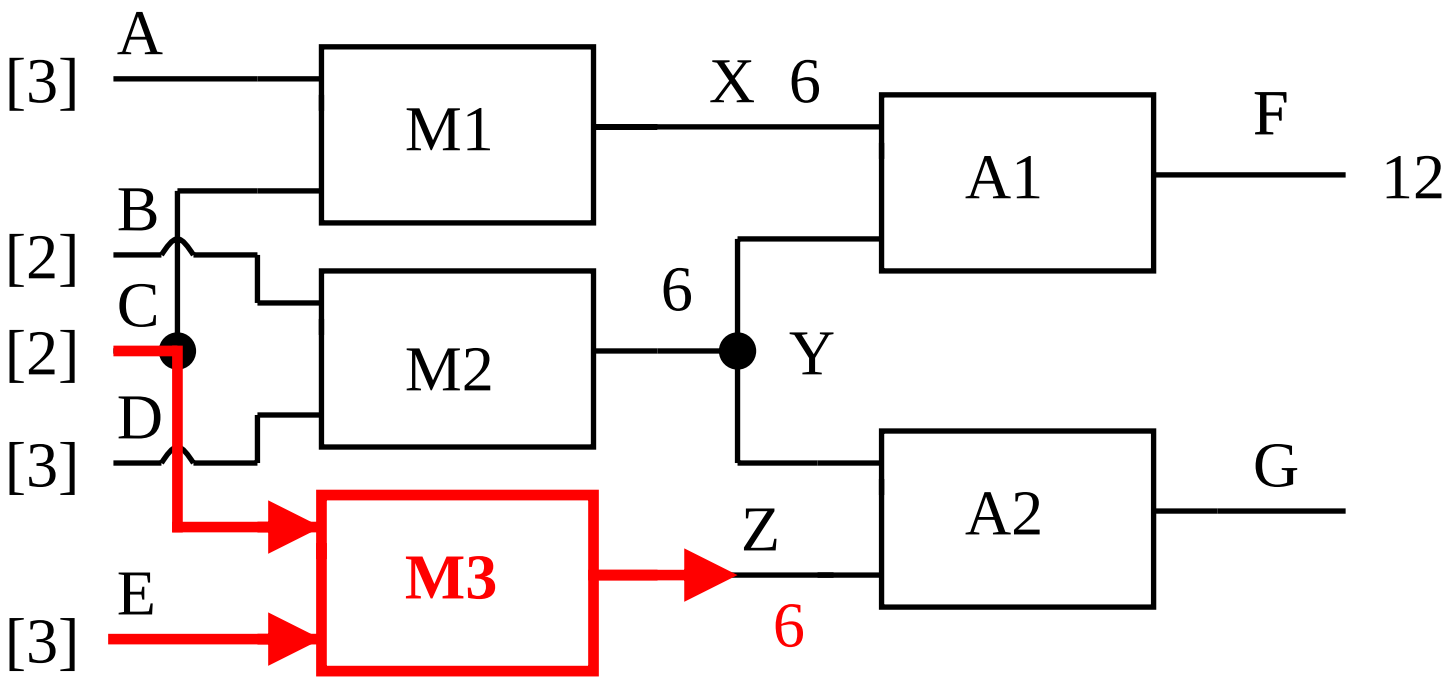
# Local propagation (II)



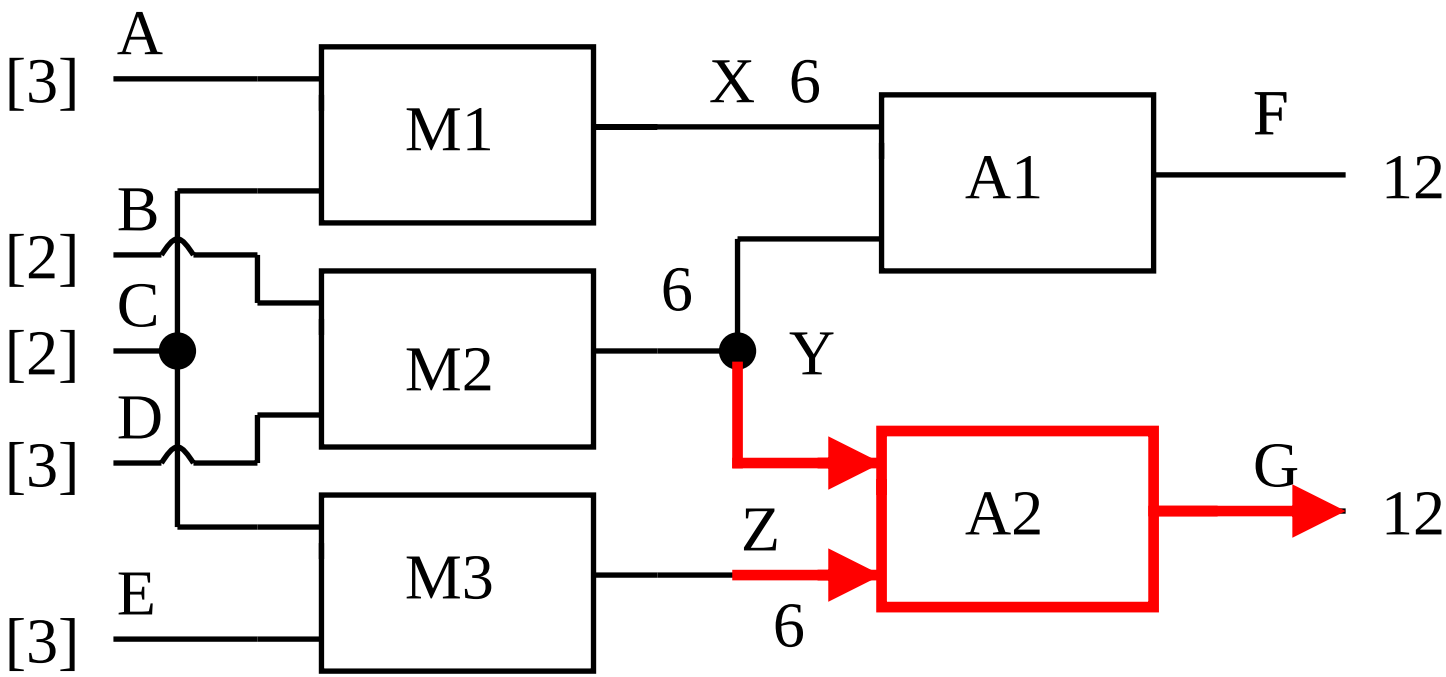
# Local propagation (III)



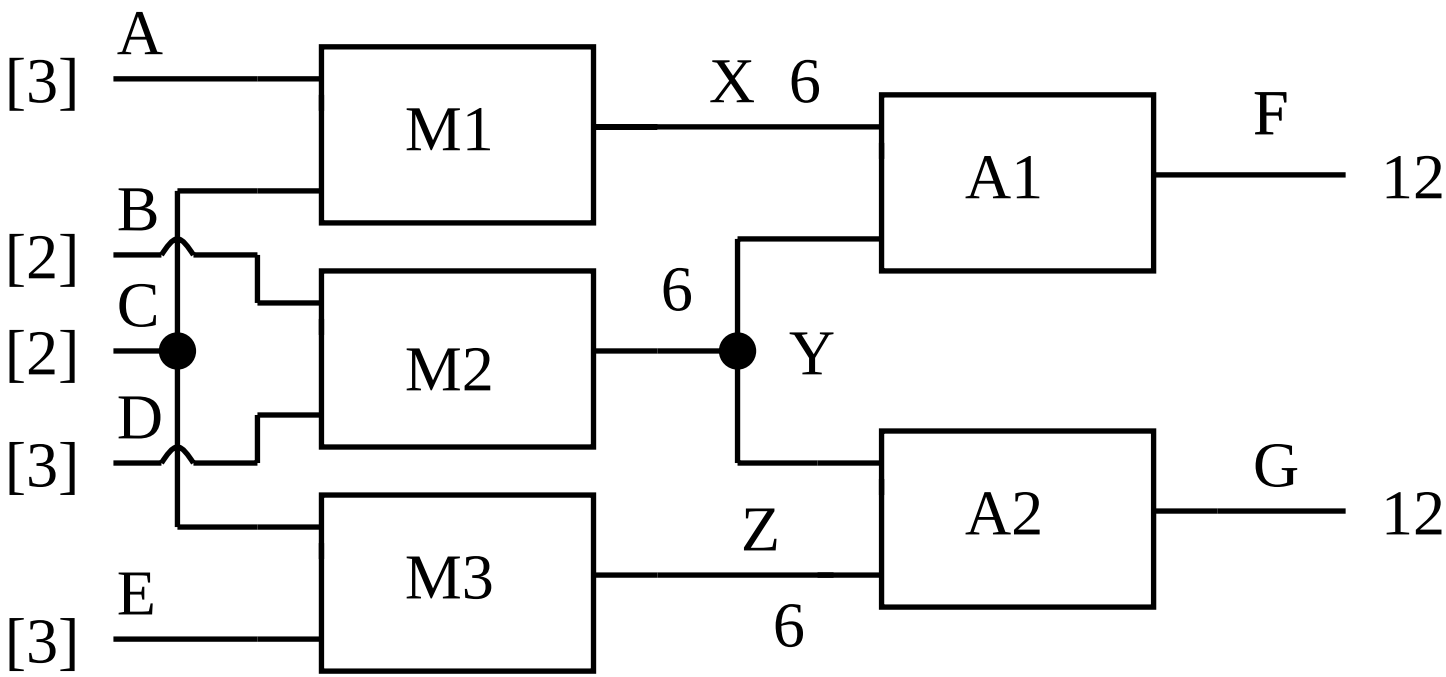
# Local propagation (IV)



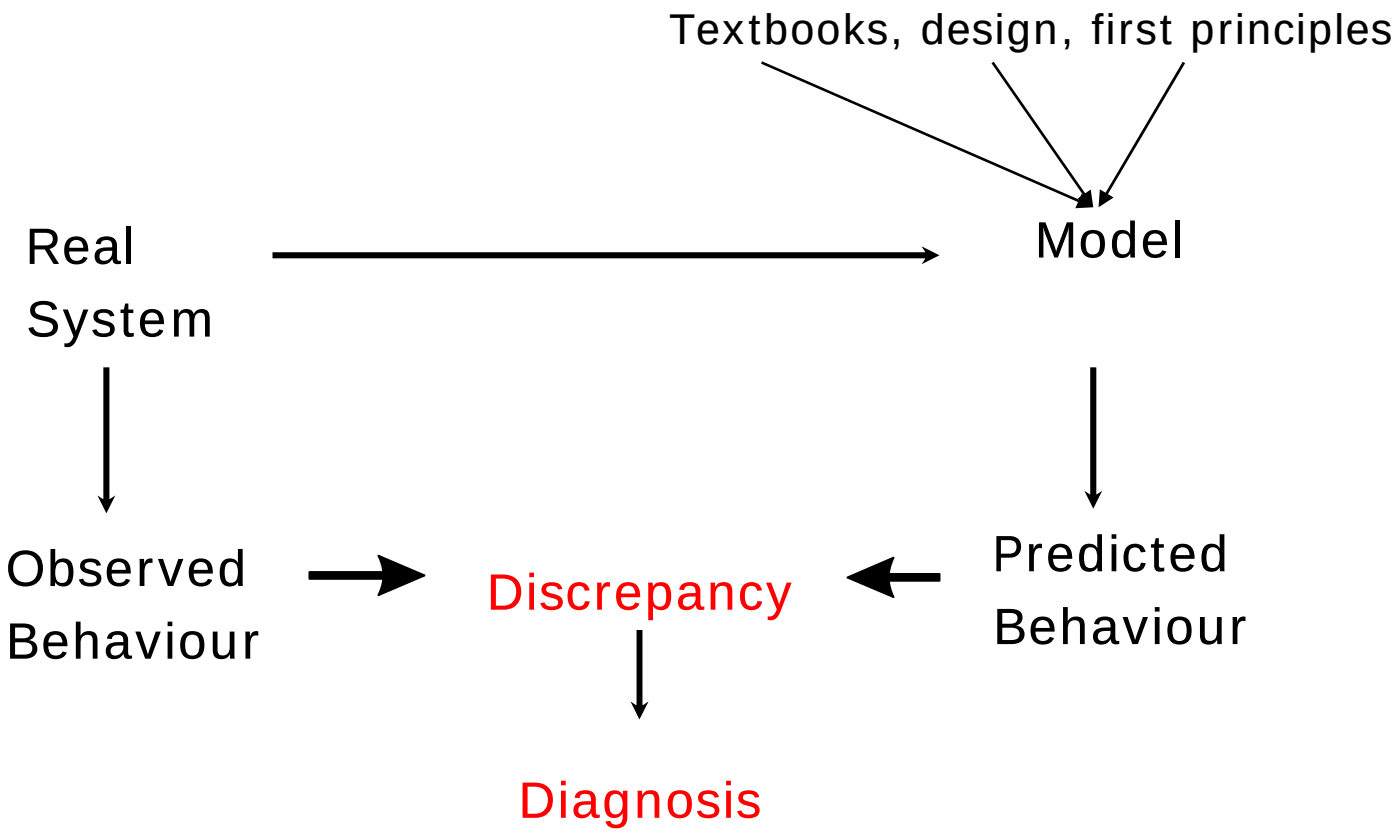
# Local propagation (V)



