Diagnostic

Boi Faltings

Laboratoire d'Intelligence Artificielle
boi.faltings@epfl.ch
http://moodle.epfl.ch/

Tâches Abductives

il y a 2 types de tâches abductives qui trouvent beaucoup d'applications pratiques:

- le *diagnostic*, où il faut trouver les composants d'un système qui sont responsables d'un dysfonctionnement.
- la *planification*, où il faut trouver une séquence d'actions qui atteint un certain ensemble de buts.

$\Rightarrow$ méthodologies spécifiques.

Le Problème du Diagnostic

- Motivation: corriger un dysfonctionnement dans un dispositif/système technique ou biologique.
- Entrées: observations du comportement, en particulier des *symptômes* du dysfonctionnement.
- Entrées supplementaires: mesures en cours de route.
- Sorties: actions à entreprendre pour corriger le défaut: (composants à remplacer ou à réparer).

Difficulté: chaque dispositif est différent $\Rightarrow$ il faut un raisonnement qui en tient compte.

Formalisation

Etant données:

- un modèle du système (MS)
- un ensemble d'observations (OBS)

trouver:

- des candidats (CAND) de diagnostic

Candidat = combinaison de composantes défectueuses

Quels sont les candidats valables?

Comportement défaillant:

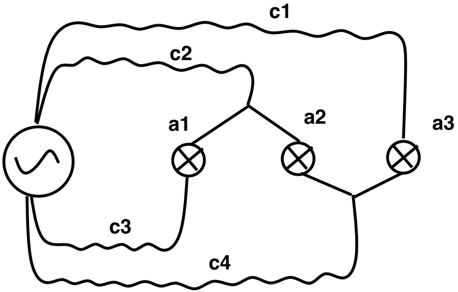
$$MS \cup OBS \vdash \bot (\text{Contradiction})$$

2 Versions:

- le candidat *explique* les observations:
  $MS \cup CAND \vdash OBS$
- le candidat rend les observations consistantes:
  $(MS - CAND) \cup OBS \not\vdash \bot$

Exemple

Considérons le circuit électrique:



Observations: fonctionnement des trois ampoules a1 - a3
A trouver: câbles défectueux parmi c1 - c4
Hypothèse: seuls les câbles peuvent tomber en panne.

Règles "Experts"

`éteint(a1)` $\Rightarrow$ `déf(c2)` $\vee$ `déf(c3)`

...

Pas de clauses de Horn!

Meilleure approche: abduction d'une simulation (modèle)

`déf(c2)` $\Rightarrow$ `éteint(a1)`

`déf(c3)` $\Rightarrow$ `éteint(a1)`

...

Modélisation

Modèle = composants + lois de comportement (règles/contraintes)

Choix du niveau:

- aussi abstrait que possible pour limiter la complexité des candidats
- suffisement précis pour pouvoir identifier l'unité à remplacer

En général, modélisation de chaque composant échangeable.

Modélisation logique du système (MS)

Circuit:

$\texttt{ampoule}(a_1), \texttt{ampoule}(a_2), \texttt{ampoule}(a_3)$
$\texttt{cable}(c_1), \texttt{cable}(c_2), \texttt{cable}(c_3), \texttt{cable}(c_4)$
$\texttt{connexion}(c_1, \texttt{src}, a_3), \texttt{connexion}(c_2, \texttt{src}, a_1)$
$\texttt{connexion}(c_2, \texttt{src}, a_2), \texttt{connexion}(c_3, a_1, \texttt{src})$
$\texttt{connexion}(c_4, a_2, \texttt{src}), \texttt{connexion}(c_4, a_3, \texttt{src})$

Règle:

$\texttt{ampoule(x)} \wedge \texttt{cable(y)} \wedge \texttt{connexion(y,src,x)} \wedge \texttt{cable(z)}$
$\wedge\ \texttt{connexion(z,x,src)} \Rightarrow \texttt{allumée(x)}$

Formalisme valable pour *n'importe quel* circuit!
Permet de simuler le circuit $MS \Rightarrow OBS$

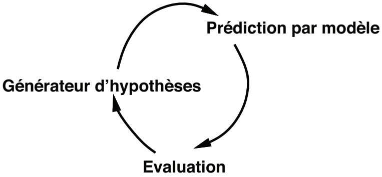
Comment implementer l'abduction?

