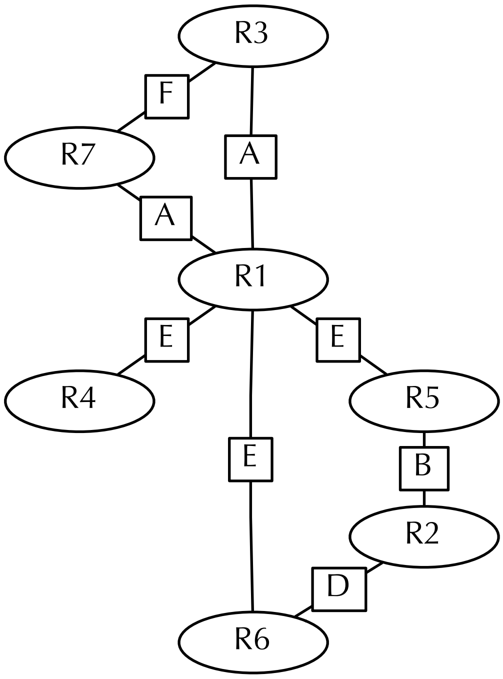
Intelligence artificielle — TD

Système expert

BC (Base de Connaissance) + BR (Base de Règle)
Chaînage arrière : But
Chaînage avant : Saturation (+ But)



Exercice p.3 — Chaînage avant

R1	A → E
R2	B → D
R3	H → A ∧ F
R4	E ∧ G → C
R5	E ∧ K → B
R6	D ∧ E ∧ K → C
R7	G ∧ K ∧ F → A

Ordre des règles : R3 - R7 - R1 - R4, R5 - R2 - R6
Faits non déductibles : H, G, K
Base de faits initiale : H, K But : C

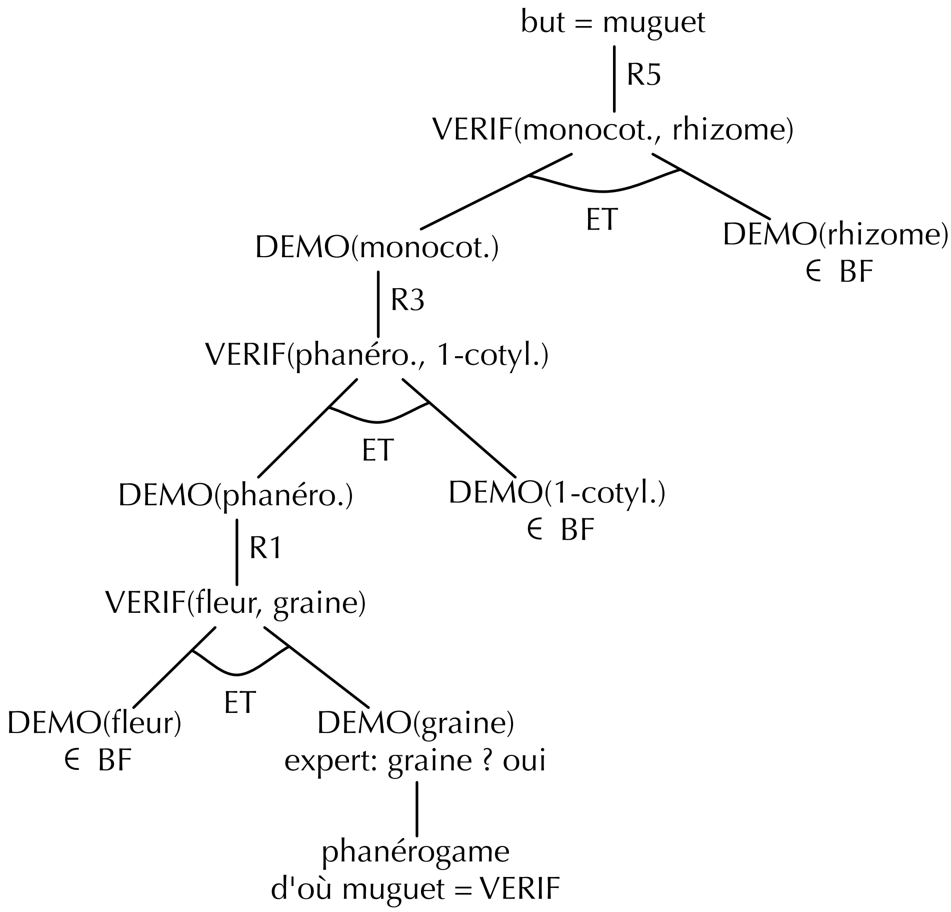
Nb Inférence	BF	BR	Explication
0	H K	toute la base	
1	H K A F	BR — { R3 }	H → A ∧ F
2	H K A F E	BR — { R3, R1 }	A → E
3	H K A F E B	BR — { R3, R1, R5 }	E ∧ K → B
4	H K A F E B D	BR — { R3, R1, R5, R2 }	B → D
5	H K A F E B D C	BR — { R3, R1, R5, R2, R6 }	D ∧ E ∧ K →C

Exercice p.8 — Chaînage arrière

R1	fleur ∧ graine	→ phanérogame ∧ ¬ cryptogame
R2	phanérogame ∧ graine nue	→ sapin
R3	phanérogame ∧ 1-cotylédone	→ monocotylédone
R4	phanérogame ∧ 2-cotylédone	→ dicotylédone
R5	monocotylédone ∧ rhizome	→ muguet
R6	dicotylédone	→ anémone
R7	monocotylédone ∧ ¬ rhizome	→ lilas
R8	feuille ∧ ¬ fleur	→ cryptogame
R9	cryptogame ∧ ¬ racine	→ mousse
R10	cryptogame ∧ racine	→ fougère
R11	¬ feuille ∧ plante	→ thallophyte
R12	thallophyte ∧ chlorophylle	→ algue
R13	thallophyte ∧ ¬ chlorophylle	→ champignon
R14	¬ feuille ∧ ¬ fleur ∧ ¬ plante	→ colibacille

Faits non déductibles :
fleur, graine, graine nue, 1-cotylédone, 2-cotylédone, feuille, racine, plante, chlorophylle

Base de faits initiale :
rhizome, fleur, 1-cotylédone



Exercice 1 — TD numéro 1

a) Chaînage avant

R1 **SI** Responsabilité **ET** Langue-facile **ET** Néerlandais-parlé **ALORS** Dynamique
R2 **SI** Langue-facile **ET** Anglais-parlé **ALORS** Adaptabilité
R3 **SI** Slave **ET** Dynamique **ALORS** Adaptabilité
R4 **SI** Responsabilité **ALORS** Leadership
R5 **SI** Langue-facile **ALORS** Néerlandais-parlé
R6 **SI** Adaptabilité **ET** Leadership **ALORS** Accepté
R7 **SI** Slave **ALORS** Langue-facile
R8 **SI** Leadership **ET** Slave **ALORS** Adaptabilité

