
Diagnostic: un raisonnement logique

Yannick Pencolé

CNRS-LAAS, Université de Toulouse, FRANCE

11 Juin 2012

Formation EDSYS : module diagnostic (20h)

Objectifs de cette formation :

- Introduction la notion de raisonnement diagnostic
- Fournir un **spectre large** des outils et méthodes pour le diagnostic

Intervenants :

- Marie-Veronique Lelann (Professeur INSA, groupe DISCO, LAAS)
- Louise Travé-Massuyès (Directrice de recherche CNRS, groupe DISCO, LAAS)
- Xavier Pucel (Ingénieur recherche, DCSD, ONERA)
- Yannick Pencilé (Chargé de Recherche CNRS, DISCO, LAAS)

Organisation du module

- Lundi 11 : 9h30-12h30 Diagnostic raisonnement logique I (Yannick Pencolé)
- Lundi 11 : 14h00-15h00 : Diagnostic raisonnement logique II (Yannick Pencolé)
- Lundi 11 : 15h00-17h00 : Diagnostic et Apprentissage (Marie-Veronique Lelan)

- Mardi 12 : 9h30-12h30 : Diagnostic pour les systèmes continus (Yannick Pencolé)
- Mardi 12 : 14h00-17h00 : Diagnostic pour les systèmes à événements discrets (Yannick Pencolé)

- Mercredi 13 : 8h30-12h30 : Diagnosticabilité (Xavier Pucel)
- Mercredi 13 : 14h00-18h00 : Vers l'unification des théories en diagnostic (Louise Travé-Massuyès)

Qu'est-ce que le diagnostic ?

Demandons à wikipedia :

- « Le diagnostic est le **raisonnement** menant à l'identification de la **cause** (l'origine) d'une défaillance, d'un problème ou d'une maladie, ou tout simplement à la **détermination** d'une espèce biologique par rapport à une autre (**taxinomie**), à partir des caractères ou **symptômes** relevés par des **observations**, des contrôles ou des tests. »

Demandons à google.fr les premiers liens :

- 1 diagnostic médical (maladies, symptômes, recherche de cause)
- 2 diagnostic immobilier (détermination de la classe d'un appartement, isolé/pas isolé, salubre/insalubre)

Un peu d'étymologie

Le mot diagnostic vient du Grec ancien διά-γνωση

- διά préfixe de **séparation**
- γνωση **connaissance**

Étymologiquement, diagnostic \equiv **discernement**.

Séparer le bien du mal, le faux du vrai, le normal de l'anormal....

Au final, le diagnostic c'est..

- C'est un raisonnement
- Nécessite des **observations** sur le système
- Nécessite de définir des **objectifs**
 - 1 Recherche de causes (maladies, problèmes, pannes, fautes)
 - 2 Détermination de propriétés (estimation d'états sûrs, critiques, anormaux)
- Le diagnostic n'est utile que si l'on s'en sert
 - 1 pour des réparations/restructurations (recherche de la source des défaillances, des problèmes)
 - 2 pour des décisions (estimation de la véracité d'une propriété sur l'état actuel)

3 types de raisonnement : déduction

Syllogisme d'Aristote :

- « Socrate est un homme »
- « Tout homme est mortel »
- DONC « Socrate est mortel »

C'est le principe bien connu de la **déduction**, le fameux *modus ponens*.

Élémentaire mon cher Watson !

Sherlock Holmes entra dans la chambre, vit Socrate allongé sur le lit, mort, puis il dit :

« Socrate est mortel »

De part son passé, son expérience, Sherlock Holmes savait :

« Tout homme est mortel »

Alors le docteur Watson lui demanda :

« quelles sont vos conclusions, Holmes ? »

Sherlock répliqua :

« Élémentaire mon cher Watson, il est mort parce que c'est un homme. »

Holmes est-il le roi de la déduction ?

Et bien non ! mon cher Watson !

Holmes est le roi de l'**abduction**.

- « Tout homme est mortel »
- « Socrate est mortel »
- IL EST POSSIBLE QUE « Socrate est un homme »

La conclusion de Holmes est ... une hypothèse. « Socrate est un rat » est une hypothèse toute aussi valide sachant que « Tout rat est mortel ».

Et le dernier type de raisonnement : induction, apprentissage

- « Socrate est un homme »
- « Socrate est mortel »
- EN GENERALISANT : « Tout homme est mortel »

Diagnostic : raisonnement abductif

Étant donné,

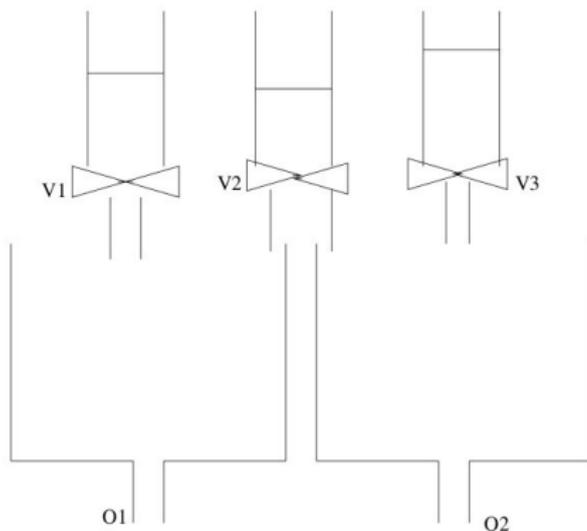
- 1 un **système** (et la **connaissance** que l'on en a au travers de modèle)
- 2 un ensemble d'**observations** (des mesures, des alarmes, des indices)

Le diagnostic consiste à

- établir une ou plusieurs **hypothèses** (candidats) sur les propriétés du système
 - ▶ propriétés type : fonctionnement normal/anormal, fonctionnement critique/non critique....
- telle que chaque hypothèse **explique** les observations
 - ▶ « Socrate est un homme » est une hypothèse
 - ▶ elle explique l'observation « Socrate est mortel » en s'appuyant sur le modèle « Tout homme est mortel ».

Exemple

Mélanges, 3 réservoirs, 3 vannes, 2 entonnoirs.



$COMPS = \{V_1, V_2, V_3\}$ (position : ouvert, fermé).

Première expérience :

OBS= O_1, \bar{O}_2 (O_1 coule, O_2 ne coule pas).