Le diagnostic est un problème abductif.
3 techniques principales:

1. abduction explicite
2. transformation en deduction
3. raisonnement incertain (probabilités, logique floue)

Abduction explicite

Mécanisme général:



Hypothèse d'un monde clos:

- le circuit ne change pas
- on considère toutes les hypothèses possibles

appliquée au moment de l'exécution de la recherche.

Abduction par recherche

Génération des hypothèses par un algorithme de recherche, p. ex.:

- noeud de recherche = ensemble de défauts
- noeud initial = ensemble vide
- fonction de successeur = ajouter un défaut
- terminaison quand les défauts permettent de déduire les observations

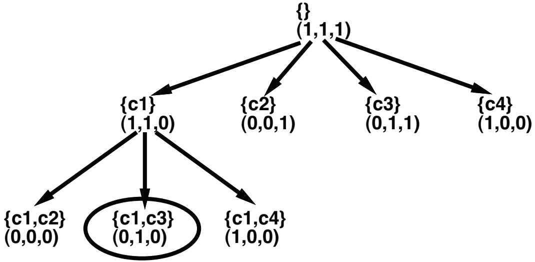
Complexité peut être élevée!

Exemple

Supposons l'observation:

`éteint(a₁), allumé(a₂), éteint(a₃)`

alors on effectue la recherche:



Exploration en largeur d'abord: on trouve d'abord le diagnostic le plus simple.

Variante: utiliser A^*, mais pas toujours facile de choisir une fonction heuristique!

Transformation en déduction

Appliquer l'hypothèse d'un monde clos:

- le circuit ne change pas
- on ne découvre pas de nouvelles possibilités de fautes

$\Rightarrow$ 2 étapes:

1. simuler le comportement pour tous les défauts possibles (recherche exhaustive)
2. construire des règles pour la déduction du diagnostic à partir des observations.

Inférences du modèle

c1	c2	c3	c4	ampoules éteints
1	1	1	1	{}
1	1	1	0	a_2, a_3
1	1	0	1	a_1
1	1	0	0	a_1, a_2, a_3
1	0	1	1	a_1, a_2
1	0	1	0	a_1, a_2, a_3
1	0	0	1	a_1, a_2
1	0	0	0	a_1, a_2, a_3
0	1	1	1	a_3
0	1	1	0	a_2, a_3
0	1	0	1	a_1, a_3
0	1	0	0	a_1, a_2, a_3
0	0	1	1	a_1, a_2, a_3
0	0	1	0	a_1, a_2, a_3
0	0	0	1	a_1, a_2, a_3
0	0	0	0	a_1, a_2, a_3

Inférences abductives:

Hypothèse du monde clos: le modèle est complet

$\Rightarrow$ inférences:

a_1	a_2	a_3	Diagnostic
1	1	1	$\{\}$
1	1	0	c_1
1	0	0	$c_4 \vee (c_1 \wedge c_4)$
1	0	1	-
0	0	1	c_2
0	1	1	c_3
0	1	0	$c_3 \wedge c_1$
0	0	0	$(c_1 \wedge c_2) \vee (c_3 \wedge c_4) \vee (c_2 \wedge c_4)$

Notez: le cas où toutes les ampoules sont éteintes ne peut pas être traduit en clauses de Horn.