Faits non déductibles : Slave, Responsabilité
Base de faits initiale : Slave, Responsabilité

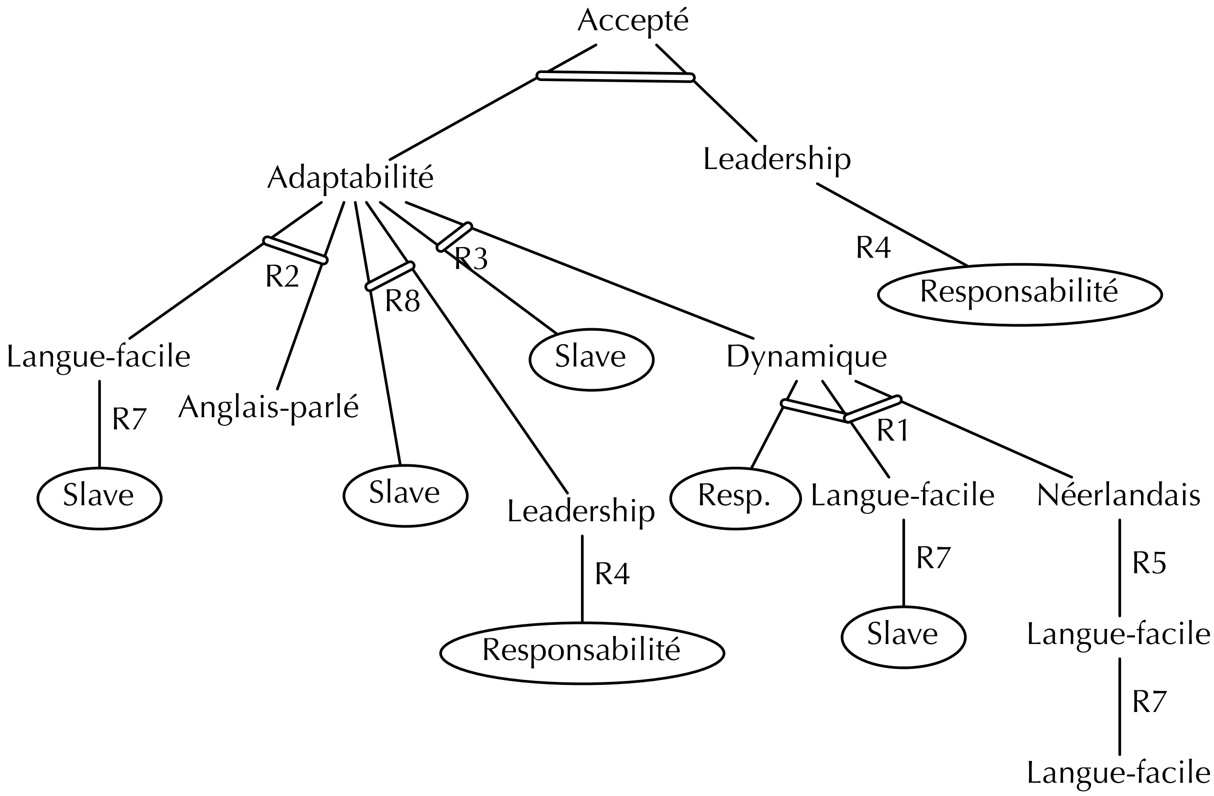
Chaînage avant en largeur

On déclenche toutes les règles possibles à chaque étape : on enrichit la BF au maximum.

Chaînage avant en profondeur

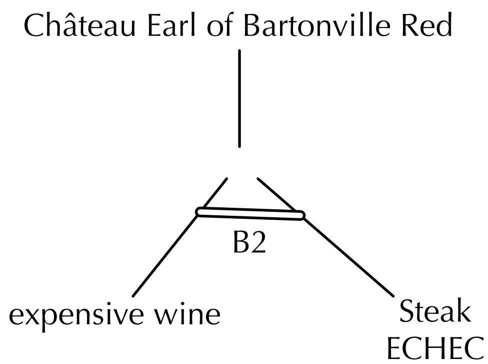
1.	(BF)	Slave, Responsabilité	Propriétés de niveau 0
	(R4, R7)	Leadership, Langue-facile	Propriétés de niveau 1
	(R5, R8)	Néerlandais-parlé, Adaptabilité	Propriétés de niveau 2
	(R1, R6)	Dynamique, Accepté	Propriétés de niveau 3