DIAGNOSTIC : V_1 ouverte, V_2 fermée, V_3 fermée.

Seule configuration possible \Rightarrow diagnostic non ambigu.

Deuxième expérience :

OBS= O_1, O_2 (O_1 coule, O_2 coule).

DIAGNOSTIC1 : V_1 ouverte, V_2 fermée, V_3 ouverte.

DIAGNOSTIC2 : V_1 fermée, V_2 ouverte, V_3 fermée.

DIAGNOSTIC3 : V_1 ouverte, V_2 ouverte, V_3 fermée.

DIAGNOSTIC4 : V_1 fermée, V_2 ouverte, V_3 ouverte.

DIAGNOSTIC5 : V_1 ouverte, V_2 ouverte, V_3 ouverte.

Situation mal (non complètement) identifiée, diagnostic ambigu.

Et Dr House alors ?

- Dr House : le Sherlock Holmes de la médecine, **diagnosticien**

Modèle d'un épisode

- 1 Un patient est malade → premiers symptômes
- 2 Réunion de Dr House et de ses ouailles
 - ▶ Listing des maladies POSSIBLES (candidats) sur le tableau.
- 3 Suite de tests/traitements/enquêtes (diagnostic actif) → nouvelles observations
- 4 Dr House raye les hypothèses invalidées au cours de l'épisode.
- 5 Fin à l'américaine : il en reste toujours qu'une !



Très beaux exemples de diagnostic (maladies multiples, masquage de symptômes)

Historique des travaux en automatisation du diagnostic

- années 70 : approches heuristiques (système expert)
 - ▶ Base de connaissance = ensemble de règles abductives
 - ▶ Si *fièvre* > 38 et mal de cou alors grippe. etc.
 - ▶ Diagnostic : déterminer les règles à instancier en fonction des observations
- années 80 : diagnostic à base de modèles (systèmes statiques)
- années 90 : diagnostic à base de modèles (systèmes dynamiques continus/discrets)
- années 00 : diagnostic à base de modèles (systèmes hybrides)

Deux communautés scientifiques différentes mais qui convergent l'une vers l'autre :

- communauté Intelligence Artificielle (aspect raisonnement)
- communauté Automatique (aspect détection, estimation)

Concept de système et de modèle

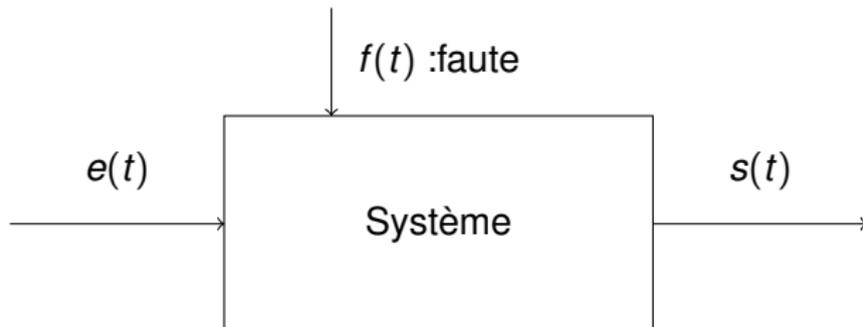
- Qu'est-ce qu'un système ?
 - ▶ C'est une **réalité**
 - ▶ Assemblage de composants

An aggregation or assemblage of things so combined by nature or man as to form an integral or complex whole » [Encyclopedia Americana]

- Et un modèle ?
 - ▶ Une **représentation** d'un système
 - ▶ Modèle adapté à la tâche à effectuer :
 - compréhension, simulation, planification, commande, diagnostic, suivi
- Distinction entre système et modèle : **essentielle**
(en général mais plus particulièrement en diagnostic)

Concepts de défaut/panne/faute/erreur/défaillance

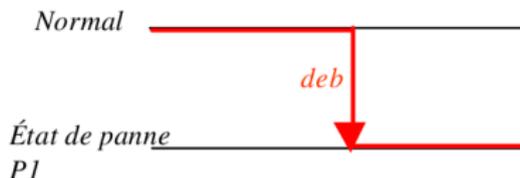
- Un système réalise une fonction
 - ▶ Si la fonction n'est pas réalisée, le système est **défaillant**
 - ▶ Perte de fonction
- La défaillance est due à des **erreurs** dans le comportement du système
- Les erreurs sont dues à des **fautes/pannes/défauts** qui surviennent sur le système.



Panne permanente

Definition

Une panne est **permanente** s'il n'existe aucun moyen de la réparer ou aucune raison qu'elle disparaisse pendant la période de diagnostic.



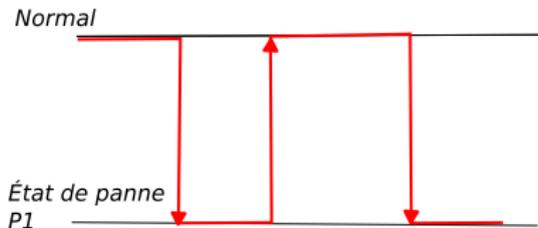
Exemple

Diagnostic en-ligne d'une automobile : une panne moteur est permanente. Cette panne ne peut pas disparaître sans une intervention extérieure (garage)

Panne intermittente

Definition

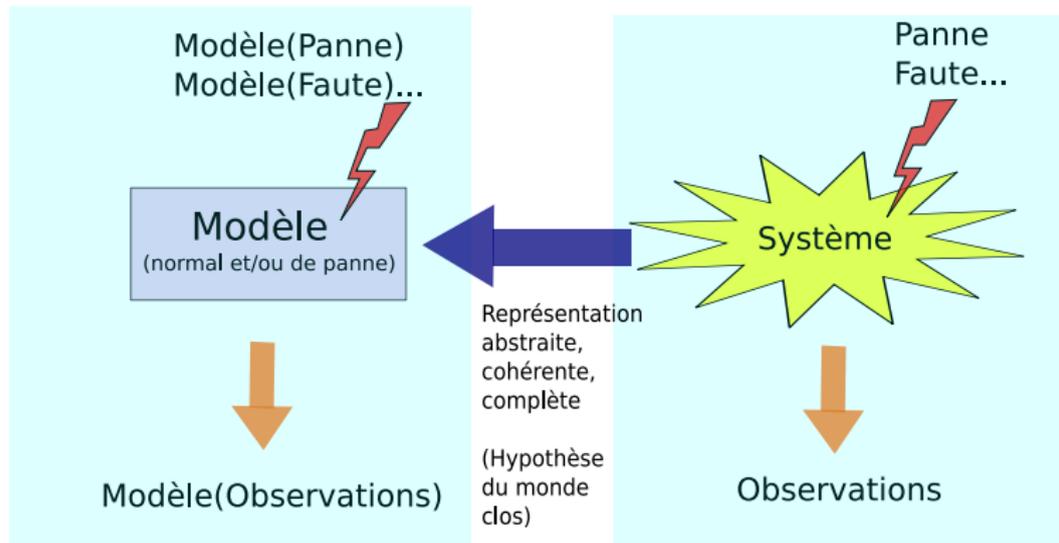
Une panne est **intermittente** si elle peut apparaître/disparaître pendant la période de diagnostic à plusieurs reprises.