Abduction $\Rightarrow$ Déduction

Connaissances + modèle + monde clos

$\Rightarrow$ règles déductives:

1. $\texttt{éteint}(a_3) \wedge \neg\ \texttt{éteint}(a_2) \Rightarrow \texttt{défectueux}(c_1)$
2. $\texttt{éteint}(a_1) \wedge \texttt{éteint}(a_2) \wedge \neg\ \texttt{éteint}(a_3)$
   $\Rightarrow \texttt{défectueux}(c_2)$
3. $\texttt{éteint}(a_1) \wedge \neg\ \texttt{éteint}(a_2) \Rightarrow \texttt{défectueux}(c_3)$
4. $\texttt{éteint}(a_2) \wedge \texttt{éteint}(a_3) \wedge \neg\ \texttt{éteint}(a_1)$
   $\Rightarrow \texttt{défectueux}(c_4)$
5. $\texttt{éteint}(a_1) \wedge \texttt{éteint}(a_2) \wedge \texttt{éteint}(a_3)$
   $\Rightarrow (\texttt{défectueux}(c_1) \wedge \texttt{défectueux}(c_2)) \vee \ldots$

$\Rightarrow$ système expert ou programme conventionel

Problèmes du diagnostic déductif

- Pas tous les cas donnent lieu à des clauses de Horn.
- L'hypothèse d'un monde clos est appliquée au moment de la construction des règles
  $\Rightarrow$ logiciel valable que pour un seul circuit
- Tout changement du circuit requiert une reprogrammation globale: très coûteux!
- Impossible de détecter si les connaissances sont applicables
- Impossible de traîter des situations non prévues par le programmeur

Diagnostic par raisonnement incertain

Idée: traduire l'ambiguité des raisons d'un dysfonctionnement en incertitudes sur le diagnostic.

Règles déductives, mais générales $\Rightarrow$ évite:

- l'hypothèse du monde clos
- système spécialisé

Abduction par incertitude (exemple)

1. $\texttt{ampoule(x)} \wedge \texttt{cable(y)} \wedge \texttt{connexion(y,src,x)} \wedge \texttt{éteint(x)} \overset{CF=0.5}{\Rightarrow} \texttt{défectueux(y)}$
2. $\texttt{ampoule(x)} \wedge \texttt{cable(y)} \wedge \texttt{connexion(y,x,src)} \wedge \texttt{éteint(x)} \overset{CF=0.5}{\Rightarrow} \texttt{défectueux(y)}$
3. $\texttt{ampoule(x)} \wedge \texttt{cable(y)} \wedge \texttt{connexion(y,src,x)} \wedge \neg\texttt{éteint(x)} \overset{CF=-1.0}{\Rightarrow} \texttt{défectueux(y)}$
4. $\texttt{ampoule(x)} \wedge \texttt{cable(y)} \wedge \texttt{connexion(y,x,src)} \wedge \neg\texttt{éteint(x)} \overset{CF=-1.0}{\Rightarrow} \texttt{défectueux(y)}$

Exemple...

$\texttt{éteint}(a_3) \Rightarrow$:

$\texttt{défectueux}(c_1)$, CF=0.5

$\texttt{défectueux}(c_4)$, CF=0.5

$\texttt{éteint}(a_1) \wedge \texttt{éteint}(a_2) \Rightarrow$:

$\texttt{défectueux}(c_2)$, CF=0.75

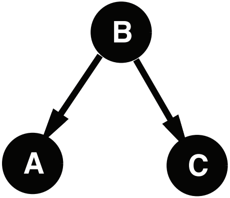
(2 règles, combinaison parallèle)

$\texttt{défectueux}(c_3)$, CF=0.5

$\texttt{défectueux}(c_4)$, CF=0.5

Abduction par Réseaux Bayesiens

Une défaillance B se manifeste par multiples effets observables, ici A et C:



$$
\begin{aligned}
p(B|A,C) &= \frac{1}{p(A,C)} p(A,B,C) \\
&= \alpha p(A|B,C) \cdot p(B,C) \\
&= \alpha p(A|B) \cdot p(C|B) \cdot p(B)
\end{aligned}
$$

(car A et C indépendants étant donné B)

En général, pour k conséquences $Y_1, .., Y_k$:

$$
p(B|Y_1, .., Y_k) = \alpha p(B) \prod_{i=1}^{k} p(Y_i|B)
$$

"Naive Bayes"

- Pour un diagnostic, on compare les probabilités de différents candidats c pour retenir le plus probable.