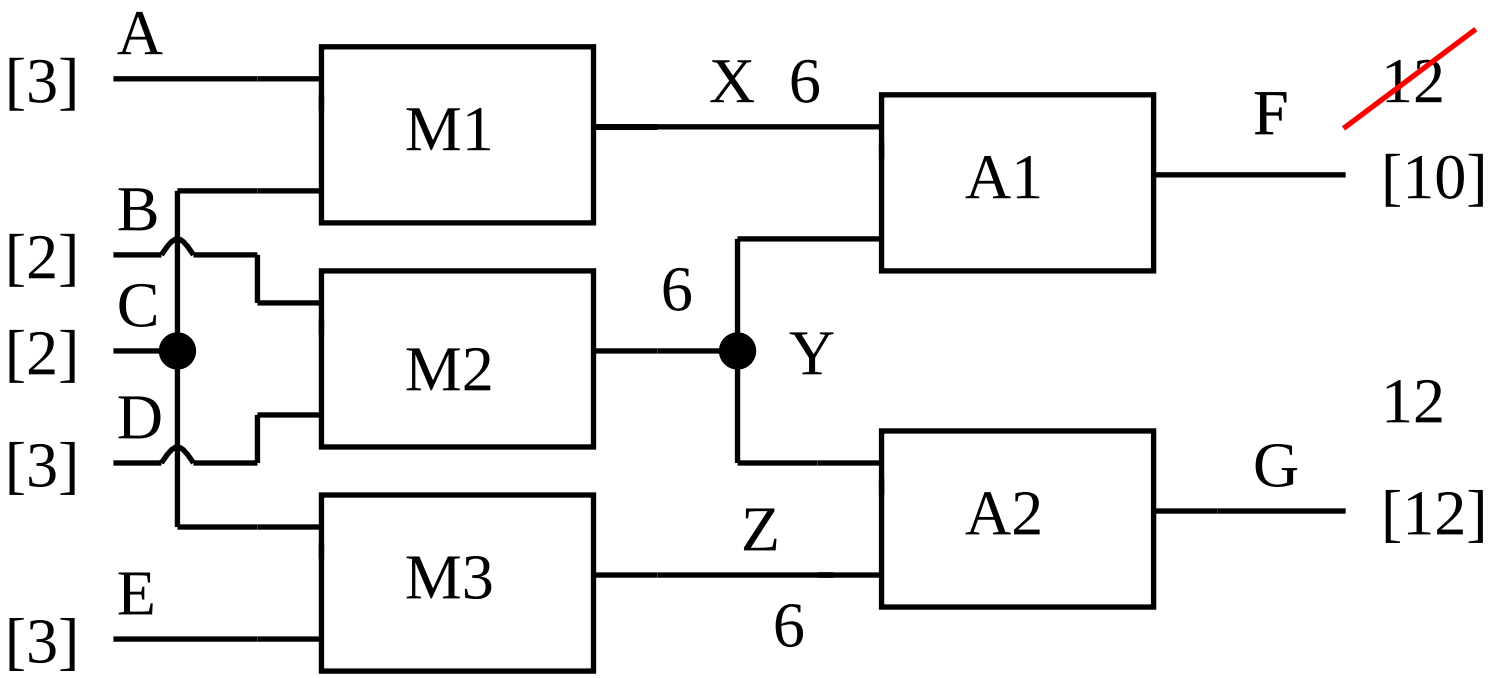
# Predicted Behaviour



# Model based approach to diagnosis

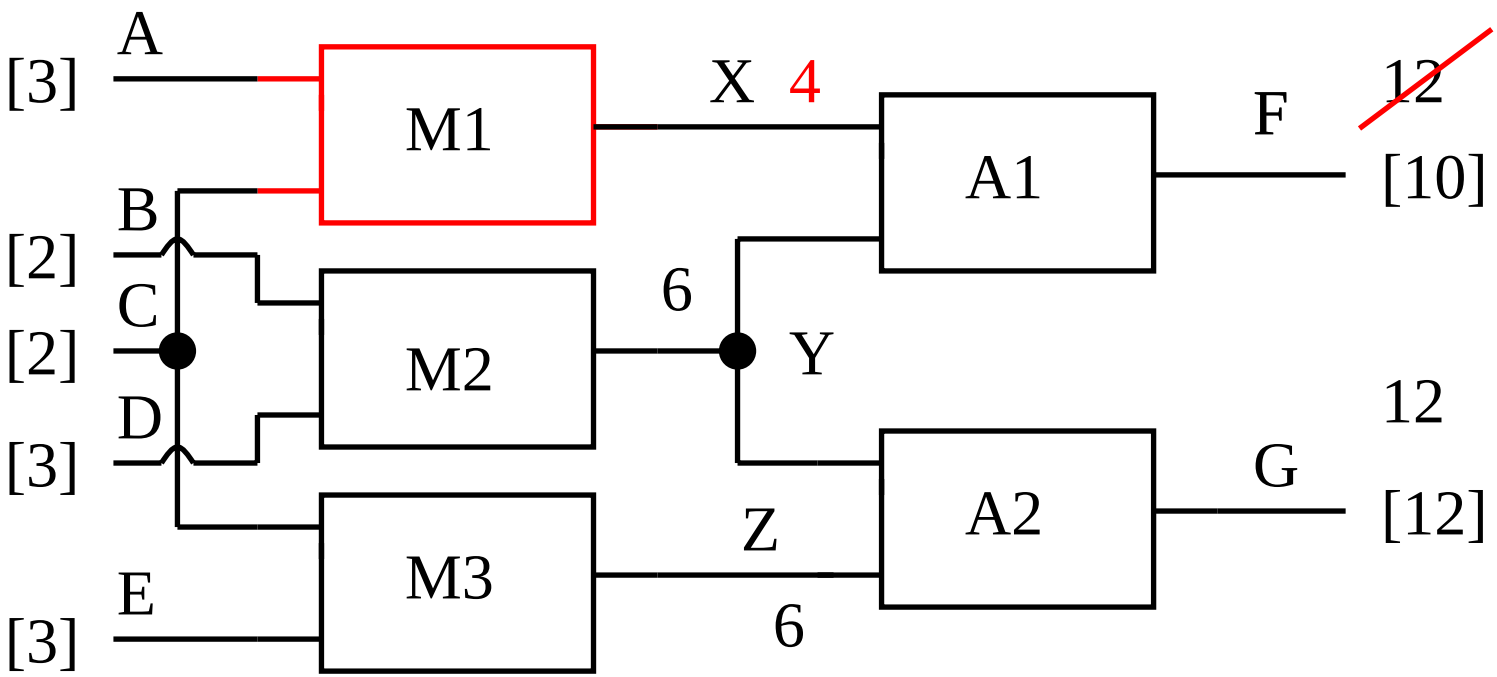


# Candidates

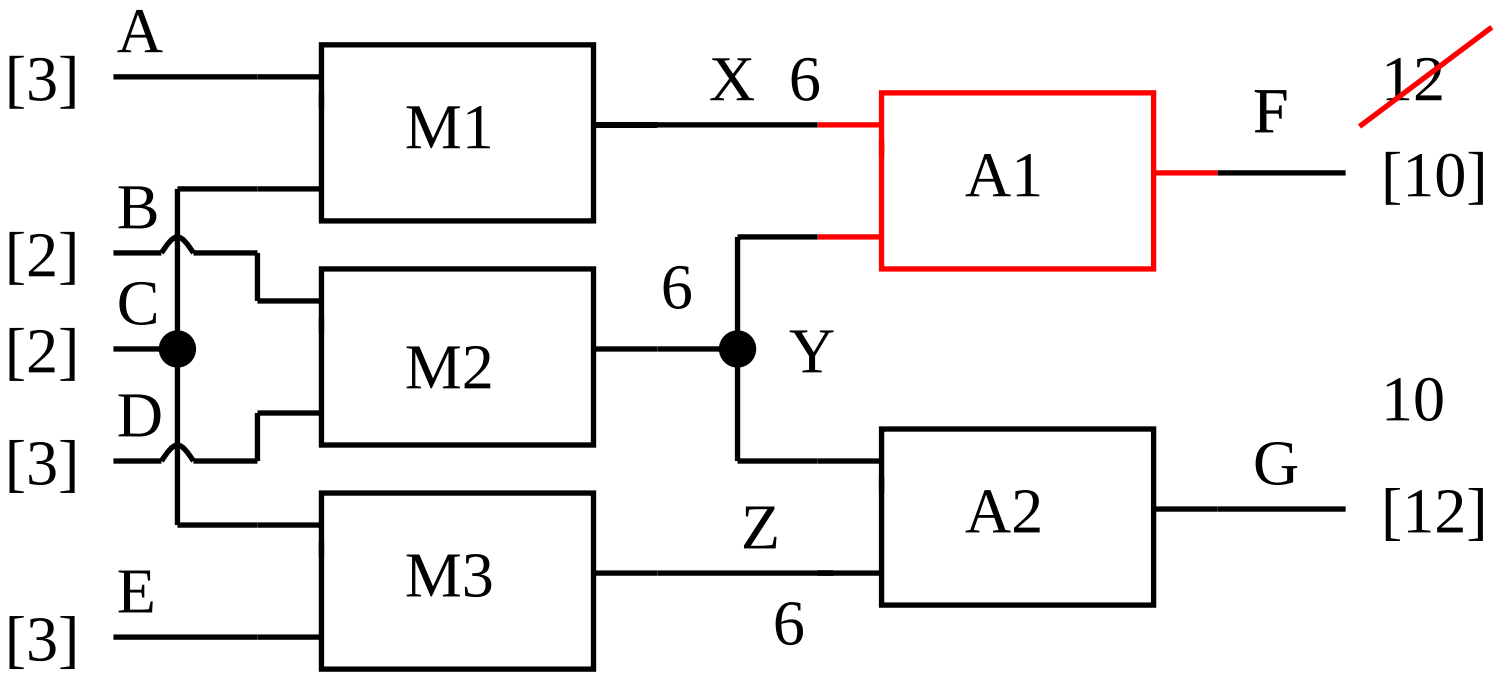



- Detect Symptoms: F=12 and F=10
- Generate Candidates: {M1}, {A1}, {M2, S2}, {M2, M3}

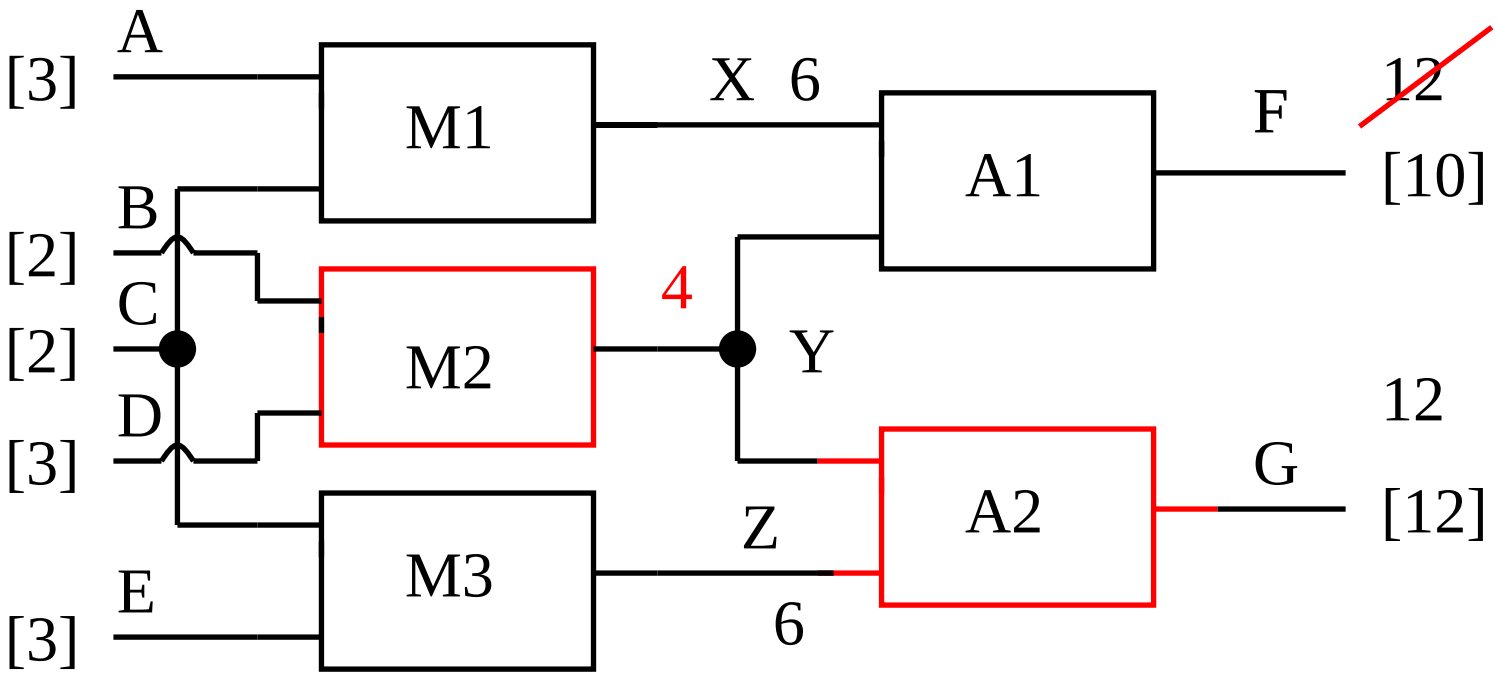
# Diagnosis for the polybox



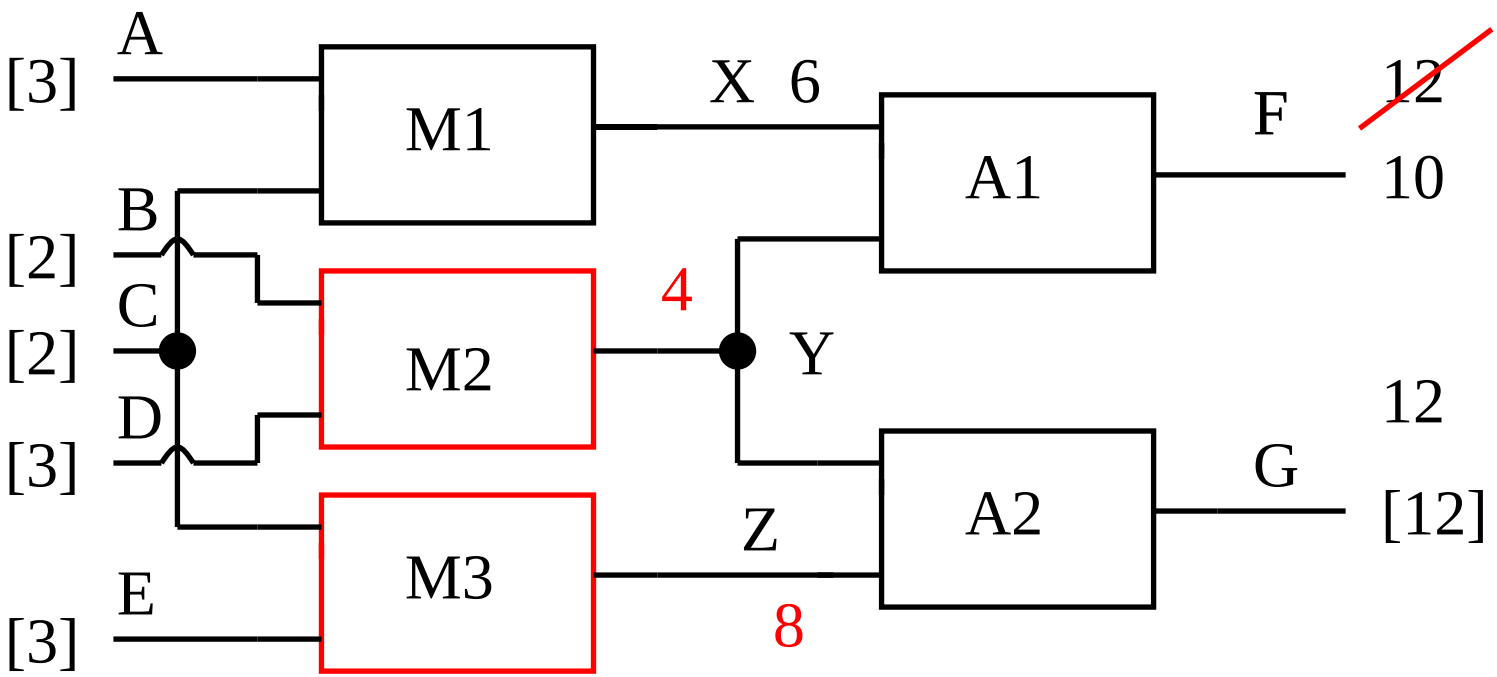
# Diagnosis for the polybox



# Diagnosis for the polybox



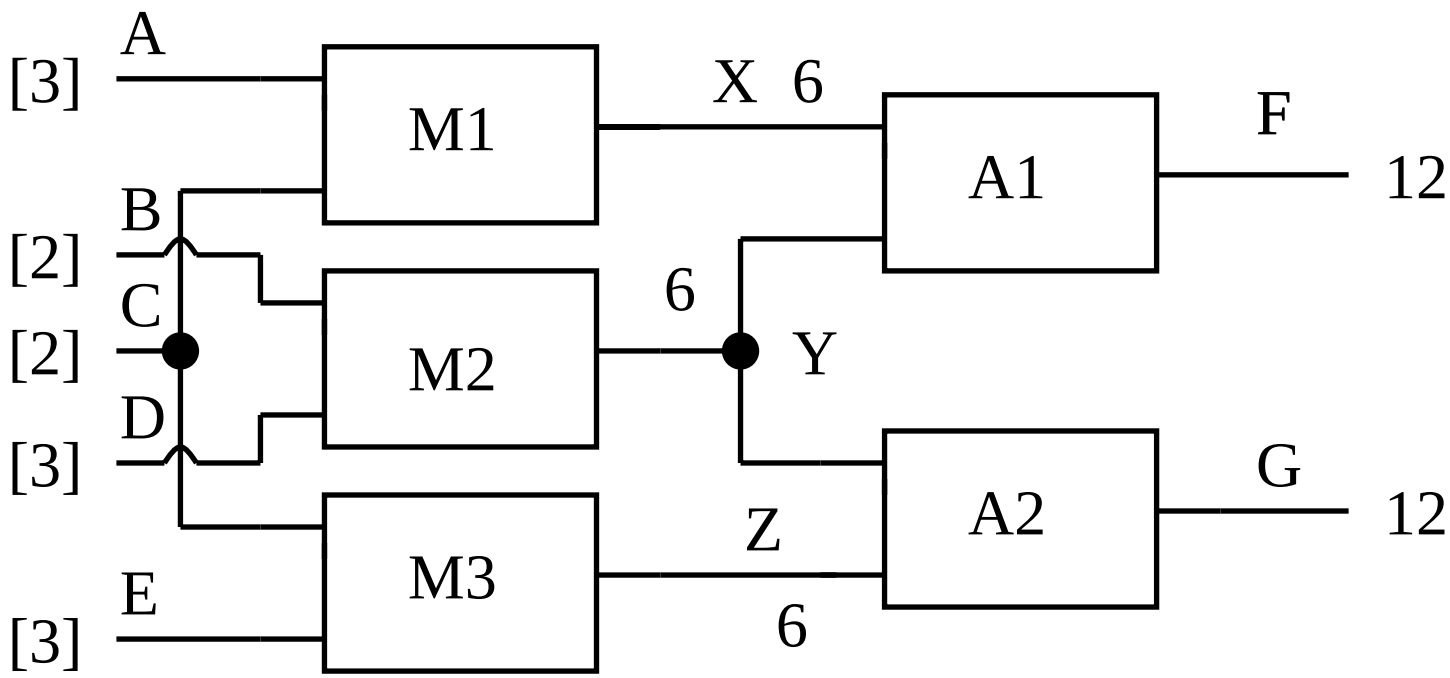
# Diagnosis for the polybox



# How GDE works?

1. Detecting every SYMPTOM
   - Prediction: propagating on every direction (even non causal!)
2. Identifying CONFLICTS
3. Generating CANDIDATES

# Prediction - Requirements




- Modelling structure
- Modelling component behaviour
- Predict overall behaviour

# Modelling Structure - Requirements

- Determine the structural elements and interconnections
  - Which entities can be the origin of malfunction?
  - Which parts can be replaced?
  - Which variables can be observed?
  - Reflect aspects and levels of (diagnostic) reasoning about the device behaviour

# Component-Oriented Modelling: Components and Connections

- Systems: **components** linked by **connections** via **terminals**
  - Components:  Normally physical objects
    - Resistors, diodes, voltage sources, tanks, valves
  - Terminals: unique communication link
  - Connections:ideal connections (but may be modelled as components)
    - No resistance wires, load less pipes…
- Possible faults: defect components, broken connection

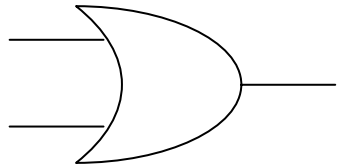
# Modelling Behaviour - Requirements

- Describe behaviour of the structural elements: Locality
- Goal: detecting discrepancies
- Consider aspects like
  - Generality: which kind of devices are to be diagnosed?
  - Robustness: which type of failure are to be detected
- Reflect the diagnostic reasoning process (e.g. simplifications)
- Which kind of information is (easily) available (e. g. qualitative information)

# Local behaviour models

- Constrains / relations among
  - Input/Output variables
  - Internal parameters
- Various directions
- No implicit reference to or implicit assumptions about context (existence or state of other components)
- **Locality**
  - Necessary for diagnosis: different context because something is broken; otherwise implicit hypothesis must be revised
  - Reusability: model library, compositionally