Exemple

Diagnostic en-ligne d'une automobile : un mauvais contact filaire entre calculateurs peut apparaître et disparaître (vibration, humidité) au cours d'un même trajet.

Concept du diagnostic à base de modèle



Diagnostic à base de modèle =

- 1) Confronter les observations au modèle (cohérence)
- 2) "Remonter" à la cause (abduction, modèle de panne)

Concept du diagnostic à base de modèle (2)

- La majeure partie des méthodes de diagnostic s'appuient sur un modèle comportemental (dynamique, statique) :
 - ▶ équations différentielles,
 - ▶ système à événements discrets,
 - ▶ équations algébriques,
 - ▶ équations logiques.
- On trouve parfois un modèle structurel décrivant des liens entre les composants (par exemple, alimentation électrique commune).
- On en tire alors des règles de fonctionnement normal et en présence de faute.

Objectif de diagnostic de pannes

Fonction des connaissances, le diagnostic a plusieurs sous-objectifs :

- 1 **Détection** : détecter la présence d'une panne (sans identifier ce qu'elle est)
- 2 **Localisation** : identifier le composant où la panne s'est produite
- 3 **Identification** : identifier la nature, le type de la panne
- 4 **Propagation** : déterminer toutes les conséquences de la panne (relation cause-effet)

Soyons logique

Logic ? But what is it ?

Principle

Logics are **formal languages** for representing information such that conclusions can be drawn. To define a logic, we need :

- 1 **syntax** : how a **sentence** of the logic looks like ?
- 2 **semantic** : what is the meaning of the sentence ?
 - ▶ Given a **world**, is the sentence **true** or **false** ?

Example

The language of arithmetic

Syntax :

$x + 2 \geq y$ is a sentence ;

$x^2 + y \geq y$ is not a sentence

Semantic :

$x + 2 \geq y$ is true in a world where $x = 7, y = 1$

$x + 2 \geq y$ is false in a world where $x = 0, y = 6$

Entailment

Entailment means that one sentence (α) **follows from** other sentences (KB) and is denoted :

$$KB \models \alpha$$

We say that the Knowledge Base KB **entails α if and only if** α is `true` in all worlds where KB is `true`. Entailment is a relationship between sentences (i.e. syntax) that is based on semantics.

Example

Knowledge Base = { "The car is blue" "The bicycle is green or yellow" }

KB entails sentences α like :

- "The car is blue"
- `true`
- "The car is blue or the bicycle is yellow"

The sentence "The car is blue and the bicycle is yellow" is not entailed by KB .

World in Logic = Model

Definition

We say m is a **model** of a sentence α if α is true in the world m . We denote by $M(\alpha)$ the set of models

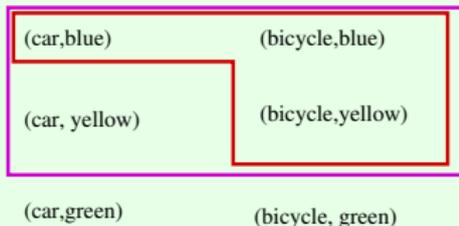
Property

KB entails α if and only if $M(KB) \subseteq M(\alpha)$.

Example

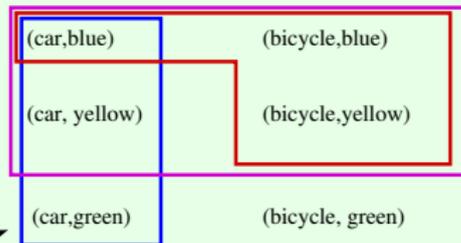
$KB = \{ \text{"The car is blue"}, \text{"The bicycle is green or yellow"} \}$

Possible models of KB



$a = \text{"The car is blue"}$

Possible models of sentence a



not a model of KB



Inference

Definition

Inference : A sentence β can be inferred from another sentence α by some inference algorithm i . This is denoted :

$$\alpha \vdash_i \beta$$

Definition

Soundness : An inference algorithm is **sound** if it produces entailed sentences

Definition

Completeness : An inference algorithm is **complete** if it can derive all the sentences which it entails.

Well-known logics

- 1 Propositional logic
- 2 First-order logic
- 3 Default logic
- 4 Circumscription
- 5 Temporal logic
- 6 Modal logic
- 7 ..

Every logic has its Pros and Cons (expressivity, soundness and completeness of inference algorithm)

Logical equivalence

Definition

Two sentences α , β are **logically equivalent** IF AND ONLY IF they are true in the same models. α entails β and vice-versa.

Logical equivalent sentences

$(\alpha \wedge \beta)$	\equiv	$(\beta \wedge \alpha)$	commutativity of \wedge
$(\alpha \vee \beta)$	\equiv	$(\beta \vee \alpha)$	commutativity of \vee
$((\alpha \wedge \beta) \wedge \gamma)$	\equiv	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of \wedge
$((\alpha \vee \beta) \vee \gamma)$	\equiv	$(\alpha \vee (\beta \vee \gamma))$	associativity of \vee
$\neg(\neg\alpha)$	\equiv	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	\equiv	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	\equiv	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	\equiv	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	\equiv	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	\equiv	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	\equiv	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of \wedge over \vee
$(\alpha \vee (\beta \wedge \gamma))$	\equiv	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of \vee over \wedge

Validity and satisfiability

Definition

A sentence is **valid** if it is true in **ALL models** :

$$a \vee \neg a, a \Rightarrow a, (a \wedge (a \Rightarrow b)) \Rightarrow b$$

Deduction theorem

KB entails α ($KB \models \alpha$) iff the sentence $KB \Rightarrow \alpha$ is valid. Validity is then connected to inference.

