Pierre De Loor – ENIB – LabSTICC- CNRS
www.enib.fr/~deloor

INTELLIGENCE ARTIFICIELLE ET SIMULATION

Objectifs :

- Module IAS – 2016-2017 - Responsable : Pierre De Loor
- **Objectifs (6 lignes max) :** Le module Intelligence Artificielle et Simulation a pour objectif d'initier les participants aux techniques de base de l'intelligence artificielle et à approfondir celles liées à la simulation de comportements intelligents.
- Deux Focus sont ensuite abordés : 1) L'usage des différentes techniques d'IA dans le domaine du jeu vidéo. 2) La modélisation et la simulation du « fonctionnement » de l'humain. Chaque partie abordée est composée d'une partie cours et de travaux pratiques.
- **Pré-requis :** Les connaissances acquises de S1 à S6 sont suffisantes pour aborder ce module

IA et Simulation

- Equipe 2017-2018:
- Pierre De Loor (deloor@enib.fr)
- Cindy Even (even@enib.fr)
- Anne-Gwenn Bosser (bosser@enib.fr)
- Pierre Chevaillier (chevaillier@enib.fr)
- Cédric Buche
- Organisation : cours/labo/travail personnel

Travail personnel

Utilisation des connaissances acquises en module pour la programmation d'un bot Crédible dans UT2004.

Journées « type »

Jour	H1	H2	H3	H4	H5	H6
ndi /09/17	H1 08:05 - 09:30 09AO-IAS DEP historique CM 2-E205 09AO-IAS	09AO-IAS DEP Introduction - Historique CM 2-E205 09AO-IAS		09AO-IAS BOA IA & Jeu Vidéo CM 2-E205 09AO-IAS	09AO-IAS BOA IA & Jeu Vidéo CM 2-E205 09AO-IAS	
ndi /09/17	09AO-IAS BOA IA & Jeu Vidéo CM 2-E205 09AO-IAS	09AO-IAS BOA IA & Jeu Vidéo CM 2-E205 09AO-IAS		09AO-IAS BOA IA & Jeu Vidéo Labo B005 09AO-IAS	09AO-IAS BOA IA & Jeu Vidéo Labo B005 09AO-IAS	09AO-IAS BOA Travail personnel Labo B005 09AO-IAS
ndi /09/17	09AO-IAS BOA IA & Jeu Vidéo Labo B005 09AO-IAS	09AO-IAS BOA IA & Jeu Vidéo Labo B005 09AO-IAS		09AO-IAS CHP Réseaux de neurones CM 2-E205 09AO-IAS	09AO-IAS CHP Réseaux de neurones CM 2-E205 09AO-IAS	09AO-IAS BOA Travail personnel Labo B005 09AO-IAS
ndi /10/17	09AO-IAS CHP Réseaux de neurones Labo B015 09AO-IAS	09AO-IAS CHP Réseaux de neurones Labo B015 09AO-IAS		09AO-IAS DEP Apprentissage par renforcement CM 2-E205 09AO-IAS	09AO-IAS DEP Apprentissage par renforcement CM 2-E205 09AO-IAS	09AO-IAS CHP Travail personnel Labo 09AO-IAS
ndi /10/17	09AO-IAS DEP Apprentissage par renforcement Labo B015 09AO-IAS	09AO-IAS DEP Apprentissage par renforcement Labo B015 09AO-IAS		09AO-IAS DEP Apprentissage par renforcement Labo B015 09AO-IAS	09AO-IAS DEP Apprentissage par renforcement Labo B015 09AO-IAS	09AO-IAS DEP Travail personnel Labo B005 09AO-IAS
ndi /10/17	09AO-IAS BUC Réseaux Bayesiens CM 2-E205 09AO-IAS	09AO-IAS BUC Réseaux Bayesiens CM 2-E205 09AO-IAS		09AO-IAS BUC Réseaux Bayesiens Labo B015 09AO-IAS	09AO-IAS BUC Réseaux Bayesiens Labo B015 09AO-IAS	09AO-IAS BUC Travail personnel Labo B005 09AO-IAS
ndi /10/17	09AO-IAS BOA Logiques CM 2-E205 09AO-IAS	09AO-IAS BOA Logiques CM 2-E205 09AO-IAS		09AO-IAS BOA Travail personnel Labo B005 09AO-IAS		
	09AO-IAS BOA	09AO-IAS BOA		09AO-IAS BOA	09AO-IAS BOA	09AO-IAS BOA

Évaluation du résultat du travail personnel en fin de semestre

Evaluation en CC en fin de P1

COMPÉTITION

& Intelligence Artificielle Jeux Vidéo

Lors de **PFIA'17**

à Caen, entre le 3 et le 7 Juillet 2017

Pour gagner, **DÉVELOPPEZ LE BOT** le plus **HUMANLIKE***!