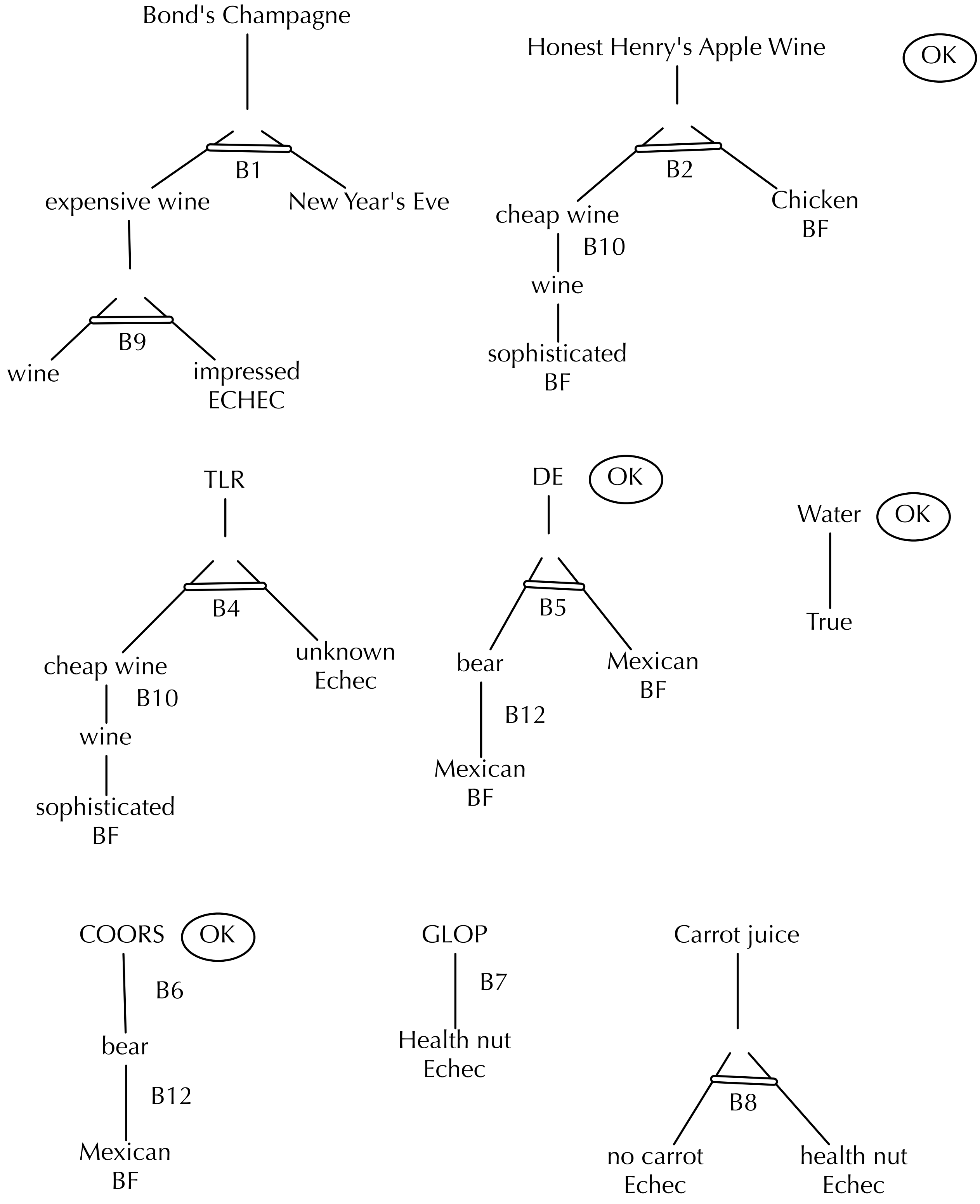
2.	(BF)	Slave, Responsabilité	Propriétés de niveau 0
	(R4)	Leadership	Propriétés de niveau 1
	(R7)	Langue-facile	Propriétés de niveau 2
	(R5)	Néerlandais-parlé	Propriétés de niveau 3
	(R1)	Dynamique	Propriétés de niveau 4
	(R3)	Adaptabilité	Propriétés de niveau 5
	(R6)	Accepté	Propriétés de niveau 6



Exercice 2

Choosen beverage : ?




Bond's Champagne
B1
expensive wine
New Year's Eve
B9
wine
impressed
ECHEC
Honest Henry's Apple Wine
OK
B2
cheap wine
Chicken
BF
B10
wine
sophisticated
BF
TLR
B4
cheap wine
unknown
Echec
B10
wine
sophisticated
BF
DE
OK
B5
bear
Mexican
BF
B12
Mexican
BF
Water
OK
True
COORS
OK
B6
bear
B12
Mexican
BF
GLOP
B7
Health nut
Echec
Carrot juice
B8
no carrot
Echec
health nut
Echec

Exercice 3

S1	*S2*
Table(A)	Table(A)
Sur(B,A)	Libre(A)
Libre(B)	Table(C)
Table(C)	Sur(B,C)
Sur(P,C)	Libre(B)
Libre(P)	Table(P)
	Libre(P)

Constante(T)

Table(A) ⇔ Sur(A,T)

Depiler (X, Y)

Condition	Sur(X,Y)
	Libre(X)
Ajout	Table(X)
	Libre(Y)
Retrait	Sur(X,Y)

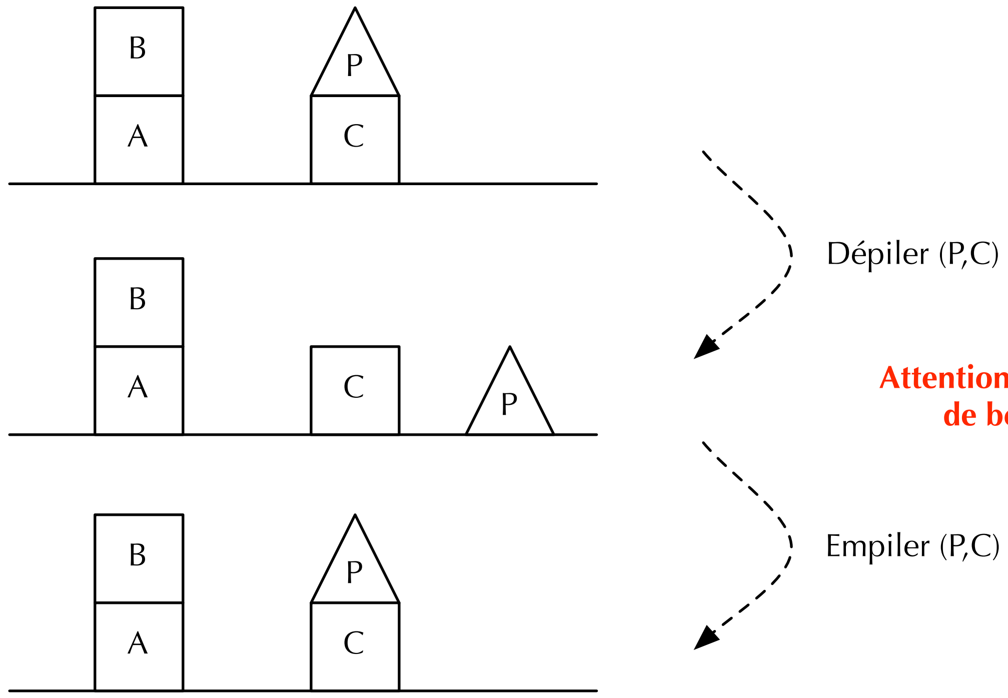
Empiler(X,Y)

Condition	Table(X)
	Libre(Y)
	Libre(X)
	Y <> P
	X <> Y
Ajout	Sur(X,Y)
Retrait	Table(X)
	Libre(Y)