Definition

① A sentence is **satisfiable** if it is true in **SOME models**. A valid sentence is satisfiable, but a satisfiable sentence may be not valid.

② A sentence is **unsatisfiable** if it is true in **NO models** :

$$a \wedge \neg a, (a \wedge b) \Leftrightarrow (\neg a \wedge c)$$

Satisfiability and inference

KB entails α ($KB \models \alpha$) iff the sentence $KB \wedge \neg \alpha$ is unsatisfiable.

Inference rules : examples

Example

Modus Ponens :

$$\frac{a, a \Rightarrow b}{b}$$

And-elimination :

$$\frac{a \wedge b}{a}$$

Factoring :

$$\frac{a \vee a}{a}$$

Logical equivalences :

$$\frac{\neg a \vee \neg b}{\neg(a \wedge b)}$$
$$\frac{a \Leftrightarrow b}{a \Rightarrow b \wedge b \Rightarrow a}$$

Resolution algorithm

Definition

Proof by contradiction : given KB , to prove α , we prove that $KB \wedge \neg\alpha$ is not satisfiable.

Example

Symbols :

- Und : “The students have understood this lecture”
- Gt : “I am a good teacher”
- $Party$: “The students went to a party last night”

Knowledge base :

$$KB = (\neg Und \Leftrightarrow (\neg Gt \vee Party)) \wedge Und$$

Query to prove : *I am a good teacher*

$$\alpha = Gt$$

First-order logic

Whereas propositional logic assumes world contains *facts*, first-order logic (like natural language) assumes the world contains

- **Objects** : people, houses, numbers, theories, colors, cricket games, centuries ... and me, and cars !!
- **Relations** : red, round, bogus, prime, multistoried ..., brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, ... and blue !!
- **Functions** : father of, third inning of, one more than, end of ... and friend of, sister of !!

Syntax of First-Order Logic

Basic elements

- 1 Constants : *KingJohn, 2, ANU, Yannick* ...
- 2 Predicate : *Sister, >* ...
- 3 Functions : *Sqrt, FriendOf* ...
- 4 Variables : *x, y, a, b* ...
- 5 Connectives : $\vee, \wedge, \neg, \Rightarrow, \Leftrightarrow$
- 6 Equality : $=$
- 7 Quantifiers : $\forall \exists$

Syntax of First-Order Logic

Term

A **term** represents an object in FOL. Its syntax is :

- a constant, or
- a variable, or
- a function of terms $function(term_1, \dots, term_n)$

Atomic sentence

An **atomic sentence** represents an elementary relation between terms. Its syntax is :

- a predicate $predicate(term_1, \dots, term_n)$
- an equality of terms $term_1 = term_2$

Example

$Brother(KingJohn, RichardTheLionheart)$
 $> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))$
 $carOf(friendOf(oneSisterOf(Yannick))) = colorOf(Ocean)$

Syntax of First-Order Logic

Complex sentences

Complex sentences are made from atomic sentences using connectives

$$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2$$

Example

$Sibling(KingJohn, Richard) \Rightarrow Sibling(Richard, KingJohn)$

$>(1,2) \vee \leq(1,2)$

$>(1,2) \wedge \neg >(1,2)$

$Sister(Marie, Yannick) \Rightarrow CarColor(FriendOf(Marie), blue)$

Truth in first-order logic

Semantics

Sentences are true with respect to a **model** and an **interpretation**.

Model

Model contains objects (**domain elements**) and relations among them.

Interpretation

Interpretation specifies referents for

- *constant symbols* \rightarrow objects
- *predicate symbols* \rightarrow relations
- *function symbols* \rightarrow functional relations

An atomic sentence $\textit{predicate}(\textit{term}_1, \dots, \textit{term}_n)$ is true iff the objects referred to by $\textit{term}_1, \dots, \textit{term}_n$ are in the relation referred to by $\textit{predicate}$.

Fun with sentences

Example

- Brothers are siblings

$$\forall x, y \text{ Brother}(x, y) \Rightarrow \text{Sibling}(x, y)$$

- “Sibling” is symmetric

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x)$$

- “One’s mother is one’s female parent”

$$\forall x, y \text{ Mother}(x, y) \Leftrightarrow (\text{Female}(x) \wedge \text{Parent}(x, y))$$

- “A first cousin is a child of a parent’s sibling”

$$\forall x, y \text{ FirstCousin}(x, y) \Leftrightarrow \exists p, ps \text{ Parent}(p, x) \wedge \text{Sibling}(ps, p) \wedge \text{Parent}(ps, y)$$

Equality

Equality

$term_1 = term_2$ is true under a given interpretation if and only if $term_1$ and $term_2$ refer to the same object.

Example

- $1 = 2$ is satisfiable (if the symbols 1 and 2 refer to the same object in the interpretation)
- $2 = 2$ is valid

Example

Definition of *Sibling* thanks to *Parent* :

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow (\neg(x = y) \wedge \exists m, f \neg(m = f) \wedge$$

$$\text{Parent}(m, x) \wedge \text{Parent}(f, x) \wedge \text{Parent}(m, y) \wedge \text{Parent}(f, y))$$