# Local behaviour model - Example



- Or-gate
  - Variables: in1, in2,out
  - Domain

  $dom(in1)=dom(in2)=dom(out)=\{0,1\}$
  - Relation

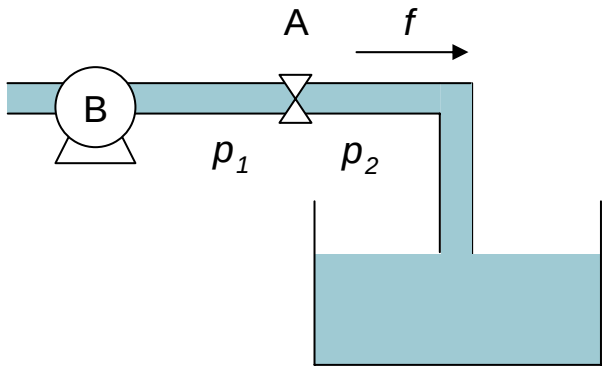
  $\{\{0,0,0\}, \{1,0,1\}, \{0,1,1\}, \{1,1,1\}\}$
  $\subset dom(in1) \times dom(in2) \times dom(out)$
- Inferences
  - $in1 = 1 \supset out = 1$
  - $in2 = 1 \supset out = 1$
  - $in1=0 \wedge in2=0 \supset out = 1$

causal direction

  - $out=0 \supset in1=0 \wedge in2=0$
  - $out=1 \wedge in1=0 \supset in2=1$
  - $out=1 \wedge in2=0 \supset in1=1$

non causal direction

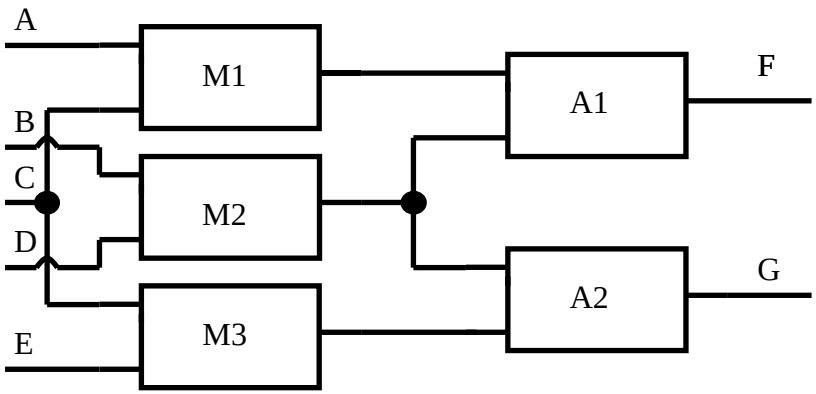
# Behaviour model of a valve




- Relation: f = k × A
  - Implicit assumption: pump is on and ok
- Relation: IF on(B) and ok(B) THEN f= k × A
  - Implicit assumption: a pump exists and is connected as in the diagram
- Better: f ∝ k' × (p1 – p2) × A

- Principle: **no function in structure**

# Abstract model

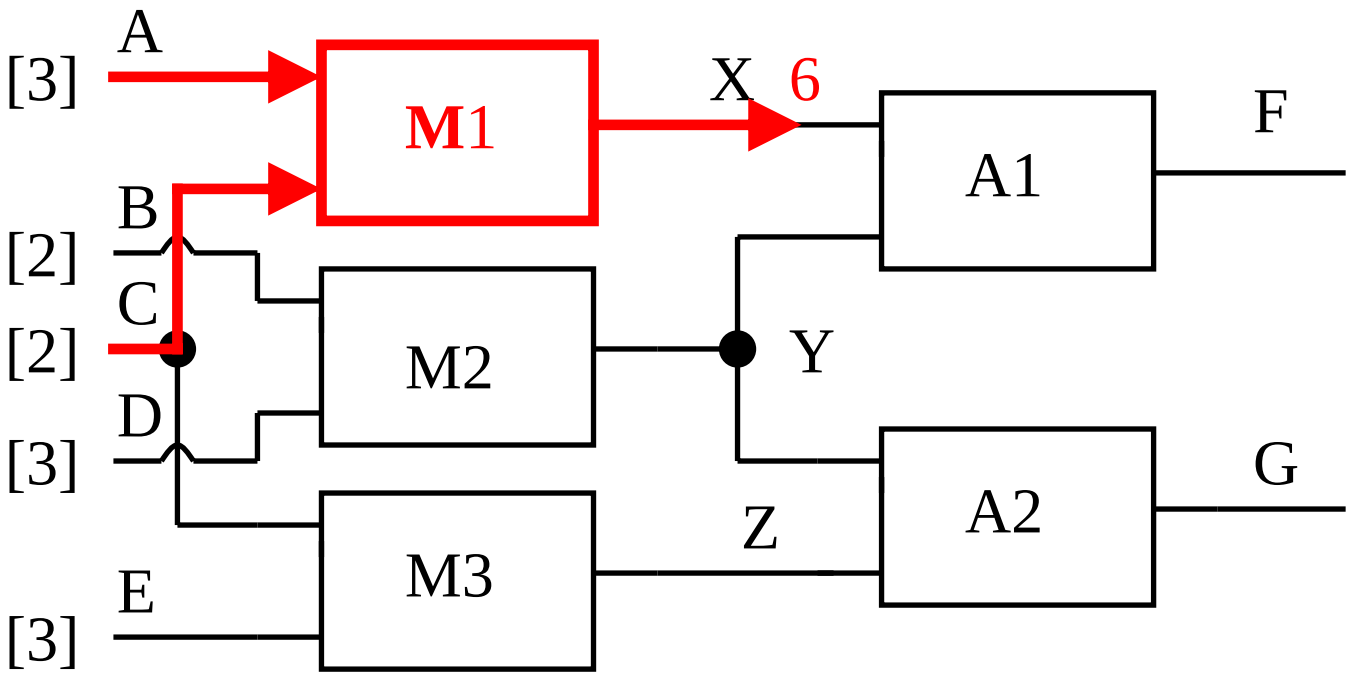



- Domain for each variable, *var*
  dom(*var*) = {*OK*, *BAD*, *?*}
- Model for each correct component, C
  IF for all input-variables, $var_i$ of C, $var_i$ = *OK*
  THEN for each output-variable, $var_o$ of C, $var_o$ = *OK*
- To avoid masking of faults by correct components
  IF there exists an input-variable, $var_i$ of C, $var_i$ = *BAD*
  THEN for each output-variable, $var_o$ of C, $var_o$ = *BAD*

# Prediction - Principles

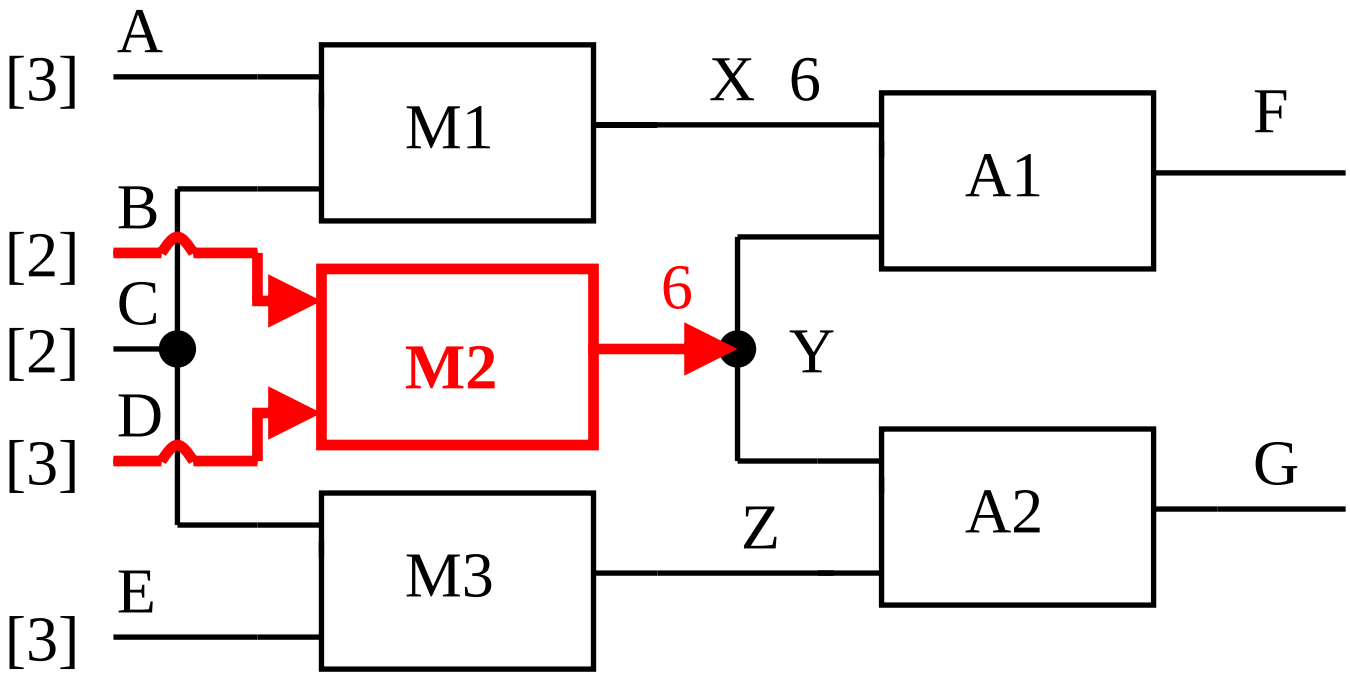
- Infer the behaviour of the entire device from
  - Observations
  - Component models
  - Structural description
- Preserve dependencies on component models
- Propagate the effects of local models along the interaction paths (connections)
- Propagate not only in the causal direction

# Propagation
# Causal direction (I)



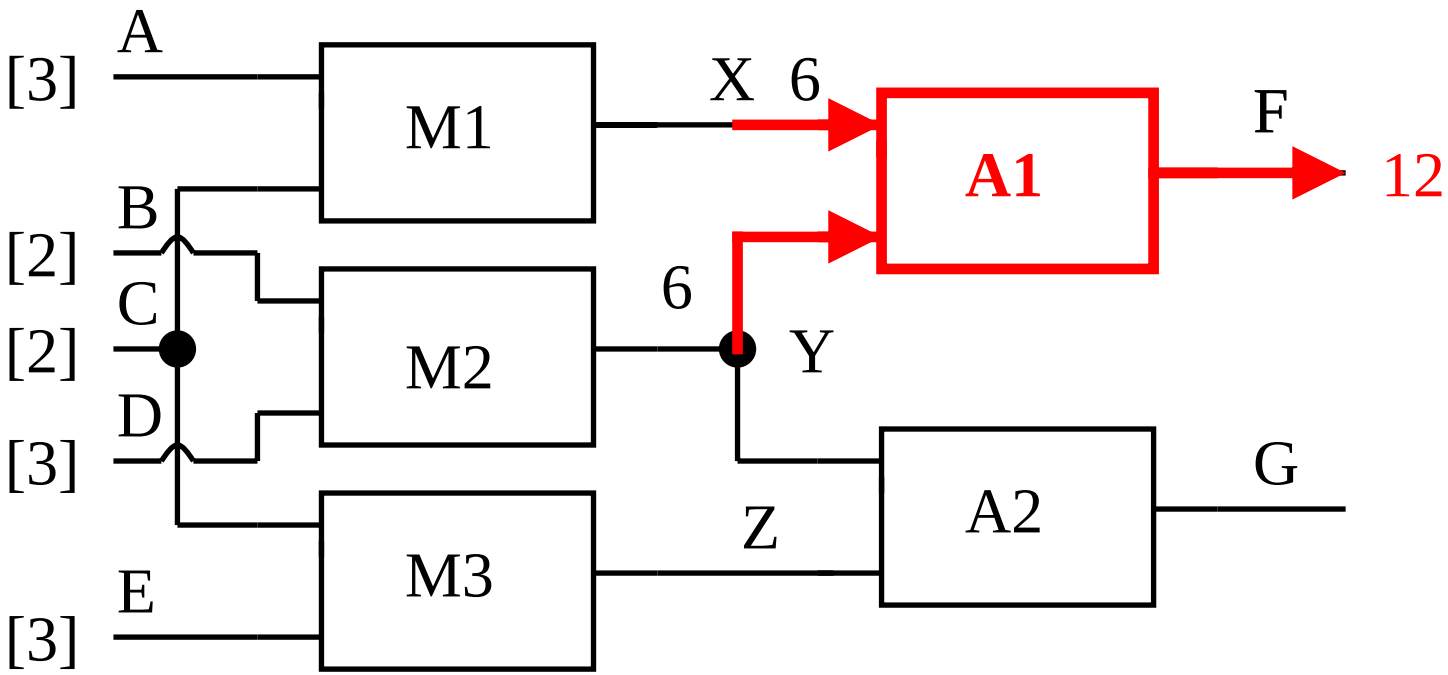

- $[A]=3 \wedge [C]=2 \supset X=6$ (M1)

# Propagation
# Causal direction (II)



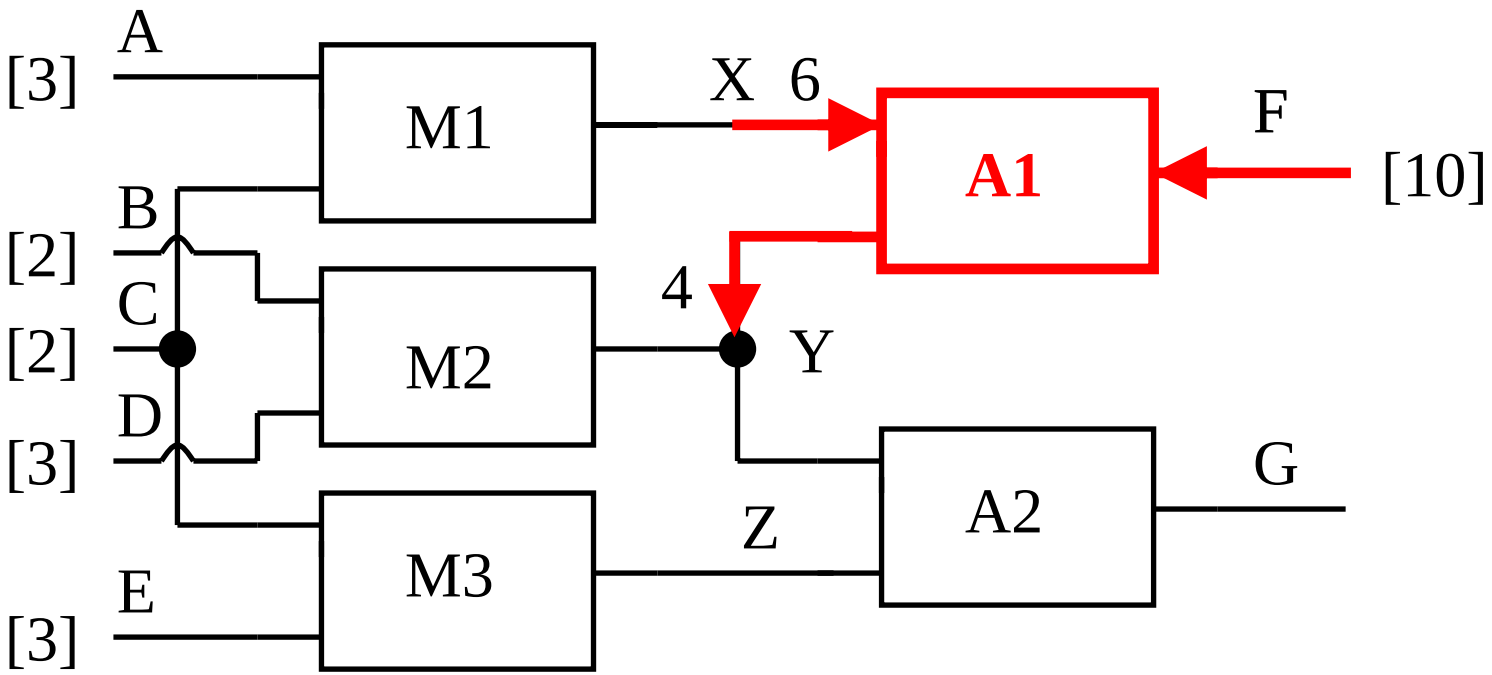

- $[B]=2 \wedge [D]=3 \supset Y=6$ (M2)

# Propagation
# Causal direction (III)




- $X=6 \wedge Y=6 \supset F=12$ (A1)

# Propagation
# “Backward” direction (II)

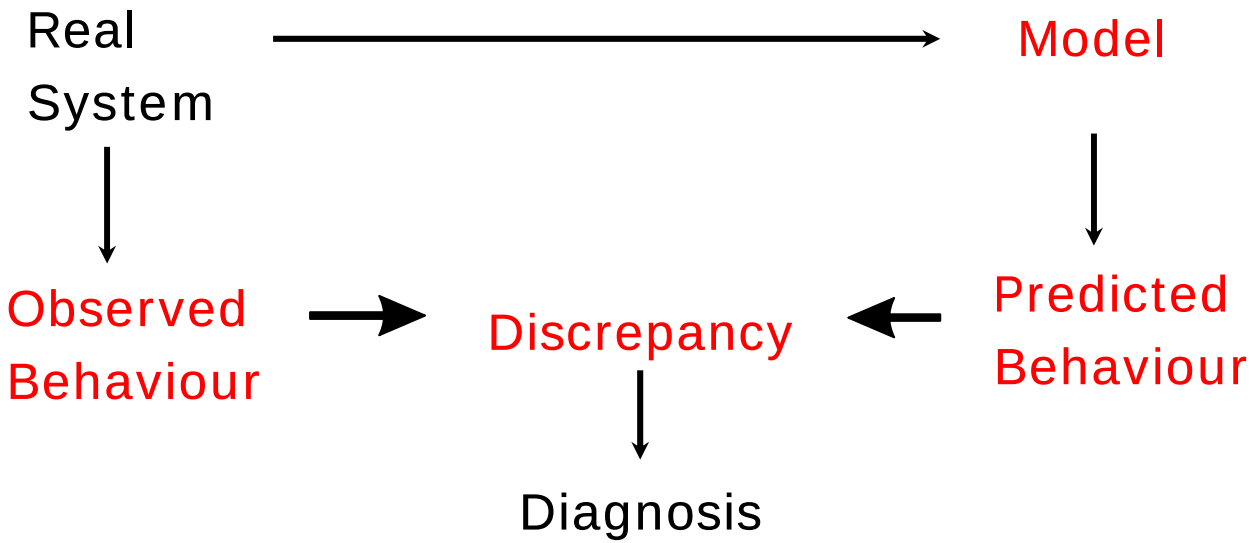



- $[F] = 10 \wedge X = 6 \supset Y = 4$ (A1)