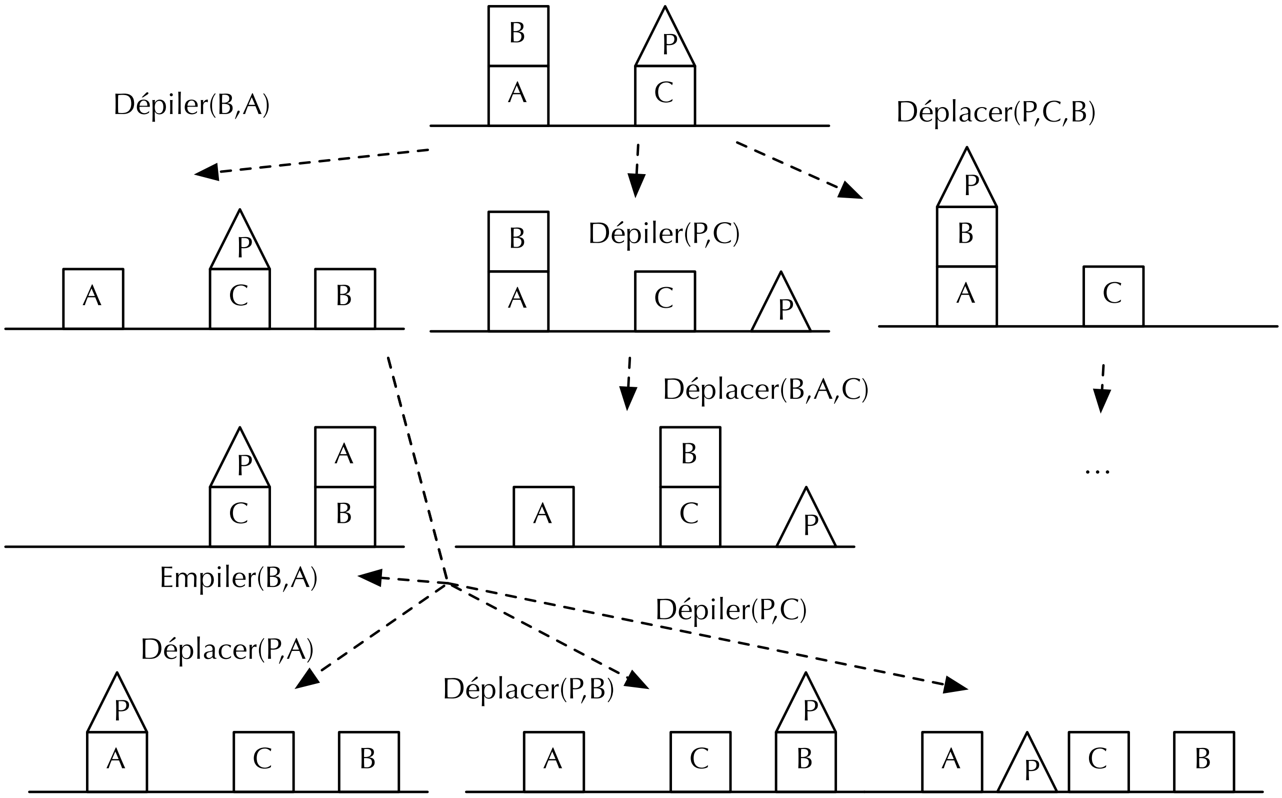
Déplacer(X,Y,Z)

Condition	Sur(X,Y)
	Libre(X)
	Libre(Z)
	Z <> P
	Z <> Y
Ajout	Sur(X,Z)
	Libre(Y)
Retrait	Sur(X,Y)
	Libre(Z)

SI	(Sur(X,Y), Libre(X))
ALORS	Appliquer(Dépiler(X,Y))
	Ajouter(Table(X), Libre(Y))
	Retirer(Sur(X,Y))



Attention aux risques de bouclage



Critère

- Compter le nombre de bloc bien placés sur la table
- Compter les faits identiques entre les deux situations

Détection d'incohérences dans une BR avec COVADIS

Remarque

Ne s'applique qu'à des BR monotones

Définition — cohérence globale d'une BR

Une base de règles est cohérente si et seulement si pour toute base de faits susceptibles d'être fournie, l'ensemble des faits obtenus par saturation satisfaitles spécifications de cohérence.

Contraintes de cohérences

- Monovaluation des attributs
  Att = Val $\in$ BF et vak $\neq$ val' $\Rightarrow$ Att = val' $\notin$ BF

- Ensemble de règles d'incohérence
  eg. Père(X,Y) , féminin(X) $\Rightarrow$ INC

- Contraintes fonctionnelles
  eg. Nb_pique + Nb_coeur + Nb_carreau + Nb_trefle = 13

Contexte d'un littéral

Le contexte d'un littéral déductible est l'ensemble des faits initiaux permettant de le déduire.

Calculer les contextes de tous les littéraux déductibles
(le calcul se fait niveau par niveau, en partant du niveau 0, faits initiaux)

Exemple

R1 : père(X,Y), mère(Y,Z) -> grand_père(X,Z)
R2 : père(X,Y), père(Y,Z) -> grand_père(X,Z)
R3 : mère(X,Y), mère(Y,Z) -> grand_mère(X,Z)
R4 : mère(X,Y), père(Y,Z) -> grand_mère(X,Z)
R5 : père(U,X), grand_père(X,Z) -> bisaïeul(U,Z)
R6 : mère(U,X), grand_mère(X,Z) -> bisaïeul(U,Z)

Initiaux	père, mère
Déductibles	grand_père(L1), grand_mère(L2), bisaïeul(L3)

Spécification des cohérences

c1 : père(X,Y), féminin(X) -> INC
c2 : grand_père(X,Y), féminin(X) -> INC
c3 : bisaïeul(X,Y), féminin(X) -> INC

CTXT(L1) = (père(X,Y) $\wedge$ mère(Y,Z)) $\vee$ (père(X,Y) $\wedge$ père(Y,Z))

c2 : grand_père(X,Y), féminin(X) -> INC

(père(X,Y) $\wedge$ mère(Y,Z)) $\vee$ (père(X,Y) $\wedge$ père(Y,Z)) $\wedge$ féminin(X) -> INC c1 : père(X,Y) ∧ féminin(X) -> INC

c3 : bisaïeul(X,Y) ∧ féminin(X) -> INC

CTXT(L3) = (père(U,X) ∧ grand_père(X,Z)) ∨ (mère(U,X) ∧ grand_mère(X,Z))

CTXT(L2) = (mère(X,Y) ∧ mère(Y,Z)) ∨ (mère(X,Y) ∧ père(Y,Z))