Summary

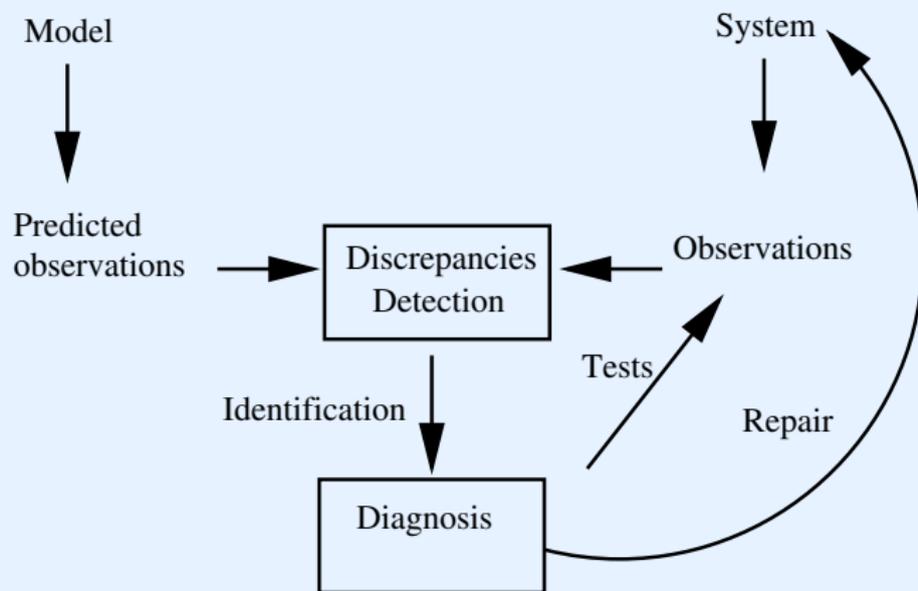
Summary

- Knowledge representation language :
 - ▶ declarative, compositional, expressive, context-independent, unambiguous
- Model : set of objects, functions and their relation
- Knowledge-base in first-order logic
 - ▶ careful process
 - 1 analyzing the **domain** (objects, functions, relations),
 - 2 choosing a **vocabulary** (interpretation)
 - 3 encoding the **axioms** (what is known in *KB*) to support the desired **inferences**

Diagnostic à base de modèle : un raisonnement logique

Rappel sur le principe de base du diagnostic à base de modèle

Model-based diagnosis



Représentation des connaissances

Definition

Le modèle d'un système est une paire ($DS, COMPS$) :

- $COMPS$ ensemble fini de constante, une constante = un composant
- SD ensemble de phrases logiques du 1er ordre
 - ▶ **Modèle comportemental** (comment un composant marche ?)
 - ▶ **Modèle structurel** (comment un composant interagit ?)

Definition

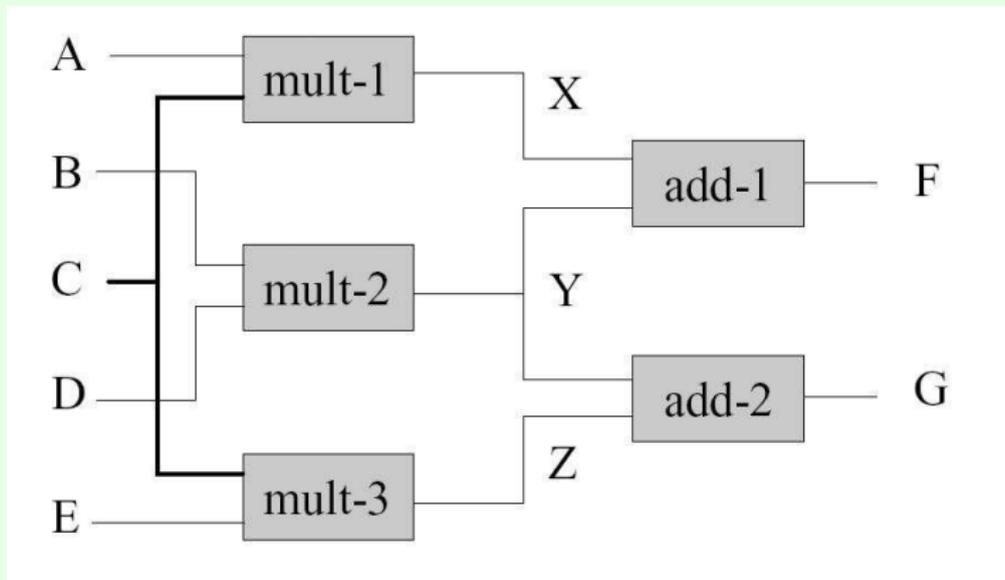
Le modèle d'un système observé is un modèle ($DS, COMPS$) auquel on ajoute OBS :

- OBS ensemble de faits logiques.
- Chaque fait logique est une observation

Exemple

Exemple

Polybox



Représentation de la connaissance : symboles

Example

$COMPS = \{a1, a2, m1, m3, m3\}$

DS prédicats :

- *Add* additionneur
- *Mult* multiplieur
- *In1* entrée 1
- *In2* entrée 2
- *Out* sortie
- *Ab* anormal
- *Sum* somme
- *Prod* produit

Représentation de la connaissance : modèle comportemental

Exemple

Toutes les variables sont universellement quantifiées

Comportement de l'additionneur :

- $Add(x) \wedge \neg Ab(x) \wedge In1(x, u) \wedge In2(x, v) \wedge Sum(u, v, w) \Rightarrow Out(x, w)$
- $Add(x) \wedge \neg Ab(x) \wedge In1(x, u) \wedge Out(x, w) \wedge Sum(u, v, w) \Rightarrow In2(x, v)$
- $Add(x) \wedge \neg Ab(x) \wedge Out(x, w) \wedge In1(x, u) \wedge Sum(u, v, w) \Rightarrow In1(x, u)$

Comportement du multiplieur :

- $Mult(x) \wedge \neg Ab(x) \wedge In1(x, u) \wedge In2(x, v) \wedge Prod(u, v, w) \Rightarrow Out(x, w)$
- $Mult(x) \wedge \neg Ab(x) \wedge In1(x, u) \wedge Out(x, w) \wedge Prod(u, v, w) \Rightarrow In2(x, v)$
- $Mult(x) \wedge \neg Ab(x) \wedge Out(x, w) \wedge In1(x, u) \wedge Prod(u, v, w) \Rightarrow In1(x, u)$

Représentation de la connaissance : modèle structurel

Exemple

Topologie, Modèle Structurel :

$COMPS = \{a1, a2, m1, m3, m3\}$

$Add(a1); Add(a2); Mult(m1); Mult(m2); Mult(m3)$

Connexion : utilisation de l'égalité

- $Out(m1, u) \wedge In1(a1, v) \Rightarrow u = v$
- $Out(m2, u) \wedge In2(a1, v) \Rightarrow u = v$
- $Out(m2, u) \wedge In1(a1, v) \Rightarrow u = v$
- $Out(m3, u) \wedge In1(a2, v) \Rightarrow u = v$
- $In2(m1, u) \wedge In1(m3, v) \Rightarrow u = v$

Représentation de la connaissance : observations

Exemple

Dans cet exemple, seules les entrées et les sorties du circuit sont **observables**.