# Candidate Generation

- Detecting SYMPTOMS (DISCREPANCIES)
- Identifying (minimal) CONFLICTS
- Generating (minimal) CANDIDATES

# Symptoms

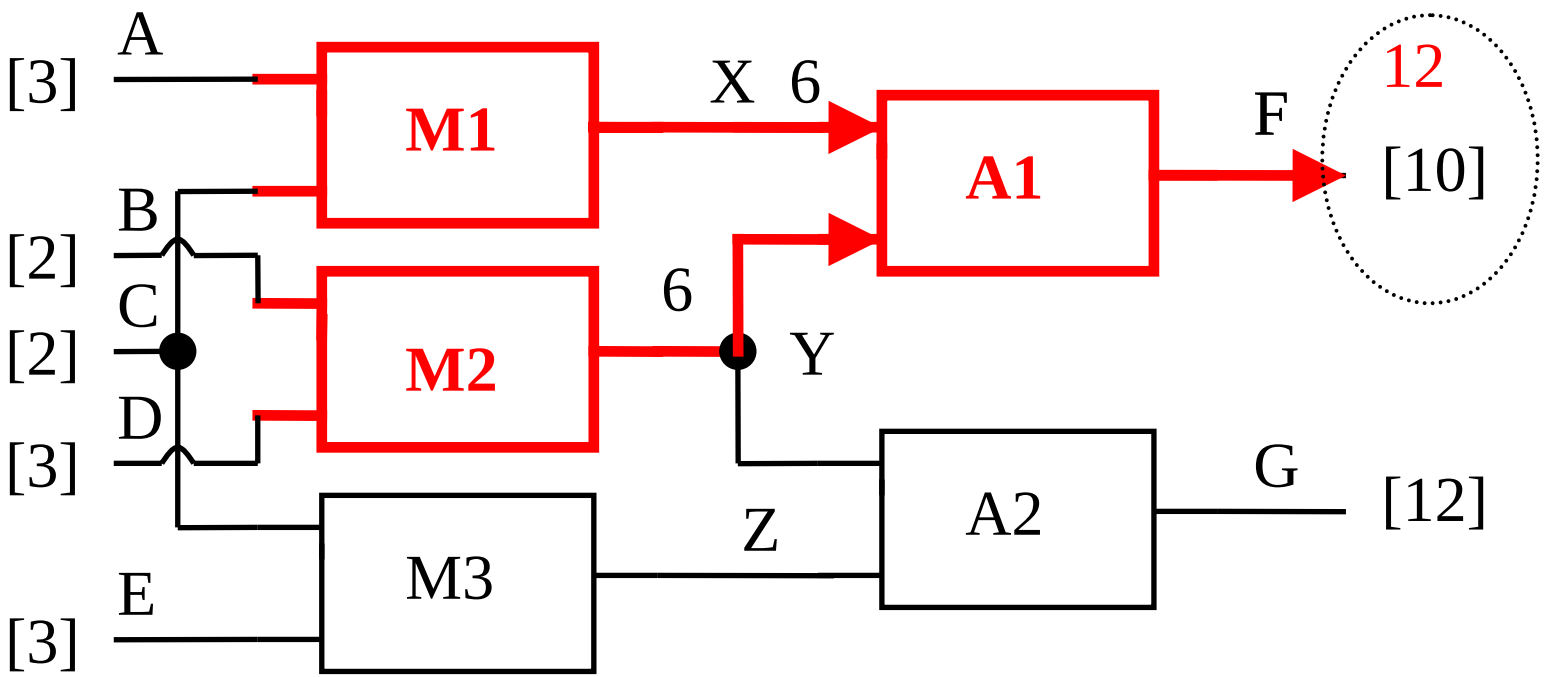



- Symptoms are contradictions that indicate an inconsistency between observations and correct behaviour
  - But other potential sources of contradictions
    - Imprecise measurements
    - Bugs in the model
    - Bugs in propagation

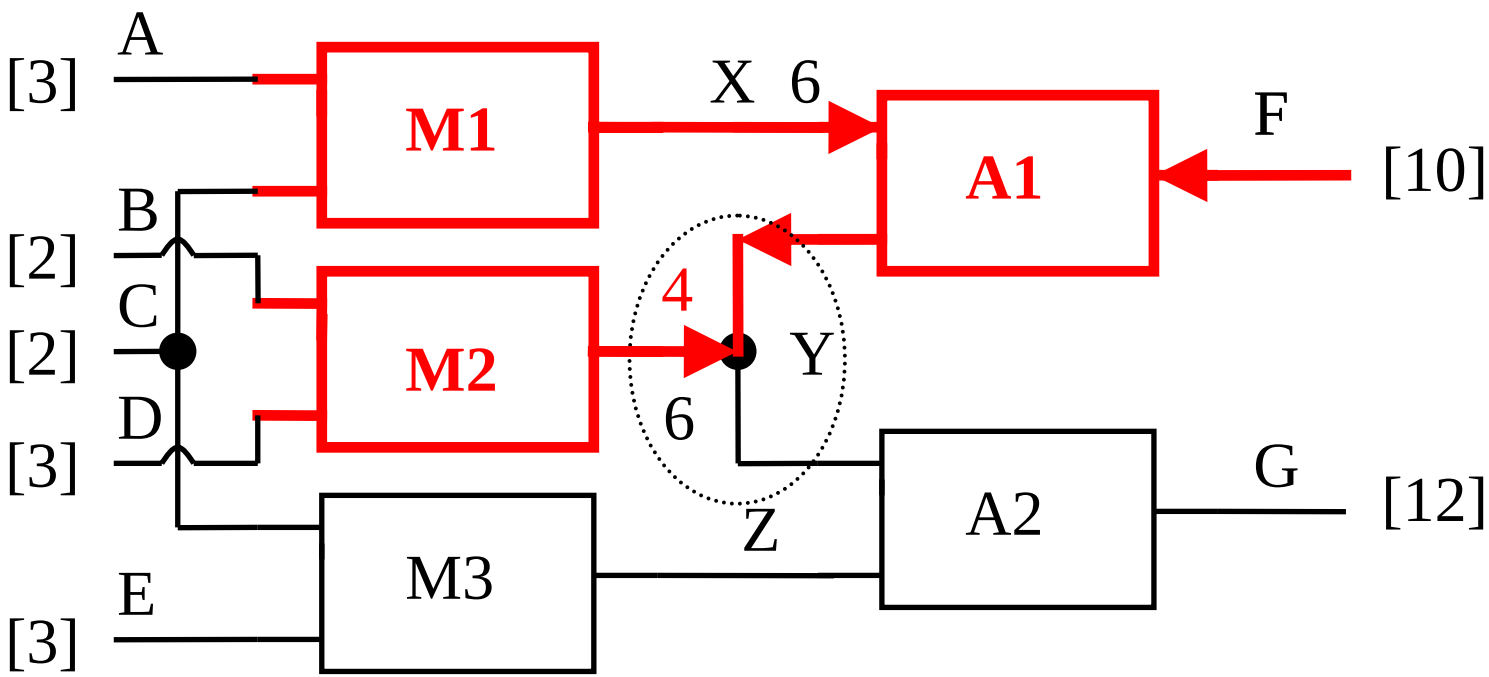
# Symptoms Detection

- Symptoms occurs as contradictory values for one variable
  - Predicted plus observed
  - Predicted following two different paths
- Discrete Variables
  - Static $x=val1 \wedge x=val2 \wedge val1 \neq val2$
  - Dynamic $x=(val1, t1) \wedge x=(val2, t2) \wedge val1 \neq val2 \wedge (t1 \cap t2) \neq \varnothing$
- Continuous Variables
  - Qualitatives (static):
    - Intervals: $x=i1 \wedge x=i2 \wedge (i1 \cap i2) \neq \varnothing$
    - Values: $x=val1 \wedge x=val2 \wedge val1 \neq val2$
    - Relations: $x \approx val1 \wedge x \langle < val2$
  - Quantitatives : distance, $\int$distance >Threshold

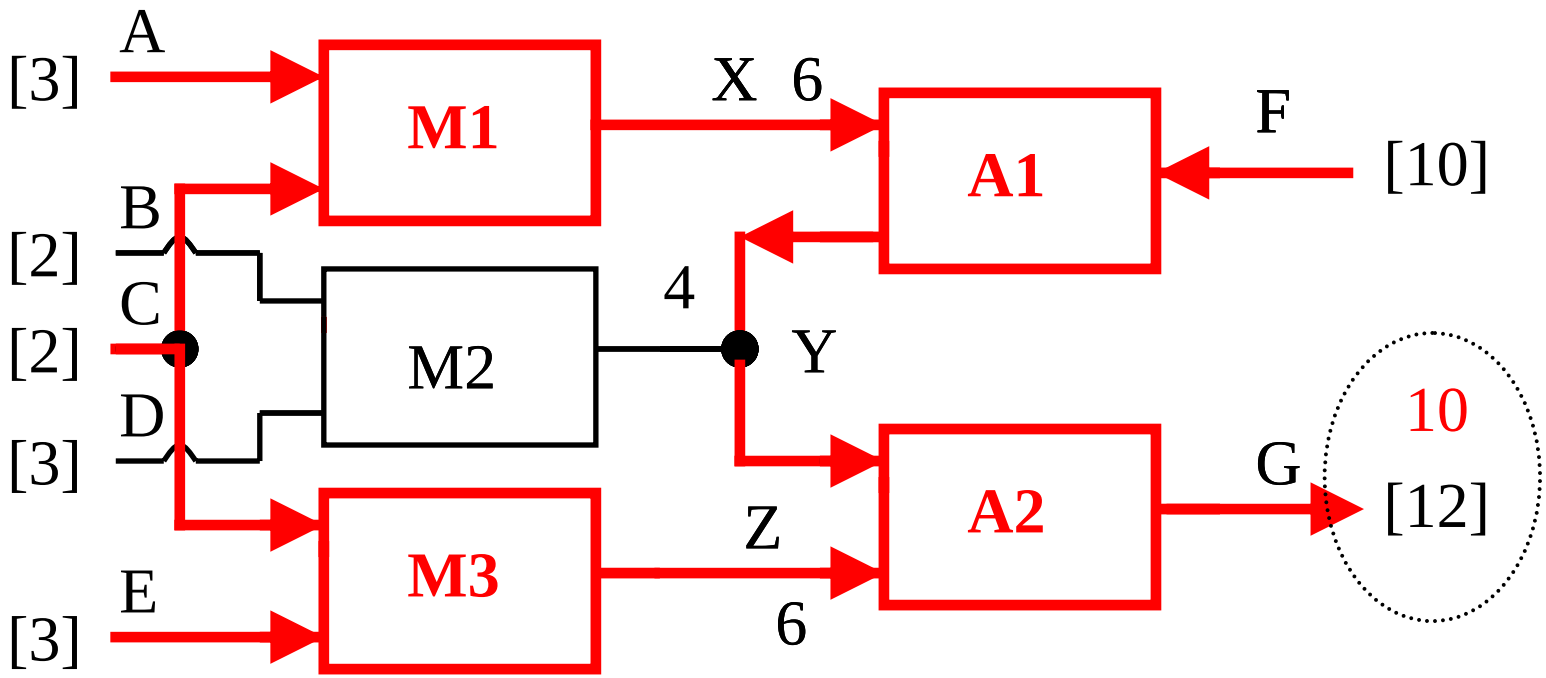
# Some symptoms for the polybox (I)



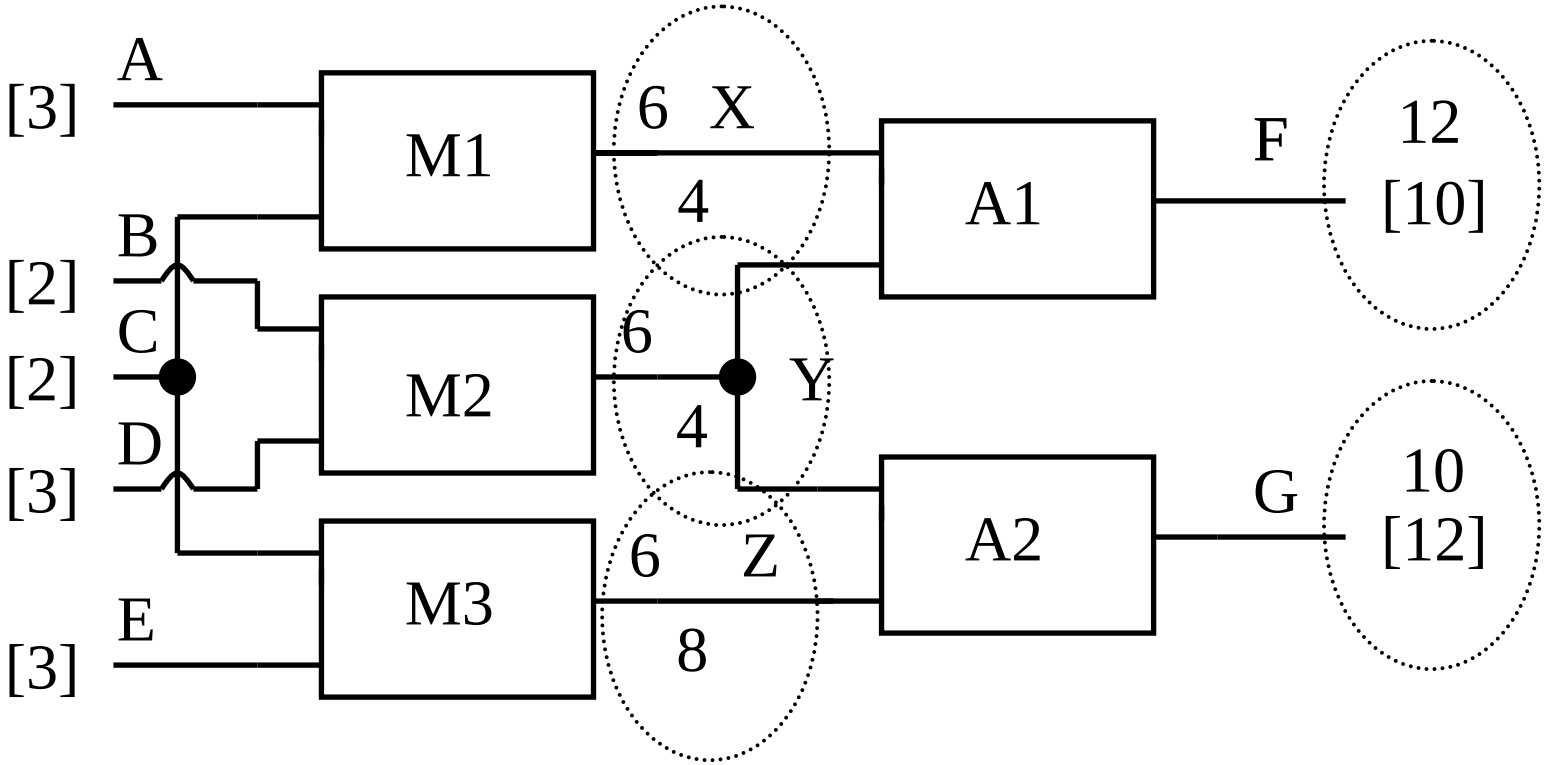
# Some symptoms for the polybox (II)



# Some symptoms for the polybox (III)



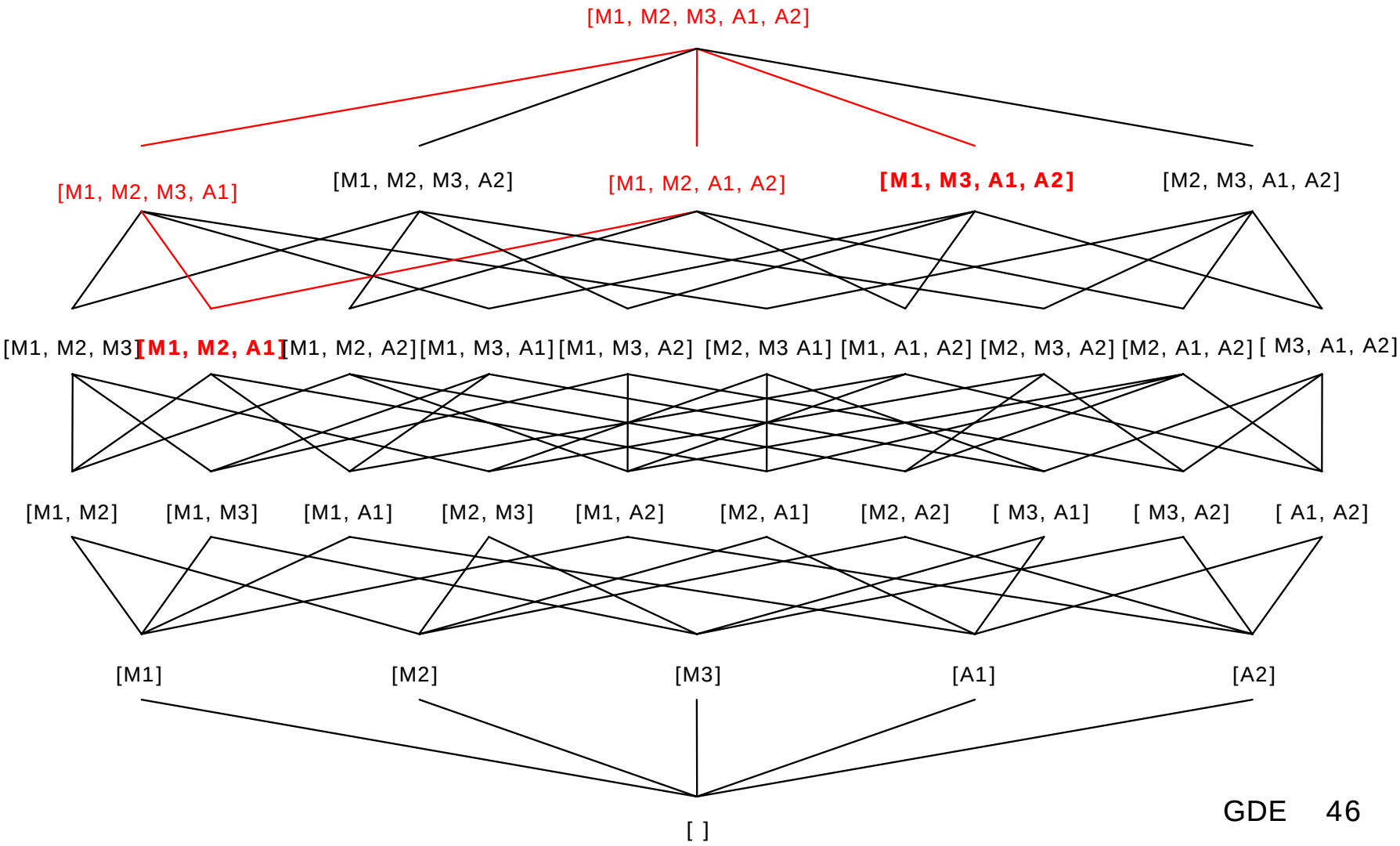
# Some symptoms for the polybox (IV)