CTXT(L3) =
père(U,X) ∧ père(X,Y) ∧ mère(Y,Z)) ∨
père(U,X) ∧ père(X,Y) ∧ père(Y,Z)) ∨
mère(U,X) ∧ père(X,Y) ∧ mère(Y,Z)) ∨
mère(U,X) ∧ père(X,Y) ∧ père(Y,Z))

c3.1 : père(U,X) ∧ père(X,Y) ∧ mère(Y,Z) ∧ féminin(U) -> INC
c3.2 : père(U,X) ∧ père(X,Y) ∧ père(Y,Z) ∧ féminin(U) -> INC
c3.3 : mère(U,X) ∧ père(X,Y) ∧ mère(Y,Z) ∧ féminin(U) -> INC
c3.4 : mère(U,X) ∧ père(X,Y) ∧ père(Y,Z) ∧ féminin(U) -> INC

-> incohérence dans la Base de Règles

Exercice

R1 : Si nbpoints $\geq$ 24 alors zone = manche

R2 : Si nbcarte $\geq$ 8 alors fité = vrai

R3 : Si fité = vrai, zone = manche alors envisagé = manche_à_la_couleur

R4 : Si couleur = pour_sûr, zone = manche alors envisagé = sans_atout

R5 : si nbpoints $\geq$ 30, fité = vrai alors zone = chaleur

[24 $\leq$ nbpoints et nbpoint < 30] zone = manche

R4 : si couleur = peu_sûr et nbpoints $\geq$ 24 et nbpoints < 30 alors envisagé = sans_aout

INC [nbcarte $\geq$ 8 et nbpoints $\geq$ 24 et nbpoints < 30 et couleur = peu_sûr]

R1' : Si 24 $\leq$ nbpoints < 30 alors zone = manche

Raisonnement non monotone

Rappel : Logique des défauts

$\frac{A : B}{C} \Leftrightarrow$ Si A alors C sauf si non B

$\Delta = < W , D >$

$\Gamma_\Delta(E)$: le plus petit ensemble de formules du langage
- contenant W
- déductivements clos
- tel que pour tous les défauts $\frac{A : B_1 \ldots B_n}{C} \in D$

pour lesquels $A \in \Gamma(E)$ et $\forall i \in [1 \ldots n]$, $\neg B_i \notin E$, on a $C \in \Gamma(E)$

E est un extension de L si et seulement si $E = \Gamma(E)$

Les faits

En général les êtres vivants ne volent pas
En général les oiseaux volent, à moins qu'ils ne soient trop jeunes ou qu'ils aient une aile cassée
Les pingouins et les autruches sont des oiseaux qui ne volent pas
Les autruches des bandes dessinées volent
Cuicui est un oiseau
Lulu est un pingouin ou une autruche
Lili est une autruche de bande dessinée
Paul est un homme
Les hommes sont des êtres vivants
Les oiseaux sont des êtres vivants

D = {

$$\frac{\text{être\_vivant}(X) : \neg\ \text{oiseau}(X)}{\neg\ \text{vole}(X)}$$

$$\frac{\text{oiseau}(X) : \neg\ \text{trop\_jeune}(X) \wedge \neg\ \text{aile\_cassée}(X) \wedge \neg\text{pingouin}(X)}{\text{vole}(X)}$$

$$\frac{\text{autruche}(X) : \neg\ \text{BD}(X)}{\neg\ \text{vole}(X)}$$

}

W = {
oiseau(Cuicui),
pingouin(Lulu) ⊕ autruche(Lulu),
autruche(Lili), BD(Lili),
homme(Paul),
homme(X) -> être_vivant(X),
oiseau(X) -> être_vivant(X),
autruche(X) -> oiseau(X),
pingouin(X) -> oiseau(X),
}

E = {
oiseau(Cuicui), pingouin (Lulu) ⊕ autruche(Lulu),
autruche(Lili), BD(Lili), homme(Paul), être_vivant(Cuicui), oiseau(Lulu),
oiseau(Lili), ¬ vole(Lulu), ¬ vole(Paul), vole(Cuicui), vole(Lili)
}

Exercice particulier

Δ = < W, D >
W = { A }

$$D = \left\{ \frac{A : \neg B}{B} \right\}$$

E ? On ne peux pas calculer E. Cette théorie n'admet pas d'extension.

Dempster Shaffer

Rappel

E : sous-ensemble de propositions
m : E -> [0…1]
(i) m(∅) = 0
(ii) ∑ m(A) = 1, A ⊆ E

1. μ(∅) = 0

$$A \neq \emptyset, \mu(A) = \frac{\sum_{S \cap T = A} m(S) * m(T)}{1 - \sum_{S \cap T = \emptyset} m(S) * m(T)}$$

$$pl(\varnothing) = 0$$

$$A \neq \varnothing, pl(S) = \frac{\sum_{A \cap B = S} m_1(A) * m_2(B)}{1 - \sum_{A \cap B = \varnothing} m_1(A) * m_2(B)}$$

$$E_1 : A_1 \ldots A_n \qquad \sum_{i=1}^{n} m_1(A_i) = 1$$

$$E_2 : B_1 \ldots B_m \qquad \sum_{j=1}^{m} m_2(B_j) = 1$$

$$\sum_{j=1}^{m} m_1(A_1) * m_2(B_j)$$

$$m_1(A_1) \underbrace{\sum_{j=1}^{m} m_2(B_j)}_{1} = m_1(A_1)$$

$$m_1(A_1) \sum_{A_1 \cap B_j \neq \varnothing} m_2(B_j) = m_1(A_1)(1 - \sum_{A_1 \cap B_j = \varnothing} m_2(B_j))$$

$$\sum_{i=1}^{n} m_1(A_i) \sum_{A_1 \cap B_j \neq \varnothing} m_2(B_j) = \sum_{i=1}^{n} m_1(A_i)(1 - \sum_{A_1 \cap B_j = \varnothing} m_2(B_j))$$

$$\sum_{i=1}^{n} m_1(A_i) - \sum_{i=1}^{n} m_1(A_i) \sum_{A_1 \cap B_j = \varnothing} m_2(B_j) = 1 - \sum_{i=1}^{n} m_1(A_i)(1 - \sum_{A_1 \cap B_j = \varnothing} m_2(B_j)) = 1 - \sum_{A_1 \cap B_j = \varnothing} m_1(A_i) m_2(B_j))$$

$$\frac{\sum_{i=1}^{n} m_1(A_i) \sum_{A_1 \cap B_j \neq \varnothing} m_2(B_j)}{1 - \sum_{A_1 \cap B_j = \varnothing} m_1(A_i) m_2(B_j))} = 1 = \sum_{S = A_1 \cap B_j \neq \varnothing} pl(S)$$

2.