- $In1(m1,3)$: "L'entrée 1 du multiplieur 1 est 3"
- $In2(m1,2)$
- $In1(m2,2)$
- $In2(m2,3)$
- $In1(m3,2)$
- $In2(m3,3)$
- $Out(a1,10)$
- $Out(a2,12)$

Représentation du diagnostic

Définition

Un état du système $DS, COMPS$ est une phrase logique Φ_Δ telle que $\Delta \subseteq COMP$ et :

$$\bigwedge_{c \in \Delta} Ab(c) \wedge \bigwedge_{c \notin \Delta} \neg Ab(c)$$

Tout composant de Δ est déclaré **anormal**, **fautif**, **en panne**.

Exemple

- 1 $\Delta = \{a1, m2\}$; $\Phi_\Delta = Ab(a1) \wedge Ab(m2) \wedge \neg Ab(a2) \wedge \neg Ab(m1) \wedge \neg Ab(m3)$
- 2 $\Delta = \emptyset$; $\Phi_\Delta = \neg Ab(a1) \wedge \neg Ab(a2) \wedge \neg Ab(m1) \wedge \neg Ab(m2) \wedge \neg Ab(m3)$ état dans lequel tout est normal
- 3 $\Delta = \{a1, a2, m1, m2, m3\}$; $\Phi_\Delta = Ab(a1) \wedge Ab(a2) \wedge Ab(m1) \wedge Ab(m2) \wedge Ab(m3)$ état dans lequel tout est fautif

Propriétés logiques d'un diagnostic

Définition

Un **diagnostic** du système $SD, COMP$ est un état Φ_{Δ} tel que :

SD, OBS, Φ_{Δ} est satisfiable

L'état Φ_{Δ} est **possible** selon DS, OBS (cohérence avec le modèle et les observations).

Définition

Un diagnostic existe ssi :

DS, OBS est satisfiable

Sinon, le modèle n'est pas correct or il est incomplet

Détection de fautes

Définition

Comportement normal du système :

$$DS, \Phi_\theta$$

avec $\Phi_\theta = \bigwedge_{c \in COMP} \neg Ab(c)$.

Définition

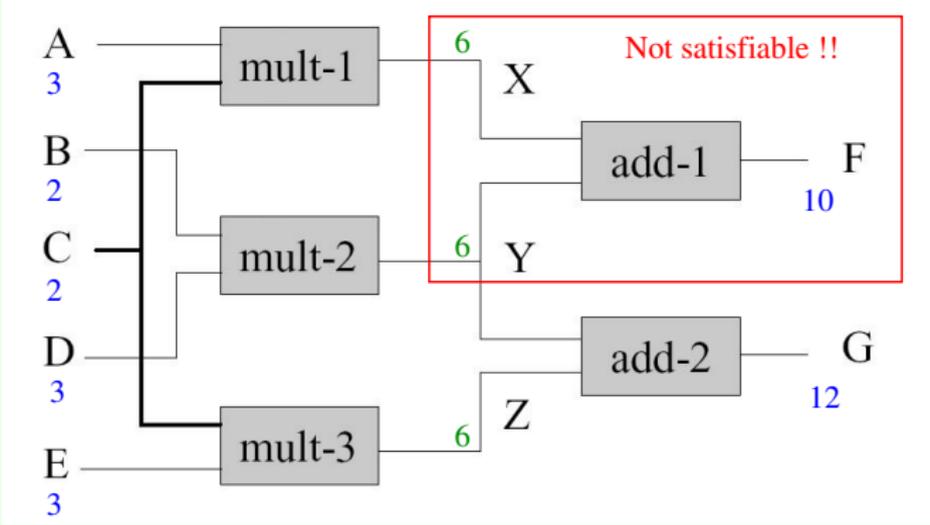
Comment détecter que quelque chose ne va pas ?

On vérifie la satisfiabilité de :

$$SD \wedge \Phi_\theta \wedge OBS$$

Exemple

Exemple



Identification/localisation de fautes

- Détection = recherche d'incohérence
- Identification : quels sont les composants fautifs ?
- Identification : on doit retrouver la cohérence !!!
- Principe : trouver au moins un état Φ_{Δ} tel que :

$$SD \wedge \Phi_{\emptyset} \wedge OBS$$

soit satisfiable

- Nombre d'états possibles $2^{|COMPS|}$, recherche par énumération impossible.

À la recherche de conflits

- Trouver un moyen plus efficace de déterminer les Δ tel que l'on récupère la cohérence de

$$DS \wedge \Phi_{\Delta} \wedge OBS$$

- Rechercher des **conflits** entre composants

Un constat :

$$DS \wedge \Phi_{\emptyset} \wedge OBS \text{ non satisfiable}$$

équivalent à

$$DS \wedge \bigwedge_{c \in COMPS} \neg Ab(c) \wedge OBS \text{ non satisfiable}$$

équivalent à

$$DS \wedge OBS \models \bigvee_{c \in COMPS} Ab(c)$$

À la recherche de conflits (2)

Définition

Un **conflit** C de $(DS, COMPS, OBS)$ est une disjonction de clauses Ab tel que :

$$DS \wedge OBS \models C$$

Autrement dit,

- un conflit C est tel que : $DS \wedge OBS \wedge \neg C$ n'est pas satisfiable
- un conflit C est une hypothèse sur l'état normal/anormal d'un groupe de composants **impossible**

À la recherche de conflits (3)