# Identify conflicts

- Conflict (informal):set of correctness assumptions underlying discrepancies
- Polybox (minimal) conflicts
  - $F=[10] \wedge F=12$ {M1, M2, A1}, {M1, M3, A1, A2}
  - $X=6 \wedge X=4$ {M1, M2, A1}, {M1, M3, A1, A2}
  - $Y=6 \wedge Y=4$ {M1, M2, A1}, {M1, M3, A1, A2}
  - $Z=6 \wedge Z=8$ {M1, M3, A1, A2}
  - $G=[12] \wedge G=10$ {M1, M3, A1, A2}
- By definition,any superset of a conflict set is a conflict
  - $\{M1, M2, A1\} \subset \{M1, M2, A1, A2\} \subset \{M1,M2, M3, A1, A2\}$
- Minimal conflict: conflict no proper subset of which is a conflict
- It is essential to represent the conflicts through the set of minimal conflicts (to avoid combinatorial explosion)

# Conflicts lattice

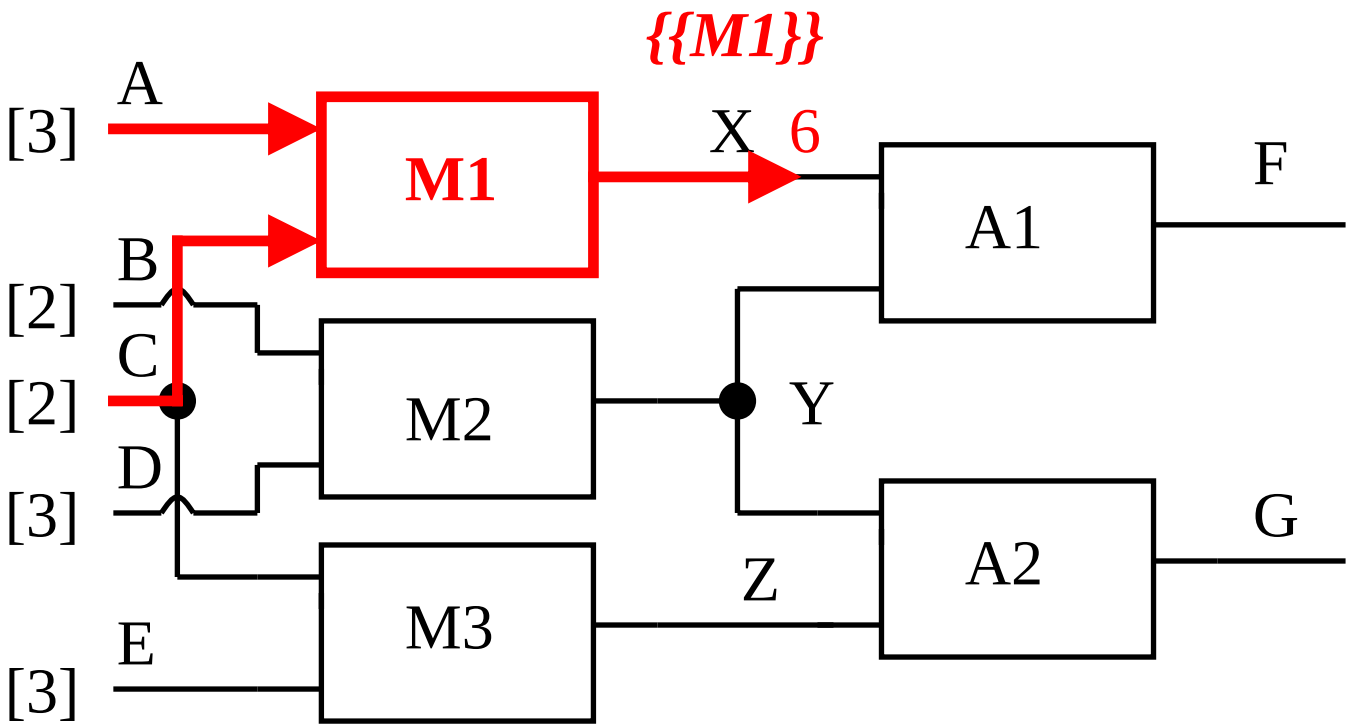


# Conflicts generation with ATMS

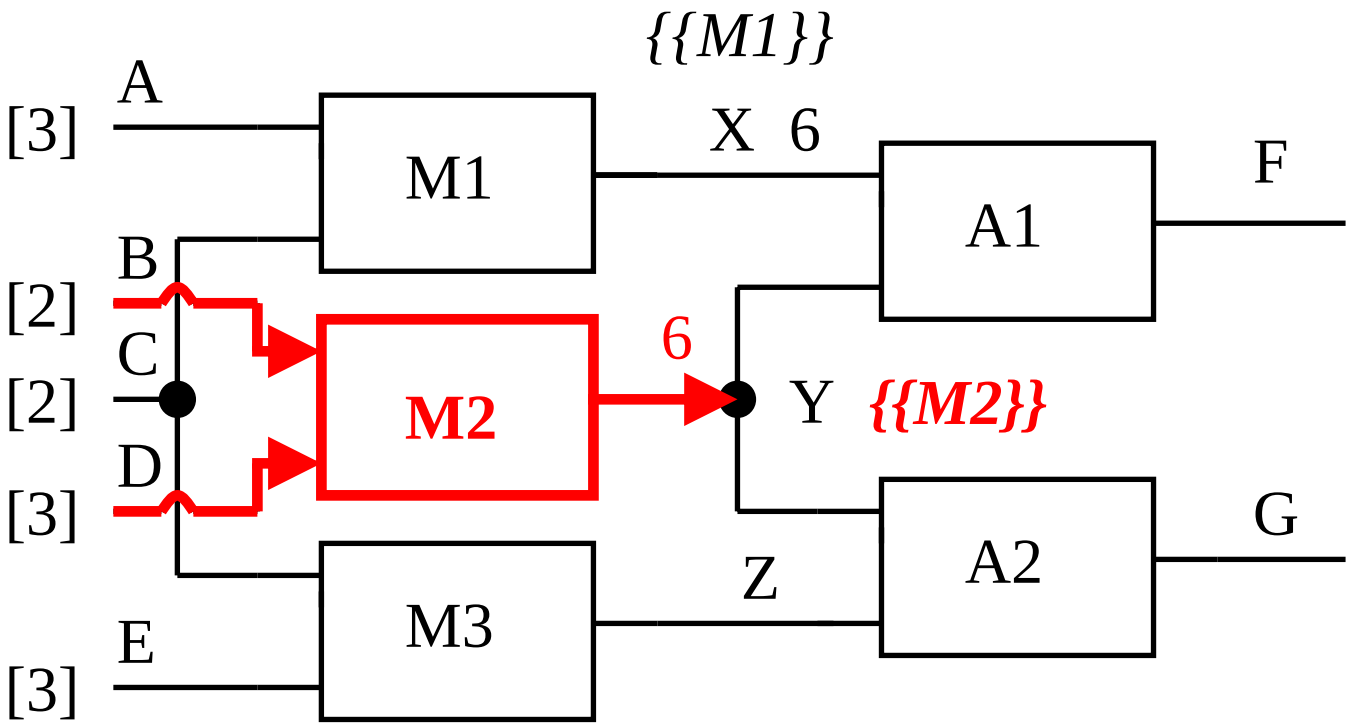
- The problem solver performs inferences
- The ATMS records the dependencies between inferences
  - Introduce observations as facts
  - Support each local propagation with a correctness assumption for the component
  - Label of a node:(minimal) environments that entails the prediction
    - Records components that support prediction
    - Avoids recomputation
  - Symptoms: produce NOGOODS

NOGOODS are the MINIMAL CONFLICTS

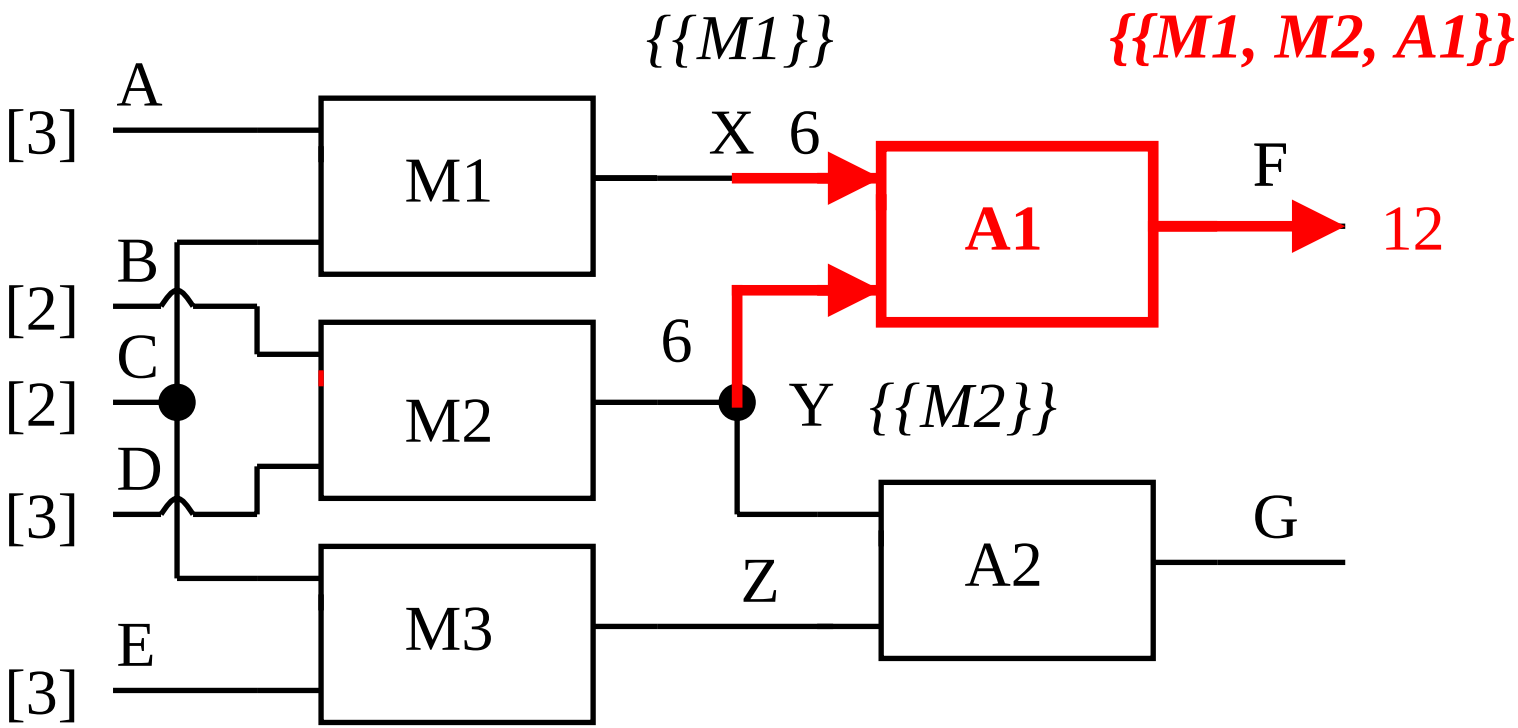
# Conflicts generation, detailed model, first minimal conflict (I)



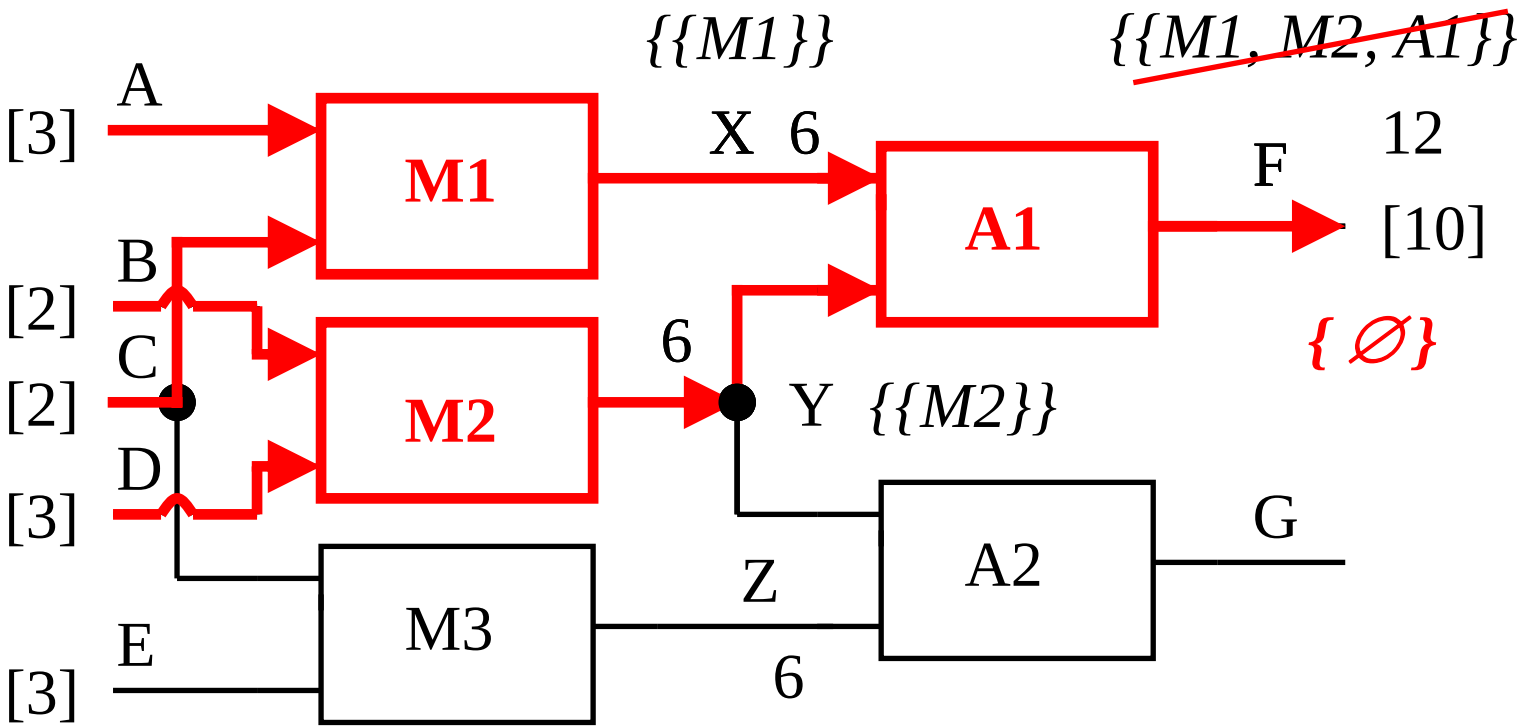
# Conflicts generation, detailed model, first minimal conflict (II)