$\Rightarrow$ pour chaque observation $o \in OBS$, spécifier $p(o|c)$ (modélisation du comportement).

- Trier les candidats par

$$\alpha p(c) \prod_{o \in OBS} p(o|c)$$

α est toujours identique $\Rightarrow$

$$p(c) \prod_{o \in OBS} p(o|c)$$

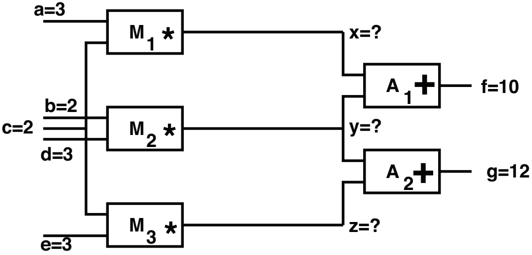
est suffisant comme critère.

Limites...

- Pas clair comment choisir les facteurs de certitude.
- Aucune garantie de trouver le bon diagnostic.
  $\Rightarrow$ nécessite de l'expérimentation.
- Raisonnement probabiliste est bien pour des structures simples, mais l'inférence peut devenir complexe selon les rélations causales.
- La plupart des systèmes experts pour le diagnostic utilisent un raisonnement incertain.

Difficultés de modélisation

- Diagnostic par abduction: le diagnostic doit *prédire* les observations.
- Comportement défectueux peut être difficile à modéliser
- Exemple: un circuit digital pour calculer F=AC+BD, G=BD+CE:



⇒ nombreuses possibilités de défaillance.

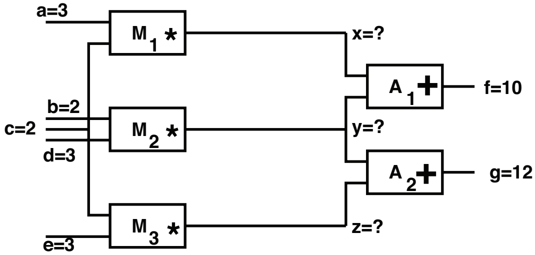
Diagnostic basé sur la consistance

- Principe: diagnostic $CAND$ doit rendre les observations consistantes:

$$(MS - CAND) \cup OBS \not\vdash \perp$$

- $CAND$ = ensemble de composants défectueux.
- On enlève du modèle toutes les prévisions qui dépendent de $CAND$.

Prévisions



Règles de comportement ⇒ prévision de mesures:

$$\begin{aligned} a = 3, c = 2 &\vdash x = 6(M_1) \\ b = 2, d = 3 &\vdash y = 6(M_2) \\ x = 6, y = 6 &\vdash f = 12(A_1, M_1, M_2) \end{aligned}$$

Justifiées par comportement correct des composantes impliqués: M_1, M_2 et A_1.

Symptômes et conflits

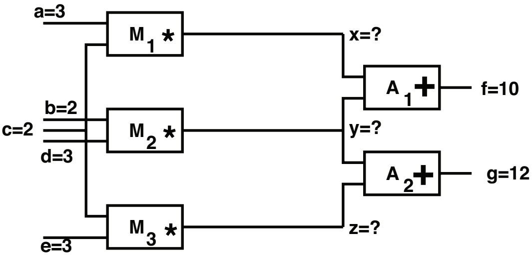
Toute différence entre

- prévision et mesure, ou
- 2 prévisions,

est un symptôme.

Les justifications des prévisions impliquées représentent un *conflit*: au moins un des composants doit être défectueux.