$$m_A\left(\left\{pierre, paul\right\}\right) = 0{,}5$$

$$m_A\left(\left\{paul, marie\right\}\right) = 0{,}2$$

$$m_A\left(\left\{pierre, paul, marie\right\}\right) = 0{,}3$$

$$m_B\left(\left\{pierre\right\}\right) = 0{,}6$$

$$m_B\left(\left\{pierre, paul, marie\right\}\right) = 0{,}4$$

$$pl_{AB}(S) = \frac{\sum_{S = A \cap B \neq \varnothing} m_A(A) * m_B(B)}{1 - \sum_{S = A \cap B = \varnothing} m_A(A) * m_B(B)}$$

$$pl_{AB}\left(\left\{pierre\right\}\right) = \frac{0,6*0,5+0,6*0,3}{1-0,6*0,2} = \frac{48}{88}$$

$$pl_{AB}\left(\left\{paul,marie\right\}\right) = \frac{0,2*0,4}{1-0,12} = \frac{8}{88}$$

$$pl_{AB}\left(\left\{pierre,paul,marie\right\}\right) = \frac{0,3*0,4}{0,88} = \frac{12}{88}$$

$$pl_{AB}\left(\left\{pierre,paul\right\}\right) = \frac{0,5*0,4}{0,88} = \frac{20}{88}$$

$$m_C\left(\left\{paul,marie\right\}\right) = 0,5$$

$$m_C\left(\left\{pierre,paul,marie\right\}\right) = 0,5$$

$$pl_{ABC}\left(\left\{pierre\right\}\right) = \frac{\frac{48}{88}*0,5}{1-0,5*\frac{48}{88}} = \frac{24}{64}$$

$$pl_{ABC}\left(\left\{paul\right\}\right) = \frac{10}{64}$$

$$pl_{ABC}\left(\left\{pierre,paul\right\}\right) = \frac{10}{64}$$

$$pl_{ABC}\left(\left\{paul,marie\right\}\right) = \frac{14}{64}$$

$$pl_{ABC}\left(\left\{pierre,paul,marie\right\}\right) = \frac{6}{64}$$

On a d'abord calculer pl_{AB} puis pl_{ABC} , on trouve les mêmes sous-ensembles, donc la confrontation est associative.

Cas pathologique

$$\begin{cases} m_A\left(\left\{pierre\right\}\right) = 0,99 \\ m_A\left(\left\{paul\right\}\right) = 0,01 \\ m_B\left(\left\{marie\right\}\right) = 0,99 \\ m_B\left(\left\{paul\right\}\right) = 0,01 \end{cases} \Rightarrow \begin{cases} pl_{AB}\left(\left\{pierre\right\}\right) = 0 \\ pl_{AB}\left(\left\{marie\right\}\right) = 0 \\ pl_{AB}\left(\left\{paul\right\}\right) = 1 \end{cases}$$

3.

Grand et grand	0,6 * 0,2 = 0,12	Pierre, Paul
Grand et lunettes	0,6 * 0,7 = 0,42	Pierre
Grand et Ø	0,6 * 0,1 = 0,06	Pierre, Paul
Blond et grand	0,3 * 0,2 = 0,18	Paul
Blond et lunettes	0,3 * 0,7 = 0,21	Ø
Blond et Ø	0,3 * 0,1 = 0,03	Paul
Ø et grand	0,1 * 0,2 = 0,02	Pierre, Paul
Ø et lunettes	0,1 * 0,7 = 0,07	Pierre
Ø et Ø	0,1 * 0,1 = 0,01	Pierre, Paul, Jacques

$$pl\left(\left\{pierre\right\}\right) = \frac{0,42 + 0,07}{1 - 0,21} = \frac{49}{79} \simeq 0,62$$

$$pl\left(\left\{paul\right\}\right) = \frac{0,06 + 0,03}{1 - 0,21} = \frac{9}{79}$$

$$pl\left(\left\{pierre, paul\right\}\right) = \frac{0,12 + 0,06 + 0,02}{1 - 0,21} = \frac{20}{79}$$

$$pl\left(\left\{pierre, paul, jacques\right\}\right) = \frac{0,01}{1 - 0,21} = \frac{1}{79}$$

Mycin

Rappels

- Règle + affectée d'un coefficient p :

$$MB_{nouveau} = MB_{ancien} + p\left(1 - MB_{ancien}\right)$$

$$MD_{nouveau} = MD_{ancien}$$

- Règle – affectée d'un coefficient p :

$$MB_{nouveau} = MB_{ancien}$$

$$MD_{nouveau} = MD_{ancien} + p\left(1 - MD_{ancien}\right)$$

Si un même fait est la conclusion de n règles, la valeur finale de son CF (Certainty Factor) est-elle indépendante de l'ordre d'application des règles ?

Soient 2 règles renforçant le MB avec des coefficients p et p'

$$CF = MB_{nouveau} - MD_{nouveau}$$

avec

$$MB_{intermediaire} = MB_{ancien} + p\left(1 - MB_{ancien}\right) \qquad MD_{intermediaire} = MD_{ancien}$$

$$MB_{nouveau} = MB_{intermediaire} + p'\left(1 - MB_{intermediaire}\right) \qquad MD_{nouveau} = MD_{intermediaire}$$

$$MB_{nouveau} = MB_{ancien} + p\left(1 - MB_{ancien}\right) + p'\left(1 - MB_{ancien} - p\left(1 - MB_{ancien}\right)\right)$$

$$MB_{nouveau} = MB_{ancien} + \left(1 - MB_{ancien}\right)\left(p + p' - pp'\right) \qquad MD_{nouveau} = MD_{ancien}$$

Concordances de témoignages

1. E a eu lieu — E n'a pas eu lieu

   p(B dit E) ?

   A dit E(p) et B dit E(p')
   A dit ¬E(1 - p) et B dit E(1 - p')

   pp' + (1 - p)(1 - p')

2.

$$p\left(E \,/\, \text{A dit E} \wedge \text{B dit E}\right) = \frac{p\left(E \wedge \text{A dit E} \wedge \text{B dit E}\right)}{p\left(\text{A dit E} \wedge \text{B dit E}\right)} = \frac{p_E * p * p'}{p\left(\text{A dit E} \wedge \text{B dit E}\right)}$$

$$= \frac{p_E * p * p'}{p\left(\text{E} \wedge \text{A dit E} \wedge \text{B dit E}\right) + p\left(\neg\text{E} \wedge \text{A dit E} \wedge \text{B dit E}\right)}$$

$$= \frac{p_E * p * p'}{p_E * p * p' + \left(1 - p_E\right)\left(1 - p\right)\left(1 - p'\right)}$$