# Conflicts generation, detailed model, first minimal conflict (III)



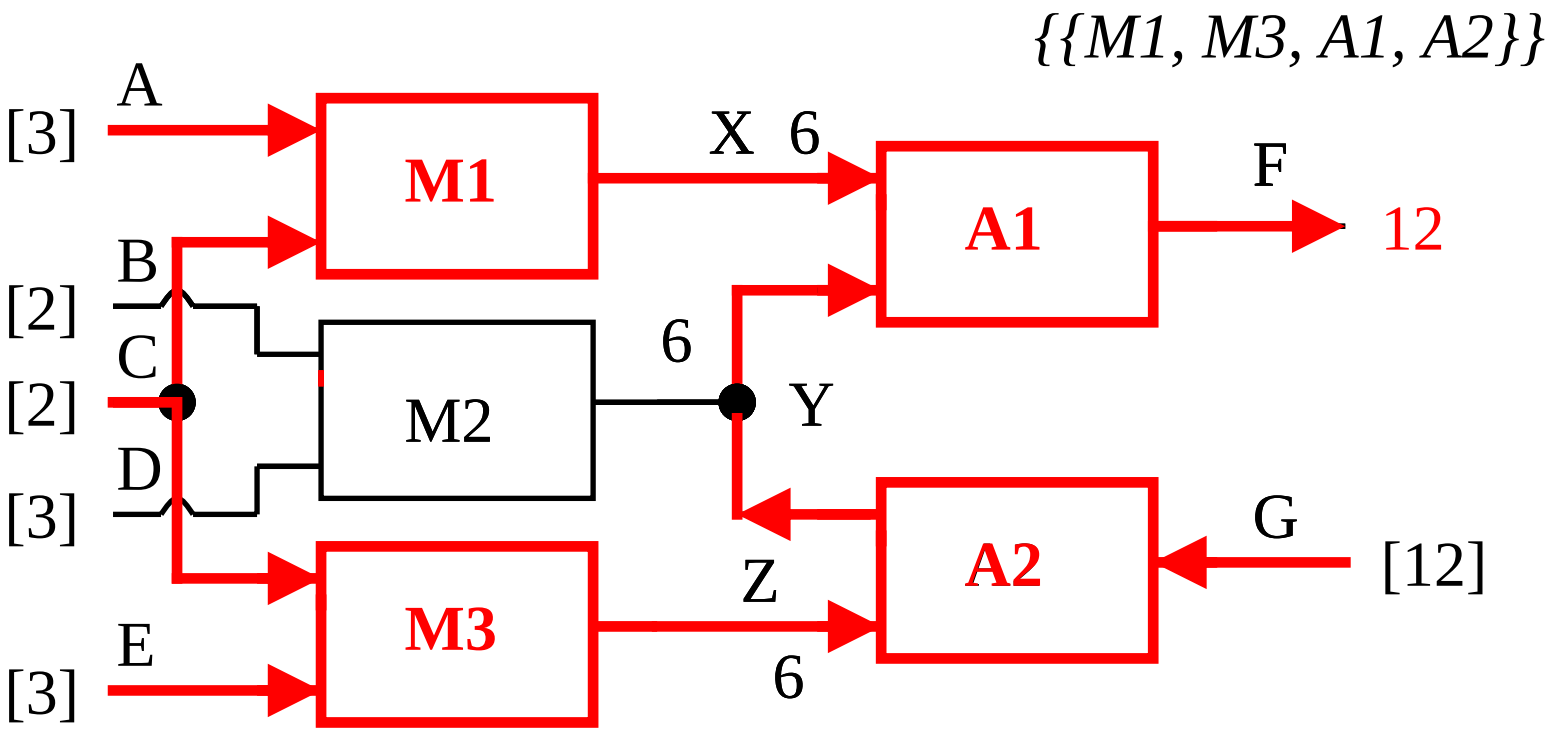
# Conflicts generation, detailed model, first minimal conflict (IV)



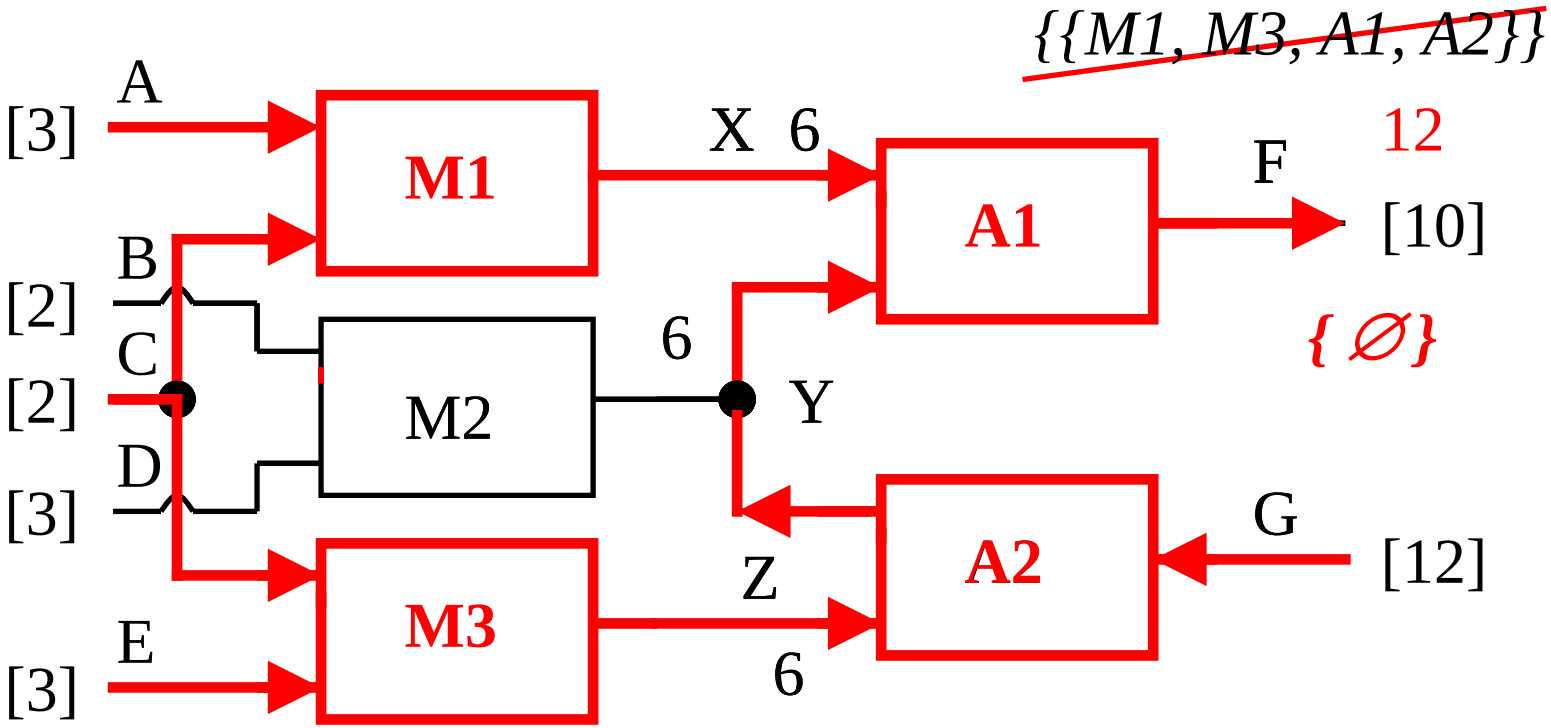

$F=[10] \wedge F=12$

**{M1, M2, A1}**

# Conflicts generation, detailed model, second minimal conflict (I)



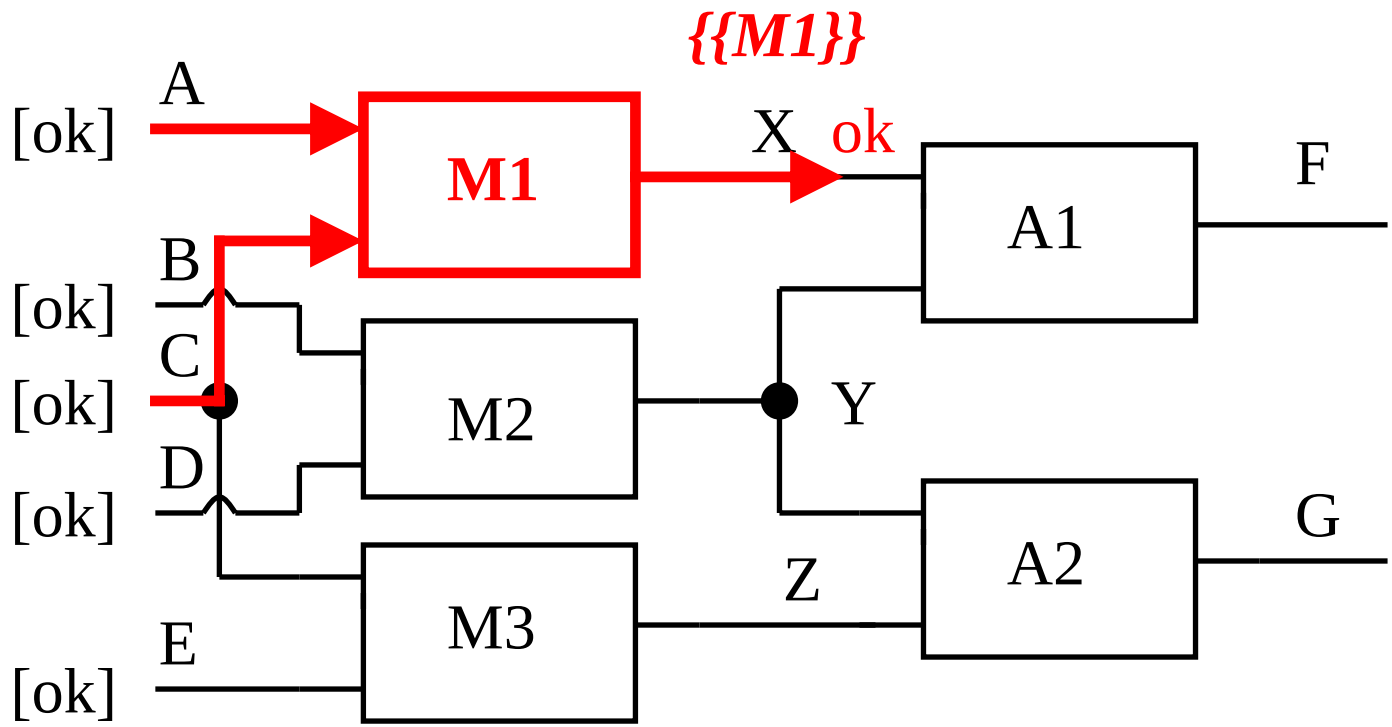
# Conflicts generation, detailed model, second minimal conflict (II)



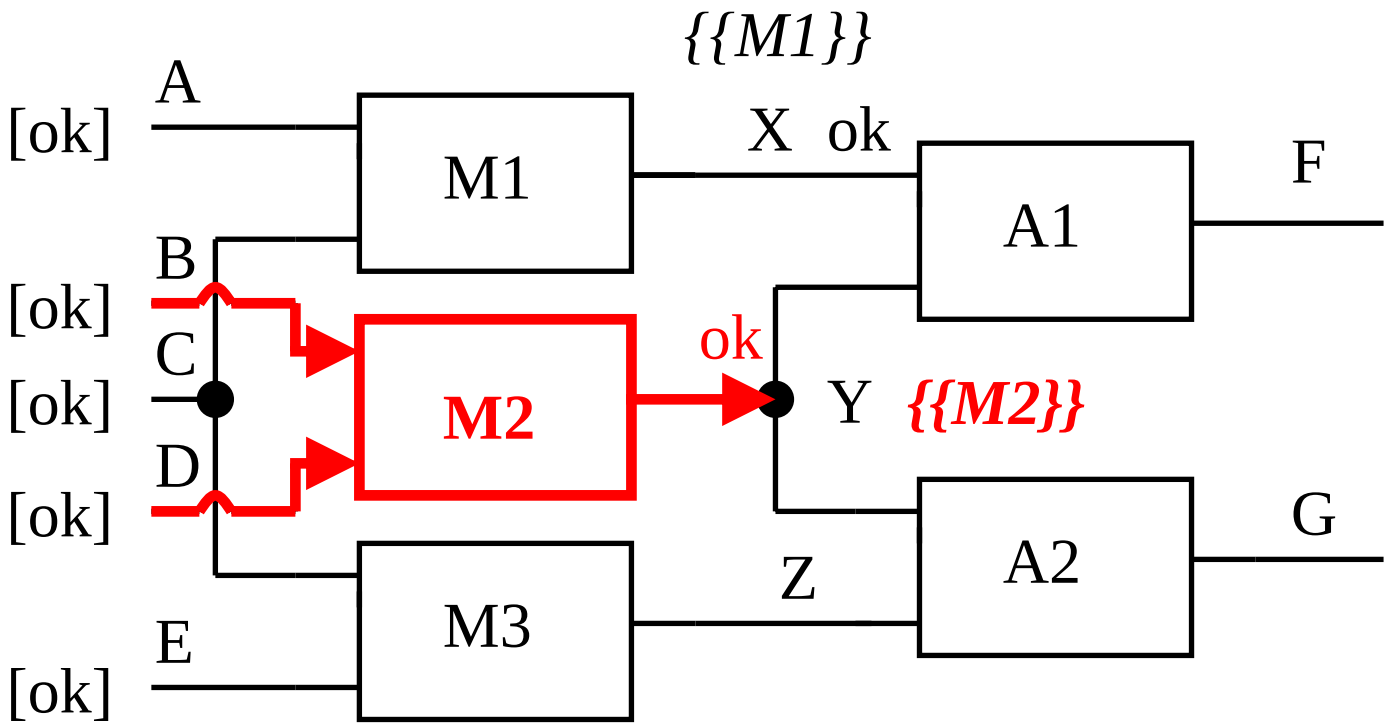

F=[10] ∧ F=12

**{M1, M3, A1, A2}**

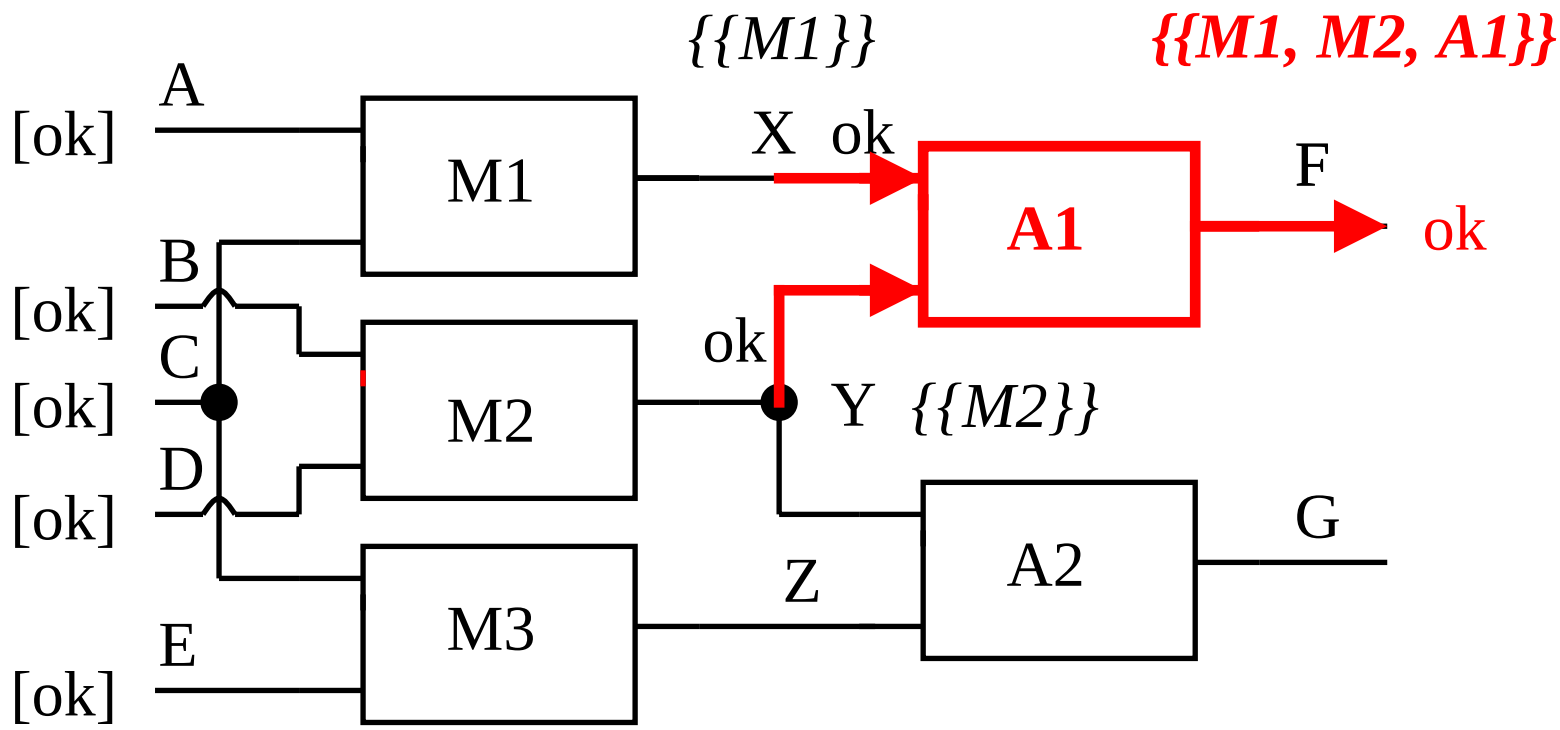
# Conflicts generation, abstract model, first minimal conflict (I)