Exemple



Mesure f=10 $\Rightarrow$ contradiction avec la prévision f=12.
$\Rightarrow$ conflit:

$$\{M_1, M_2, A_1\}$$

Conflits $\Rightarrow$ Candidats

- Candidat = Ensemble de composants défectueux.
- Conflit $\{M_1, M_2, A_1\} \Rightarrow$ candidats:

$$\{M_1\}, \{M_2\}, \{A_1\},$$
$$\{M_1, M_2\}, \{M_1, A_1\}, \{M_2, A_1\}$$
$$\{M_1, M_2, A_1\}$$

Candidats minimaux

- Composants ne tombent pas en panne simultanément.
- Candidats minimaux: aucun sous-ensemble n'est déjà candidat:

$$D = \{\{M_1\}, \{M_2\}, \{A_1\}\}$$

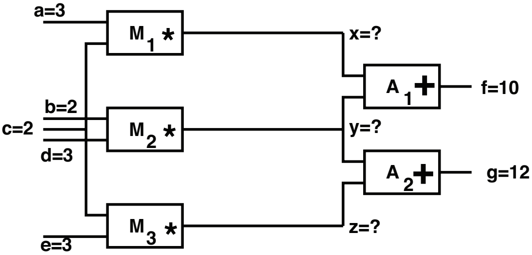
$\Rightarrow$ reduit la complexité.

Intégration d'autres conflits

On découvre les conflits en prenant des mesures.
Intégration dans les candidats en 2 étapes:

- pour chaque $CAND$ et composant $c \in$ conflit, générer $C' = CAND \cup \{c\}$.
- filtrer pour éliminer ceux qui ne sont pas minimaux.

Exemple



Supposon qu'on mesure aussi g=12, alors:

$$
\begin{aligned}
c = 2, e = 3 &\vdash \quad z = 6(M_3) \\
z = 6, g = 12 &\vdash \quad y = 6(M_3, A_2) \\
y = 6, f = 10 &\vdash \quad x = 4(M_3, A_2, A_1) \\
a = 3, c = 2 &\vdash \quad x = 6(M_1)
\end{aligned}
$$

$\Rightarrow$ nouveau conflit: $\{A_1, A_2, M_1, M_3\}$

Exemple (2)

$$D_1 = \{\{M_1\}, \{M_2\}, \{A_1\}\}$$

Intégration du conflit ($\{A_1, A_2, M_1, M_3\}$):

$$\begin{aligned} D' = \quad & \{M_1, A_1\}, \{M_2, A_1\}, \{A_1\} \\ & \{M_1, A_2\}, \{M_2, A_2\}, \{A_1, A_2\} \\ & \{M_1\}, \{M_2, M_1\}, \{A_1, M_1\} \\ & \{M_1, M_3\}, \{M_2, M_3\}, \{A_1, M_3\} \end{aligned}$$

$\Rightarrow$ candidats minimaux:

$$D_2 = \{\{M_1\}, \{A_1\}, \{M_2, M_3\}, \{A_2, M_2\}\}$$

Progrès du diagnostic

- Chaque symptôme tend à augmenter le nombre de candidats minimaux.
  $\Rightarrow$ problème de convergence vers une seule solution.
- Cependant, le nombre de candidats à petit nombre de composants diminue rapidement.
  $\Rightarrow$ de moins en moins de candidats à probabilité elevée.
  $\Rightarrow$ convergence du diagnostic.

Evaluer les candidats

- A chaque instance, le système maintient plusieurs candidats de diagnostic possibles.
- On peut attribuer des préférences sur la base de probabilités. Les probabilités de défaillance sont connues pour la plupart des composants d'un dispositif.
- Sous l'hypothèse de l'indépendance des défaillances, on peut calculer la probabilité d'un candidat

$$C = \{\ D = \textit{composants défectueux}\ \} \cup \{\ N = \textit{composants non-défectueux}\ \}$$

comme

$$P(C) = \prod_{c \in D} P(c) \cdot \prod_{c \in N} (1 - P(c))$$

Proposition de Mesures

- Comment proposer les prochaines mesures à prendre de manière à faire progresser le diagnostic le plus rapidement possible?
- Probabilités des candidats ⇒ *l'entropie* des mesures, c'est-à-dire l'incertitude quant à la valeur mesurée:

$$E(var_i) = -\sum_k p(var_i = val_{ik}) * log(p(var_i = val_{ik}))$$

- On sélectionne alors la mesure dont l'entropie est la plus élevée.