- Clause de $Ab(c)$: soit $Ab(c)$, soit $\neg Ab(c)$
- Forme d'un conflit C : $Ab(c_1) \vee \neg Ab(c_2) \vee Ab(c_3)$
- Si C est un conflit alors :

$$DS \wedge OBS \models C$$

$$DS \wedge OBS \models Ab(c_1) \vee \neg Ab(c_2) \vee Ab(c_3)$$

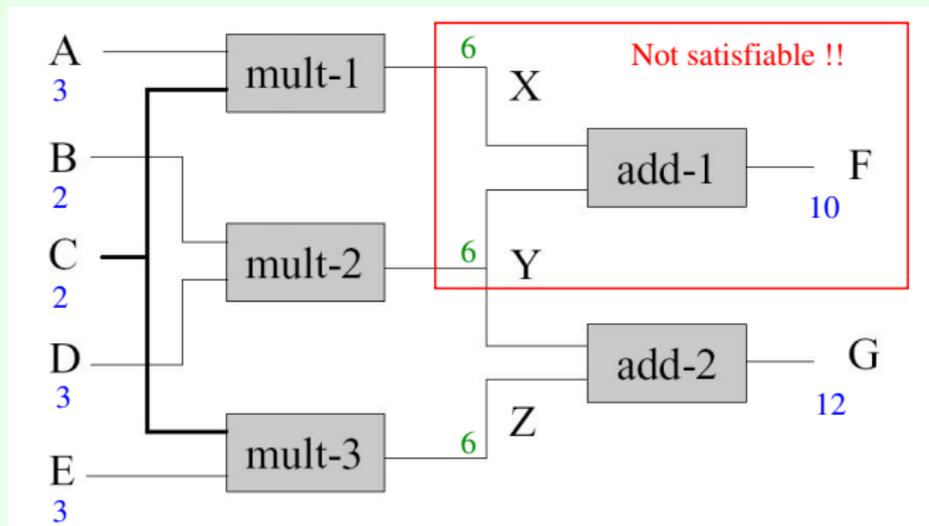
- Sachant $\neg C \equiv \neg Ab(c_1) \wedge Ab(c_2) \wedge \neg Ab(c_3)$
- C exprime :
 - ▶ selon DS et OBS , il n'est pas possible que, DANS LA MEME HYPOTHESE, c_1 et c_3 soient normaux et que c_2 soit fautif

Conflict : exemple

Exemple

$C1 : DS \wedge OBS \models Ab(a_1) \vee Ab(m_1) \vee Ab(m_2)$

$C2 : DS \wedge OBS \models \neg Ab(m_2) \vee Ab(m_3) \vee Ab(a_2)$



Type de conflits

- Conflit C minimal : toute sous-partie de C n'est pas un conflit

$$C_1 : Ab(c_1) \vee \neg Ab(c_2) \vee Ab(c_3) \text{ minimal}$$

$$C_2 : Ab(c_1) \vee \neg Ab(c_2) \text{ pas un conflit}$$

- Un conflit non minimal impose une contrainte plus restrictive
- Conflit C positif : toute clause de C est positive

$$C_3 : Ab(c_1) \vee Ab(c_2) \vee Ab(c_3) \text{ positif}$$

- Conflit positif \Rightarrow au moins l'un des composants est fautif

Des conflits aux diagnostics

Théorème

Soit Π l'ensemble des conflits minimaux de $(DS, COMPS, OBS)$, Δ est un diagnostic ssi $\Pi \wedge \Phi_{\Delta}$ est satisfiable.

- Π ensemble maximal de contraintes imposées par DS, OBS
- $DS \wedge OBS \models \Pi$
- donc on doit avoir : $DS \wedge OBS \wedge \Phi_{\Delta} \models \Pi \wedge \Phi_{\Delta}$
- Si $\Pi \wedge \Phi_{\Delta}$ n'est pas satisfiable alors $DS \wedge OBS \wedge \Phi_{\Delta}$ non-plus
- CQFD

Des conflits minimaux aux diagnostics minimaux

Définition

Un diagnostic Δ est minimal ssi $\forall \Delta' \subset \Delta$, Δ' n'est pas un diagnostic.

Théorème

Si Δ est minimal alors $DS \wedge OBS \wedge \bigwedge_{c \in COMPS \setminus \Delta} \neg Ab(c) \models \bigwedge_{c \in \Delta} Ab(c)$.

Théorème

Soit Π^+ l'ensemble des conflits minimaux positifs de $(DS, COMPS, OBS)$, tout Δ minimal tel que $\Pi^+ \wedge \Phi_\Delta$ est satisfiable est un diagnostic minimal.

On peut donc se restreindre à n'énumérer que les conflits minimaux positifs.

Diagnostics minimaux : exemple

Exemple

$C1 : DS \wedge OBS \models Ab(a_1) \vee Ab(m_1) \vee Ab(m_2)$ conflit positif minimal

$\Pi^+ = (Ab(a_1) \vee Ab(m_1) \vee Ab(m_2))$

Diagnostics minimaux possibles : $\{a_1\}, \{m_1\}, \{m_2\}$

Des diagnostics minimaux à tous les diagnostics

Exemple

① $Ab(a1) \wedge \neg Ab(a2) \wedge Ab(m1) \wedge \neg Ab(m2) \wedge \neg Ab(m3)$

- ▶ only a1 and m1 are faulty

② $Ab(a1) \wedge \neg Ab(a2) \wedge \neg Ab(m1) \wedge \neg Ab(m2) \wedge \neg Ab(m3)$

- ▶ only a1 is faulty

③ $Ab(a1) \wedge Ab(a2) \wedge Ab(m1) \wedge \neg Ab(m2) \wedge \neg Ab(m3)$

- ▶ a1, a2, and m1 are faulty

④ $Ab(a1) \wedge Ab(a2) \wedge Ab(m1) \wedge Ab(m2) \wedge Ab(m3)$

- ▶ everything can be faulty !!!

⑤ ...

Ce n'est pas un diagnostic :

① $\neg Ab(a1) \wedge \neg Ab(a2) \wedge \neg Ab(m1) \wedge \neg Ab(m2) \wedge Ab(m3)$

- ▶ if m3 is faulty there must another faulty component (m2 at least)

Algorithme de recherche de Reiter : DIAGNOSE

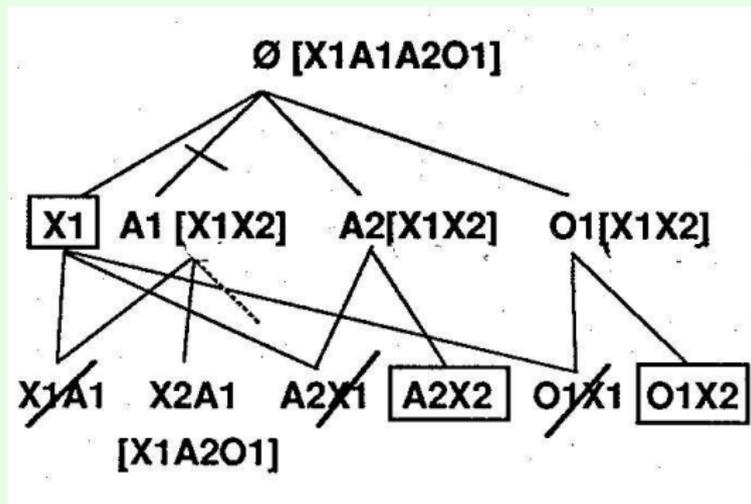
Algorithme

Recherche en largeur d'abord à partir de l'ensemble de composants $\Delta = \emptyset$ sur le treillis des hypothèses. $C(\Delta) \equiv \bigvee_{c \in COMPS \setminus \Delta} Ab(c)$