Algorithme A*

Résolution de problème par exploration d'un espace de recherche

Exemple : Recherche de chemin dans un graphe

2	8	3
1	6	4
7	_	5

état initial

1	2	3
8	_	4
7	6	5

But

Formalisation d'un problème

Définir ce qu'est un état et sa représentation

Problème

- Etat initial
- Ensemble d'opérateurs
- Fonction Test-But
- Fonction de coût

L'espace de recherche est représenté par un graphe d'états que l'on représentera par un arbre

Exemple : Taquin

1. Préciser la représentation du problème
2. Proposer une fonction qui évaluera la distance entre un état courant et le but
3. Développer une recherche par escalade pour ce problème

Etat

Position des 8 chiffres et de la case blanche

Opérateurs

Déplacer la case blanche :
- vers le haut
- vers le bas
- vers la droite
- vers la gauche

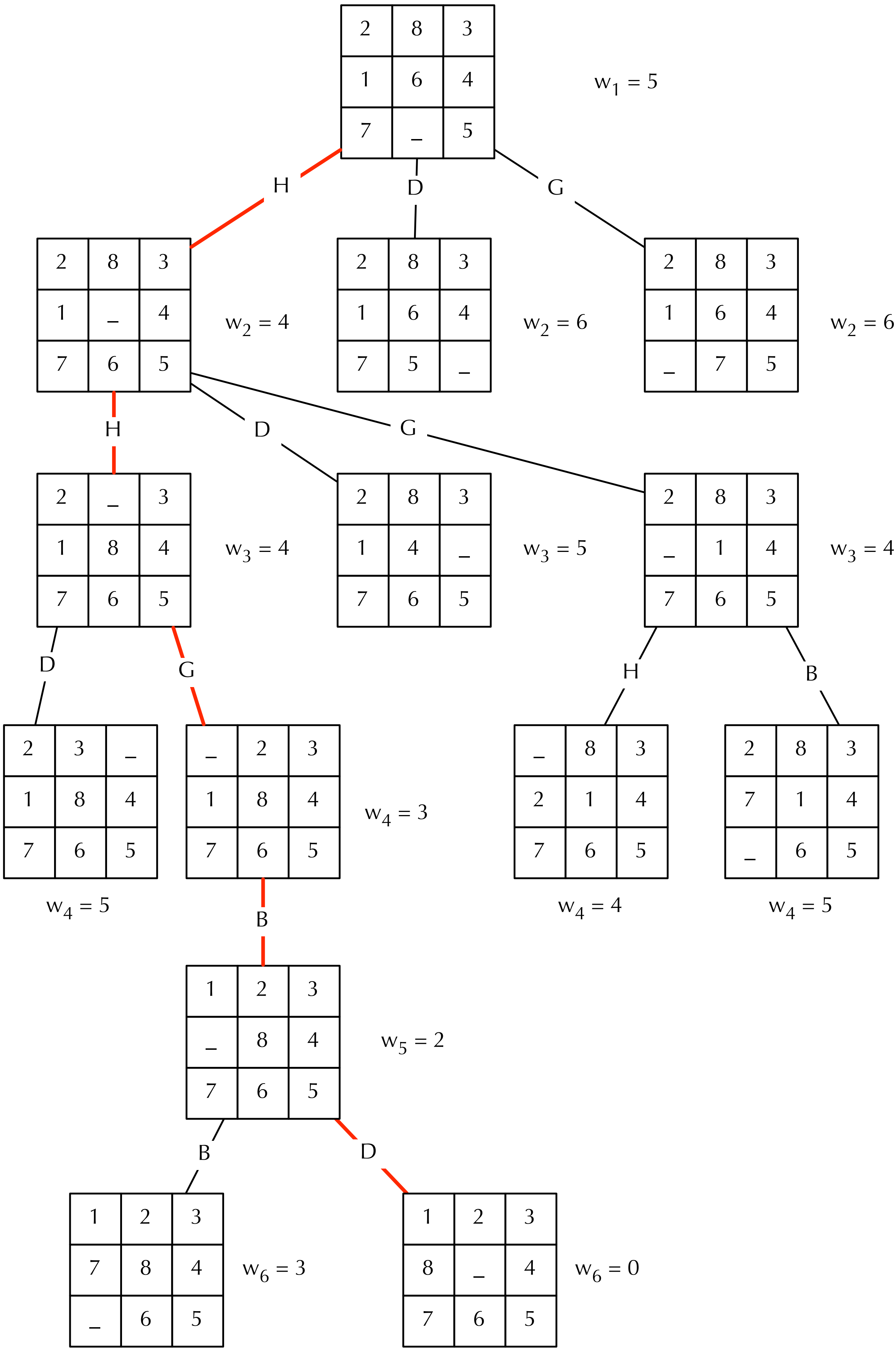
Fonction test-but

Comparer les positions des chiffres et de la case blanche dans un état avec les positions souhaitées

Fonction de coût

Coût de 1 à chaque opération

Evaluation de la distance entre un état et le but


2 8 3
1 6 4
7 _ 5
$w_1 = 5$
H
D
G
2 8 3
1 _ 4
7 6 5
$w_2 = 4$
2 8 3
1 6 4
7 5 _
$w_2 = 6$
2 8 3
1 6 4
_ 7 5
$w_2 = 6$
H
D
G
2 _ 3
1 8 4
7 6 5
$w_3 = 4$
2 8 3
1 4 _
7 6 5
$w_3 = 5$
2 8 3
_ 1 4
7 6 5
$w_3 = 4$
D
G
H
B
2 3 _
1 8 4
7 6 5
$w_4 = 5$
_ 2 3
1 8 4
7 6 5
$w_4 = 3$
_ 8 3
2 1 4
7 6 5
$w_4 = 4$
2 8 3
7 1 4
_ 6 5
$w_4 = 5$
B
1 2 3
_ 8 4
7 6 5
$w_5 = 2$
B
D
1 2 3
7 8 4
_ 6 5
$w_6 = 3$
1 2 3
8 _ 4
7 6 5
$w_6 = 0$

Recherche du meilleur d'abord : Profondeur + w_1

w′ : somme des nombres minimaux de déplacement pour amener chaque chiffre à sa place

2	8	3
1	6	4
7	_	5

w' = 1 + 2 + 1 + 1 = 5

w″ : faire tourner les cases externes autour du centre

2 pour chaque chiffre marqué par son successeur
1 pour case blache mal placée.

2	8	3
1	6	4
7	_	5

w" = 1 + 2 + 2 + 2 + 2 = 9

Complexité

En temps / en espace => en largeur d'abord / en profondeur d'abord

On suppose : chaque nœud a b successeurs (facteur de branchement)
Le problème a une solution de profondeur d.