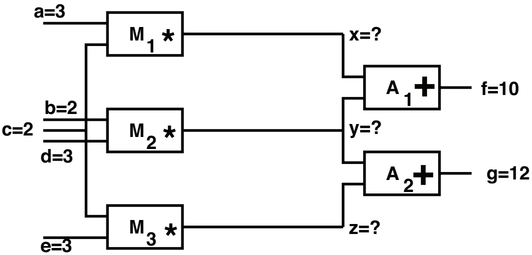
Probabilité des Valeurs

Probabilité de mesurer la valeur k pour la variable i:

$$p(var_i = val_{ik}) = \sum_{c \in P_{ik}} p(c) + \sum_{c \in U_i} p(c)/m$$

- P_{ik} = l'ensemble de candidats qui prédisent la valeur k pour la variable i.
- U_i regroupe les candidats qui ne donnent aucune valeur pour la mesure i et leur donne une probabilité de $1/m$, m = nombre de valeurs possibles pour var_i.

Exemple (rappel)



$$X = 4(M_2, A_1), (M_3, A_1, A_2) \quad X = 6(M_1)$$
$$Y = 6(M_2), (M_3, A_2) \quad Y = 4(M_1, A_1)$$
$$Z = 6(M_3), (M_2, A_2) \quad Z = 8(M_1, A_1, A_2)$$

Probabilités des Candidats

Supposons que probabilité de panne d'une composant $= 0.01$

$$\begin{aligned} p(\{M_1\}) &= 0.01 \Rightarrow 0.495 \\ p(\{A_1\}) &= 0.01 \Rightarrow 0.495 \\ p(\{M_2, A_2\}) &= 0.0001 \Rightarrow 0.005 \\ p(\{M_2, M_3\}) &= 0.0001 \Rightarrow 0.005 \end{aligned}$$

Attention: il faut normaliser les probabilités pour que

$$\sum_{CAND} p(CAND) = 1$$

comme on sait qu'un des candidats doit être correct.

Proposition de Mesures (Exemple)

Prévisions et probabilités des candidats:

$$\begin{array}{lll}
\{M_1\} & xyz = (4,6,6) & 0.495 \\
\{A_1\} & xyz = (6,6,6) & 0.495 \\
\{M_2, A_2\} & xyz = (6,4,6) & 0.005 \\
\{M_2, M_3\} & xyz = (6,4,8) & 0.005
\end{array}$$

$\Rightarrow$ entropies:

$$\begin{array}{rcl}
X : -P(6)log(P(6)) - P(4)log(P(4)) & = & \\
-0.505log(0.505) - 0.495log(0.495) & = & 0.99993bit \\
Y : -P(6)log(P(6)) - P(4)log(P(4)) & = & 0.0808bit \\
Z : -P(6)log(P(6)) - P(8)log(P(8)) & = & 0.0454bit
\end{array}$$

$\Rightarrow$ mesurer X

Application: Diagnostic de systèmes spatiaux

- Sondes spatiales difficiles à contrôler depuis la terre.
- Beaucoup d'imprévus: échouent souvent à cause de petits défauts techniques.
- Deep Space 1 (1999) était la première sonde avec un système de diagnostic autonome: LIVINGSTONE.
- Le système a trouvé et corrigé plusieurs défauts, y compris une panne d'un des moteurs.
- La sonde aurait été inopérationelle avant son but sans le système.

Résumé

- Caractérisation du Diagnostic
- Diagnostic abductif:
  - abduction explicite
  - transformation en déduction
  - raisonnement incertain
- Diagnostic basé sur la consistance
- Proposition de mesures