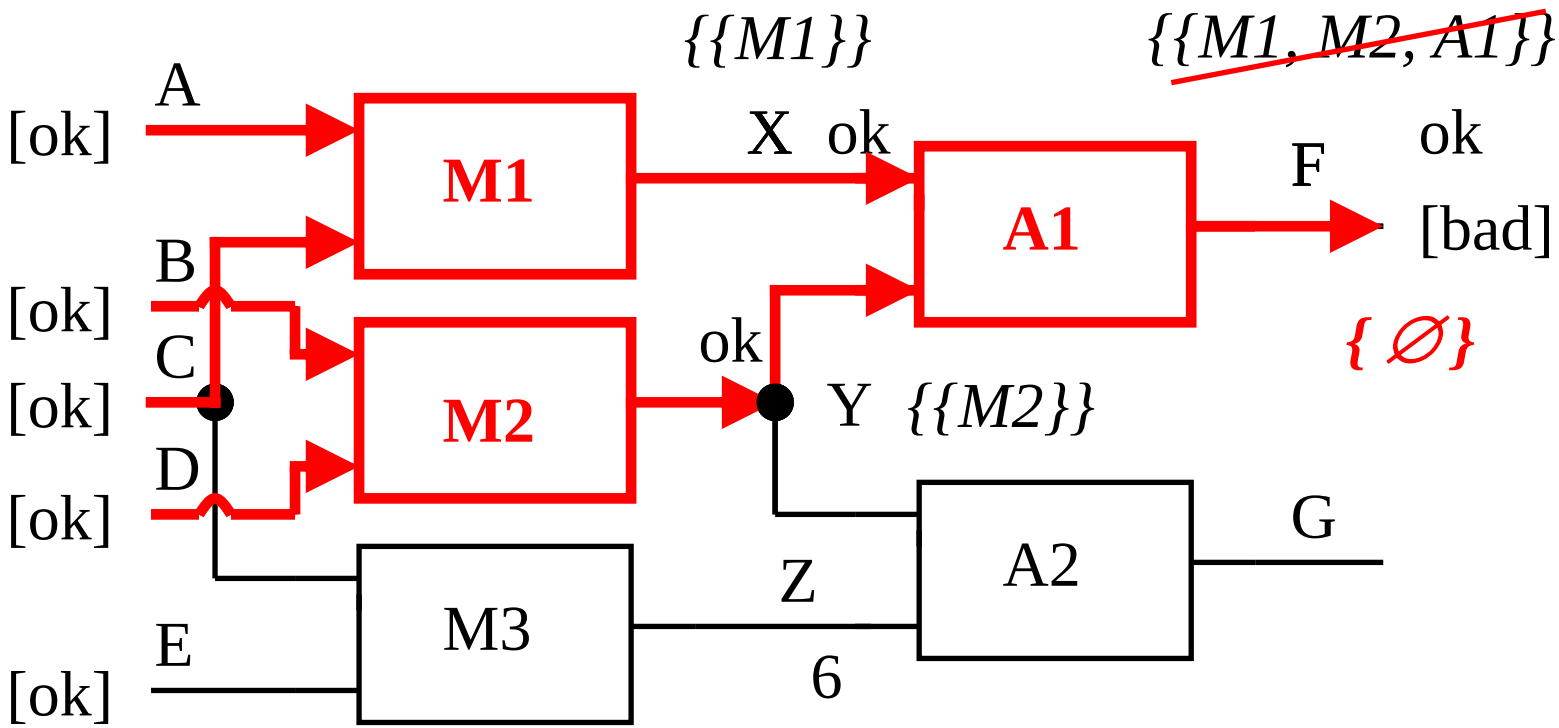
# Conflicts generation, abstract model, first minimal conflict (II)



# Conflicts generation, abstract model, first minimal conflict (III)



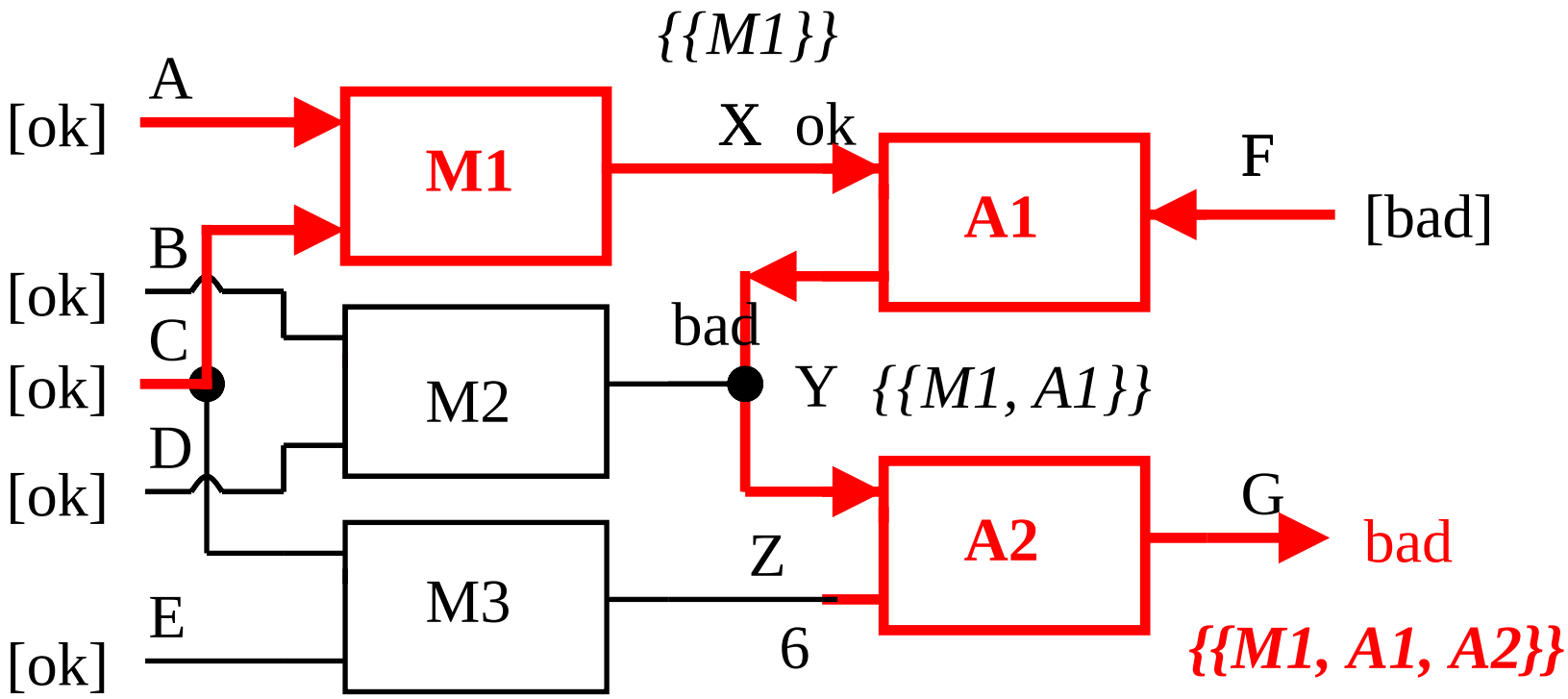
# Conflicts generation, abstract model, first minimal conflict (IV)



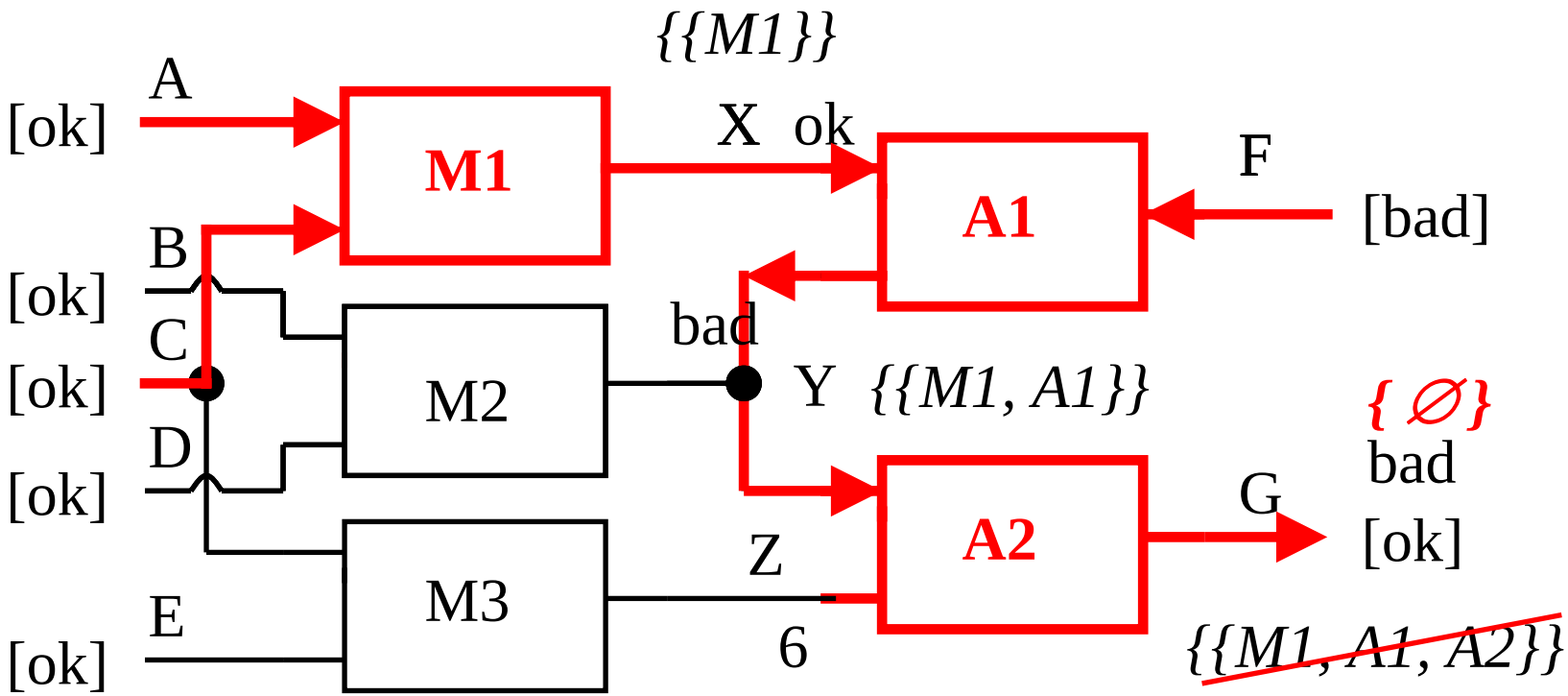

$F=[bad] \wedge F=ok$

**{M1, M2, A1}**

# Conflicts generation, abstract model, second minimal conflict (I)



# Conflicts generation, abstract model, second minimal conflict (II)




G=[bad] ∧ G=ok

**{M1, A1, A2}**

# Candidates

- Candidate: hypothesis of how the device differs from model
  - Represented as a set of assumptions
    - Assumptions included: false
    - Assumptions not included: true

Candidate example: {M2, A2}

Meaning: M2, A2 are faulty
M1, M3, A1 are correct

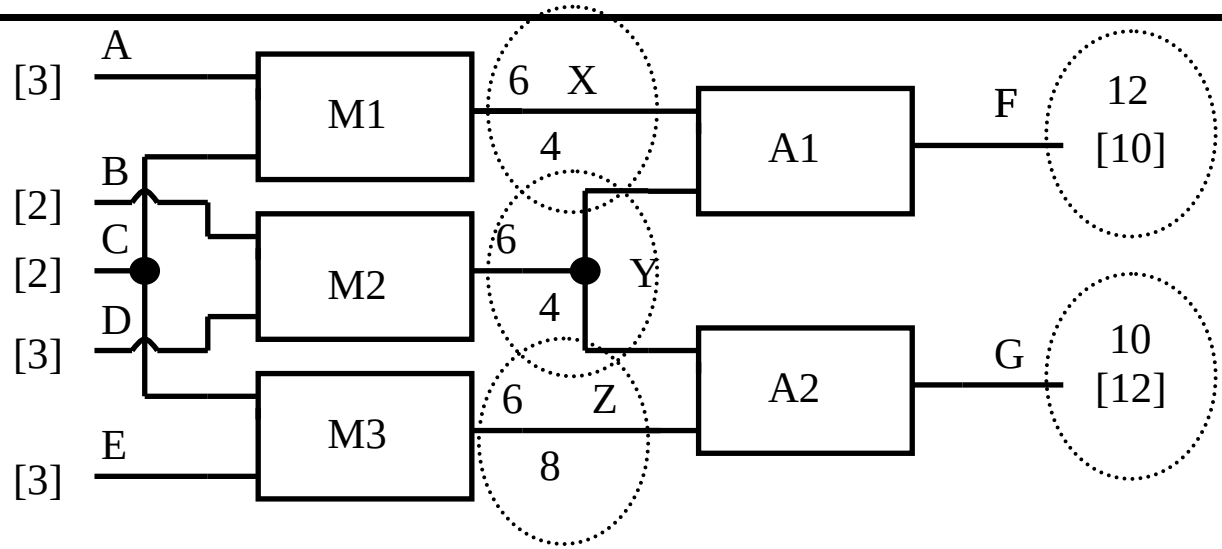
- Diagnosis: identify every candidate consistent with observations

# Candidate generation

- Each candidate has to account for all conflicts
- Each candidate has to retract at least one correctness assumption out of each conflict
- Construct candidates as Hitting Set of (minimal) conflicts
  - $C_a$ candidate, $C_i$ conflict, $C_a \cap C_i \neq \emptyset \ \forall \ C_i$
  - $\forall \ C_a$, $C_a \subset \cup_i C_i$
- Each superset of a candidate is also a candidate:

  Minimal candidates: minimal hitting set of minimal conflicts

# Candidate generation example




- Minimal conflicts

{ M1, A1, M2 }

{ M1, A1, M3, A2 }

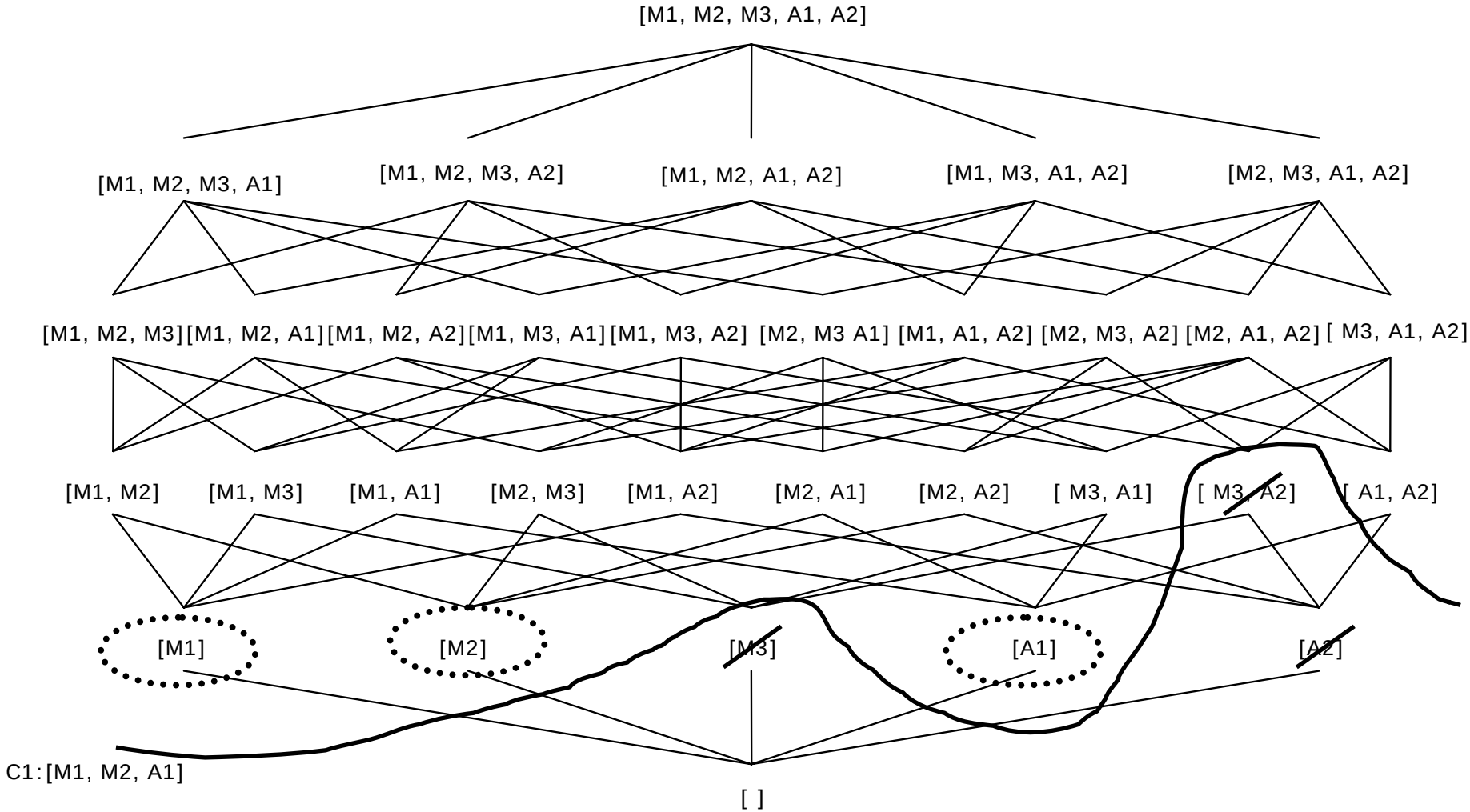
- Minimal candidates

{M1}, {A1}, {M2, M3}, {M2, A2}

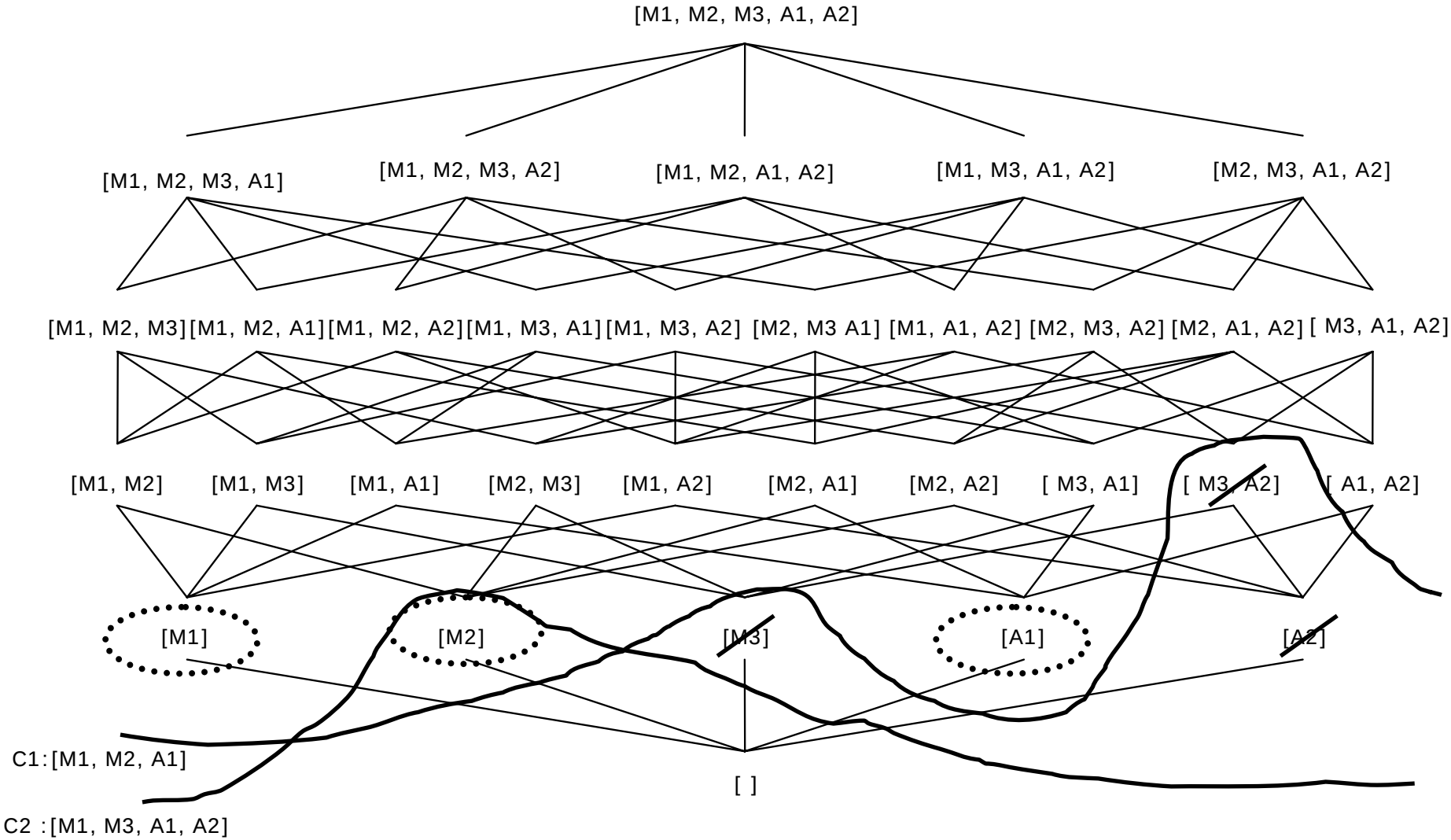
# Conflict Directed Search

1. Let M be the set of putative minimal diagnoses, initially containing only [].
2. If no more minimal conflicts, the M is the set of minimal diagnoses
3. For every new minimal conflict C
    1. For every diagnosis D in M
        1. If D identifies one component in C as faulted, do nothing.
        2. Else remove D from M and add to M all D' which have some component of C faulted.
    2. Remove duplicates from M
4. Go to 2.

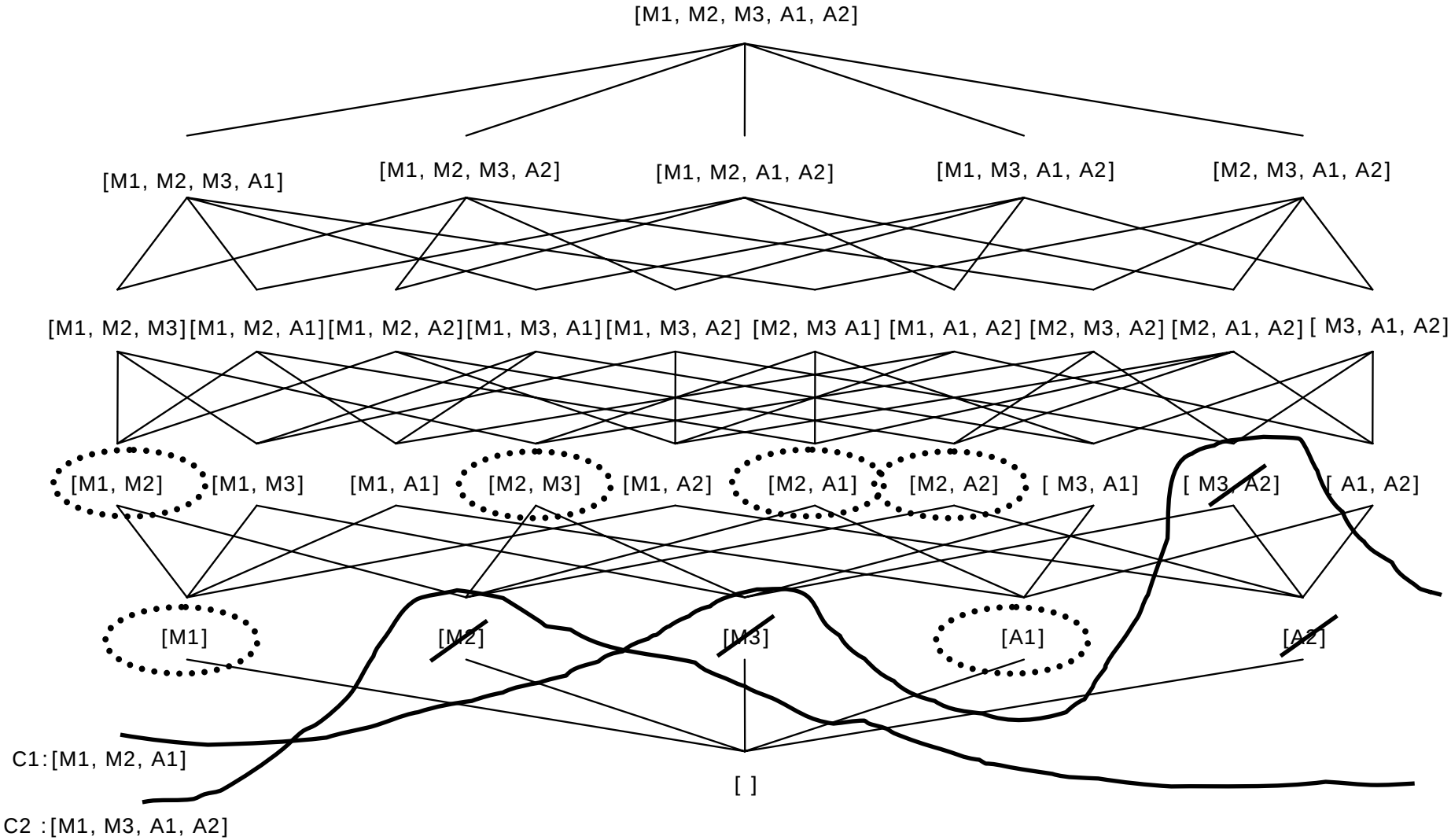
# Candidate lattice: parsimonious representation (I)



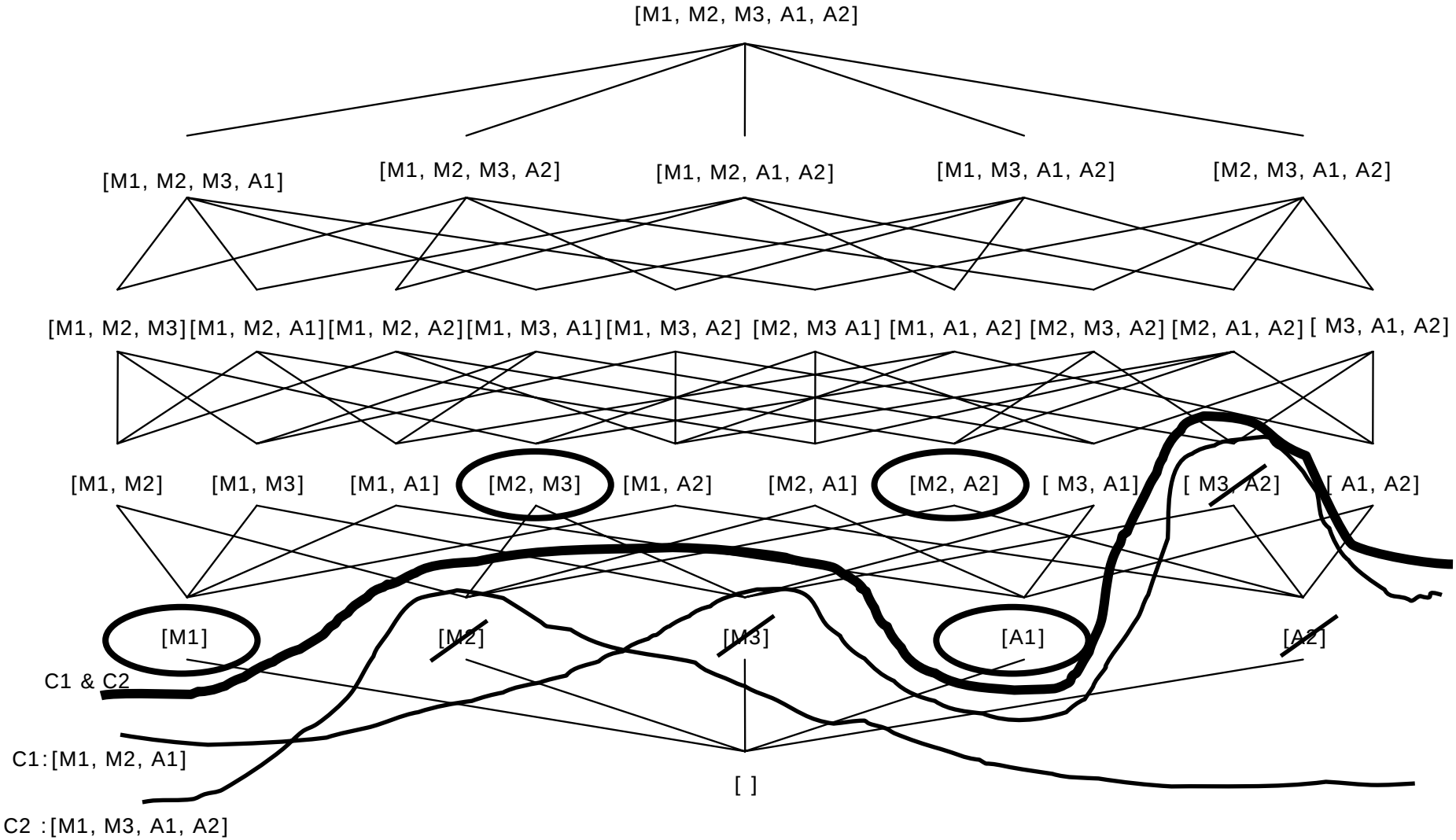
# Candidate lattice: parsimonious representation (II)



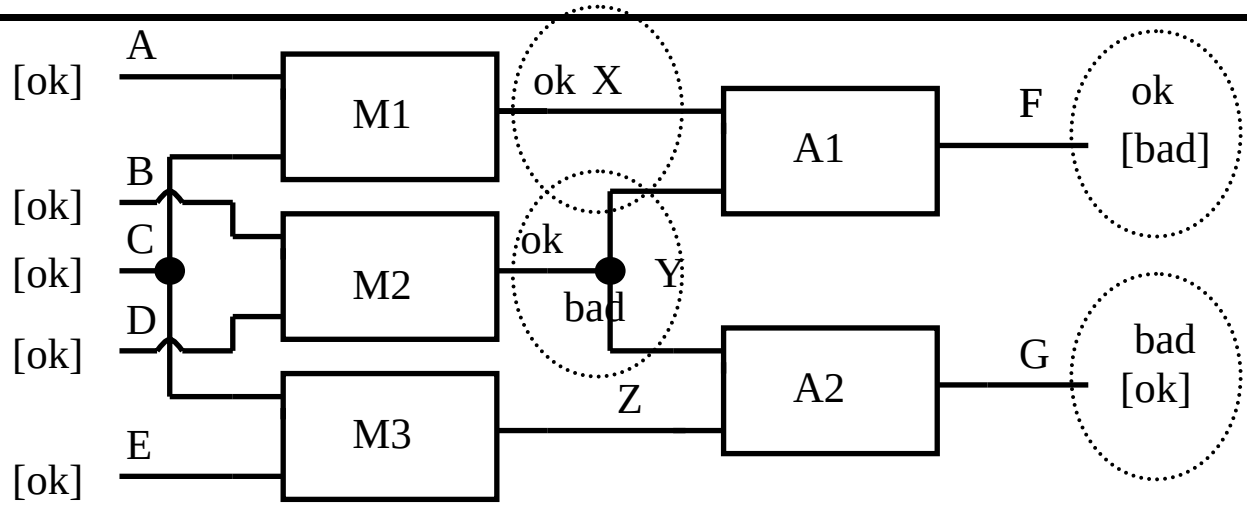
# Candidate lattice: parsimonious representation (III)



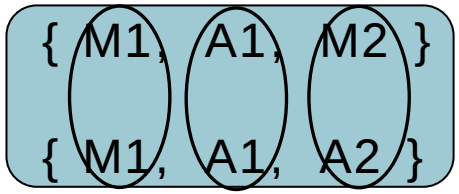
# Candidate lattice: parsimonious representation (IV)



# Candidate generation abstract model




- Minimal conflicts



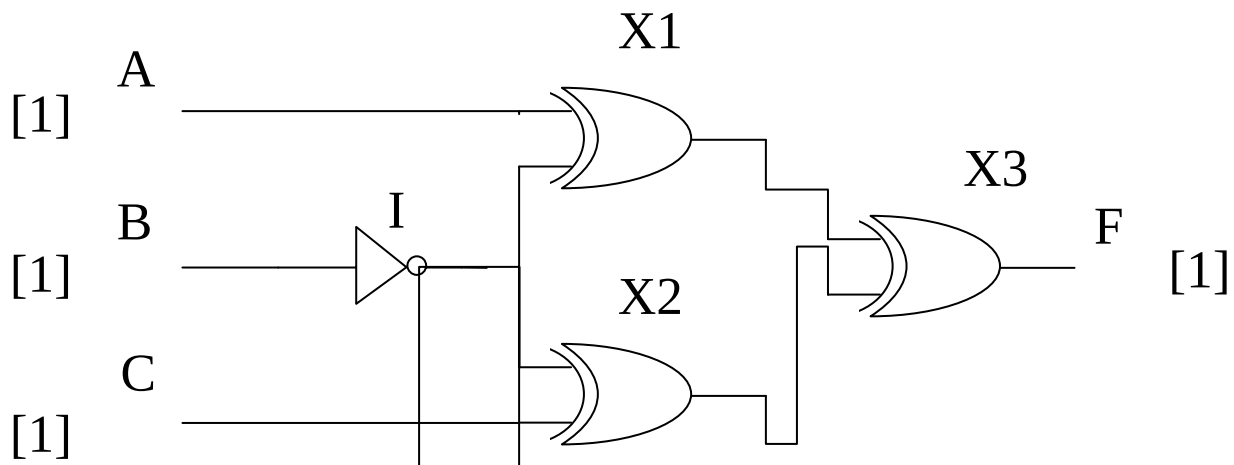

- Minimal candidates

{M1}, {A1}, {M2, A2}

# Candidate generation: problems

- Undetected symptoms
  - Insufficient observations
    - Imprecise
    - Not available
  - Insufficient models
    - Quantitative accuracy
    - Qualitative ambiguity
- Limitations of conflict generation
  - Inherent in the prediction algorithm
  - Inherent in the model

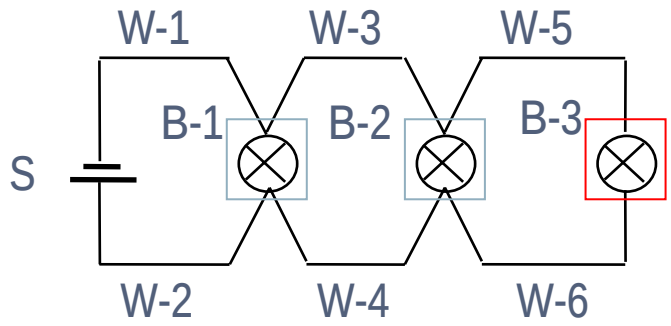
# Conflict generation: limitations due to local propagation




- Conflicts: {I, X1, X2, X3}
- Candidates: {I }, {X1}, {X2}, {X3}

- {I} should not be a candidate
- {X1, X2, X3} ought to be a conflict

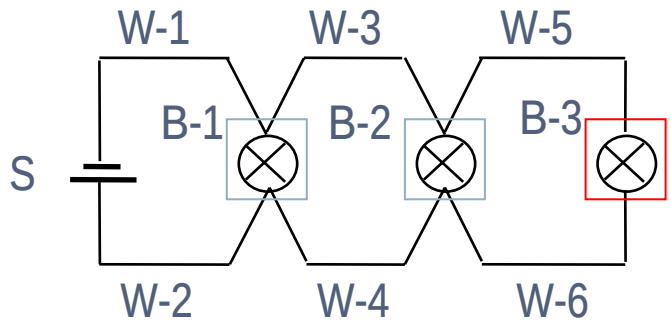
# Conflict generation: limitations due to the model (I)




- Observations: B1, B2 OFF,B3 ON
- Minimal conflicts

  {S, W1, B1, W2}, {S, W1, W3, B2, W4, W2},
  {B3, W5, B2, W6}, {B3,W5, W3, B1, W4, W6}
- ¡22 minimal candidates!
  - {B1, B2}, {S, B3}, {W1, W5} (?)

# Conflict generation: limitations due to the model (II)




- Observations: B1, B2 OFF,B3 ON
- Candidate {S, B3}
  - Logically possible
  - Physically impossible
- Due to the absence of information about faulty behaviour (only models of correct behaviour)