Largeur d'abord

Trouver la solution si elle existe. Si plusieurs solutions, on trouve la moins profonde.

Complexité en temps : O (b^d)
Complexité en espace : O (b^d)

Exemple

b = 10

1000 nœuds peuvent être développés (et testés) par seconde.
1 nœud = 100 bits en mémoire

Profondeur	Nombre de nœuds	Temps	Mémoire
0	1	1 ms	100 bits
2	111	0,1 s	10 Kbits
4	11111	11 s	1 Mbits
6	10^6	18 mn	111 Mb
8	10^8	31 h	11 Gb

Recherche par escalade (Hill-climbing)

Parmi les fils d'un nœud, on développe celui qui a la plus petite valeur pour sa fonction heuristique (fonction d'évaluation). On gère une pile dans laquelle on stocke les fils dans l'ordre décroissant de la valeur heuristique (distance évaluée à la solution)

Le meilleur d'abord (Best-first)

La structure de données est une structure ordonnée dans l'ordre croissant. Chaque fois qu'on développe un nœud, on trie de nouveau sur l'ensemble des valeurs.

Branch and Bound

Au lieu de stocker des nœuds, on stocke des chemins incomplets. On ordonne la liste des chemins par coûts et on développe l'extrémité du chemin le moins coûteux. On n'a pas vraiment d'heuristique mais on a des coûts k.

Arrêt ?

Le chemin jusqu'à un but est moins coûteux que le moins coûteux des chemins incomplets (et non seulement quand on a atteint un but)

Algorithme A* = Brand and Bound + Heuristique

k + h

f(e) = g(e) + h*(e)

g(e) : coût de l'origine à e
h*(e) : coût de e à un objectif
f(e) : coût réel d'une solution passant par e.

f(e) = g*(e) + h*(e)

g*(e) : coût optimal de l'origine à e

f(e) = g(e) + h(e)

h(e) : estimation du coût d'un chemin de e à l'objectif

Finalement, f(e) est l'estimation du coût d'une solution passant par e. Quand on fait le Branch and Bound au lieu d'ordonner par f(e), on ordonne par g(e) + h(e) -> BB amélioré

Si de plus on retrouve une manière d'aller à e de coût inférieur au coût actuel, on ne retient que le chemin de coût minimal -> Algorithme A*

Etat u développé	**g(u), f(u), père(u)**	**successeurs de u mis dans A**	**successeurs de u non modifiés**	**Ensemble ordonné A par f , g , u_i(g(u), f(u))**
u_0	0, 17, _	u_1, u_3, u_5		u_5 (2, 17), u_1 (4, 19), u_3 (5, 21)
u_5	2, 17, u_0	u_7, u_8	u_3	u_8 (7, 14), u_7 (4, 17), u_1 (4, 19), u_3 (5, 21)
u8	7, 14, u5			u7(4, 17), u1(4, 19), u3(5, 21)
u7	4, 17, u5	u10	u8, u0	u1(4, 19), u3(5, 21), u10(19, 25)
u1	4, 19, u0	u2, u4	u3	u2(5, 20) u4(9, 21) u3(5, 21) u10 (19, 25)
u2	5, 20, u1		u3, u5, u7	u4(9,21) u3(5, 21) u10(19, 25)
u_4	9, 21, u1	u6, u9		u9(13, 20) u3(5, 21) u6(12, 23) u10(19, 25)
u_9	13, 20, u_4	u_{10}, u11, u4, u15		u10(14, 20) u3(5, 21) u6(12, 23) u11(15, 24) u15(25, 25)
u_{10}	14, 20, u_9	u_{12}, u_{13}	u_5	u13(18, 20) u12(15, 21) u3(5, 21) u6(12, 23) u11(15, 24) u15(25, 25)
u_{13}	18, 20, u_{10}	u_{16}	u_{15}	u_{12}(15, 21) u3(5, 21) u16(22, 22) u6(12, 23) u11(15, 24) u15(24, 24)

Etat u développé	g(u), f(u), père(u)	successeurs de u mis dans A	successeurs de u non modifiés	Ensemble ordonné A par f , g , u_i(g(u), f(u))
u_{12}	15, 21, u_{13}	u_{15}	u_{13}, u_{16}	u_{13}(5, 21) u_{16}(22, 22) u_6(12, 23) u_{11}(15, 24) u_{15}(24, 24)
u13	5, 21, u12	u6	u4	u_{16}(22, 22) u_6(12, 23) u11(15, 24) u15(24,24)
u16	22, 22, u13			

(u0, u1, u4, u9, u10, u13, u16)

f(u16) = 22

Algorithme AO*

Etat u développé	h(u)	f(u) ($f_1(u)$, … $f_i(u)$)	Etats actualisés après le dév. de u	Etats pendants dans $G(u_0)$
u_0		(55, **47**, 52)	∅	u_8
u_8	28	(**25**, 30)	u_0 (55, **44**, 52)	u_{18}
u_{18}	5	(**35**)	u_8 (55, **30**) u_0 (55, **49**, 52)	u_{19} u_{14}
u_{19}	11	(**25**)	u_8 (55, **44**) u_0 (55, 63, **52**)	u_3 (u_4)
u_3	40	(**38**, 44)	u_0 (55, 53, **50**)	u_7
u_7	30	(**44**)	u_3 (52, **44**) u_0 (**55**, 63, 56)	u_1 u_2
u_1	20	(**35**, 39, 47)	u_0 (70, 63, **56**)	u_9 u_{10}
u_9	25	(**35**)	u_3 (**52**, 54)	u_{14}
u_{14}	4	(**15**)	u_9 (**46**) u_8 (55, 55) u_7 (**53**) u_1 (**35**, 39, 53) u_3 (**63**, 65) u_0 (**70**, 74, 75)	u_5 u_2 u_6
u_5	16	(**15**, 18)	u_1 (**34**, 39, 58) u_0 (**69**, 74, 75)	u_2 u_6 u_{11}
u_2	30	(**30**, 59)	∅	u_6 u_{11} u_{13}
u_6	15	(**19**)	u_1 (**38**, 39, 58) u_0 (**73**, 74, 75)	u_{13} u_{11} u_{12}
u_{13}	10	(**10**, 65)	∅	u_{11} u_{12}
u_{11}	10	(**12**)	u_5 (**17**, 18) u_1 (40, **39**, 58) u_0 (**74**, 76, 75)	∅

MINI-MAX et élagage α-ß