- 1 Soit Δ $C(\Delta)$ l'ensemble courant
- 2 Test de satisfiabilité (solveur SAT)
Est-ce que $C(\Delta)$ est un conflit positif ? $DS \wedge OBS \wedge \neg C(\Delta)$ non satisfiable
- 3 Si **oui**, éliminer de la recherche tout Δ' tel que $\Delta' \cap (COMPS \setminus \Delta) = \emptyset$ et poursuivre la recherche
 - ▶ Δ' ne peut pas être un diagnostic puisque $C(\Delta)$ est un conflit positif, au moins un composant de $C(\Delta)$ est suspect
- 4 Si **non**, Δ est un diagnostic minimal

DIAGNOSE algorithm : example

Example



Sets in brackets are R-conflicts.

Three minimal diagnoses : $\{X1\}$; $\{X2, O1\}$; $\{X2, A2\}$

Une autre façon de voir le problème

Propriété

L'intersection entre un diagnostic Δ et tout ensemble de composants en conflit n'est jamais vide.

Theorem

$\Delta \subseteq \text{COMPS}$ est un diagnostic minimal ssi Δ un a *ensemble couvrant minimal* pour l'ensemble des conflits positifs minimaux ($SD, COMP, OBS$)

General diagnosis engine (GDE) from de Kleer.

Ensemble couvrant minimal

Définition

Soit $\mathcal{S} = \{S_1, \dots, S_n\}$ un ensemble d'ensembles, H couvre \mathcal{S} ssi

$$H \subseteq \mathcal{S}$$

and

$$\forall S_i \in \mathcal{S}, H \cap S_i \neq \emptyset$$

Exemple

$\mathcal{S} = \{\{a, b\}, \{c, b\}, \{e, f\}\}$ Quelques ensembles couvrants de \mathcal{S} :

- $H = \{a, b, c, e\}$
- $H = \{a, c, f\}$ (H is minimal)

Ensemble non couvrant :

- $H = \{a, b\}$

Algorithme GDE

Algorithme

- 1 Calcul de tous les conflits minimaux positifs
 - ▶ Utilisation d'un ATMS (Assumption Truth Maintenance System)
 - ▶ Mise à jour des croyances par retractation des croyances et ajout de nouvelles croyances
- 2 Calcul des ensembles couvrants minimaux : ensemble de diagnostics minimaux.

Modèle de faute

Définition

Comportement fautif : connaissance sur le comportement du système en cas de faute.

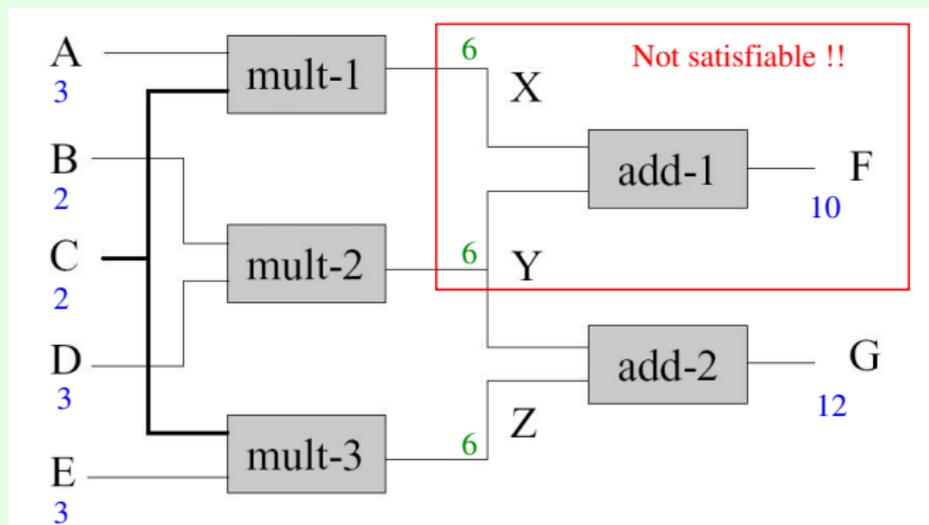
Exemple

- “Quand c’est fautif un additionneur renvoie toujours 0”
- $Ab(a2) \Rightarrow Out(a2, 0)$
- “Les additionneurs en panne se comportent comme des soustracteurs”
- $Add(x) \wedge Ab(x) \wedge In1(x, u) \wedge In2(x, v) \wedge Subtract(u, v, w) \Rightarrow Out(x, w)$

Identification de faute : exemple 2

Example

$DS, \{Ab(a2) \Rightarrow Out(a2,0)\}, OBS$



Explication et diagnostic abductif

Définition

Un diagnostic Φ_{Δ} de $(DS, COMP, OBS)$ est une **explication** pour une observation donnée $o \in OBS$ ssi

$$SD, \Phi_{\Delta} \models o$$

Définition

Un diagnostic abductif Δ est un diagnostic qui explique toutes les observations. $\forall o \in OBS$

$$SD, \Phi_{\Delta} \models o$$

Théorème

Tout diagnostic Δ a une partie explicative ($OBS = OBS_{SAT} \wedge OBS_{EXP}$).

$$SD \wedge \Phi_{\Delta} \wedge OBS_{SAT} \models OBS_{EXP}.$$

Explanation : example

Example

Example 1 :

All the diagnoses that cover the following sentence (which is not a partial diagnosis) are explanations of $Out(a2, 12)$

$$\neg Ab(m2) \wedge \neg Ab(m3) \wedge \neg Ab(a2)$$

for instance :

$$Ab(a1) \wedge \neg Ab(a2) \wedge \neg Ab(m1) \wedge \neg Ab(m2) \wedge \neg Ab(m3)$$