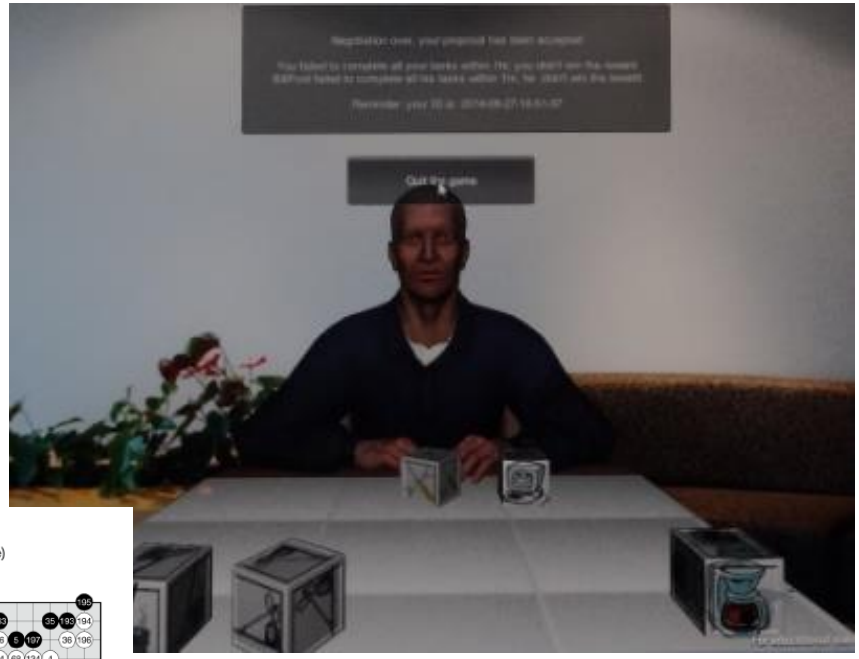
Plus d'info : http://goo.gl/rVZODq

Contact : botcontest2017@enib.fr

AfIA

Chapitre 1

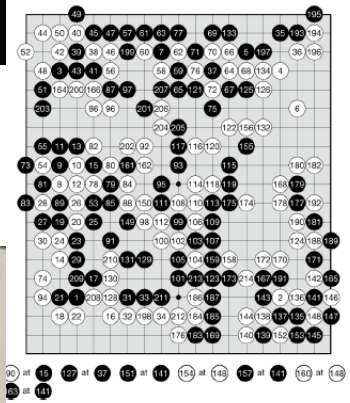
PRESENTATION GENERALE



Game 5

Fan Hui (Black), AlphaGo (White)

AlphaGo wins by resignation

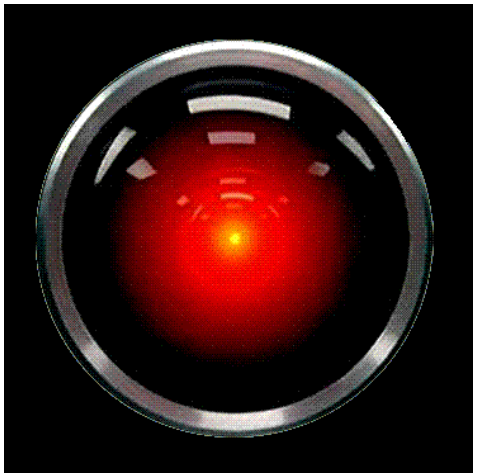


Oui, mais …

– Supercalculateur K (RIKEN, Japon, 2013)

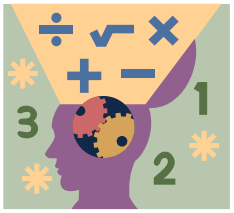
– Simulation de 1% du cerveau (1,73 milliard de neurones et 10,4 billiard de synapses)
– 83 000 CPU, 1Po, 20 minutes de simulation pour 1 seconde de fonctionnement cérébral

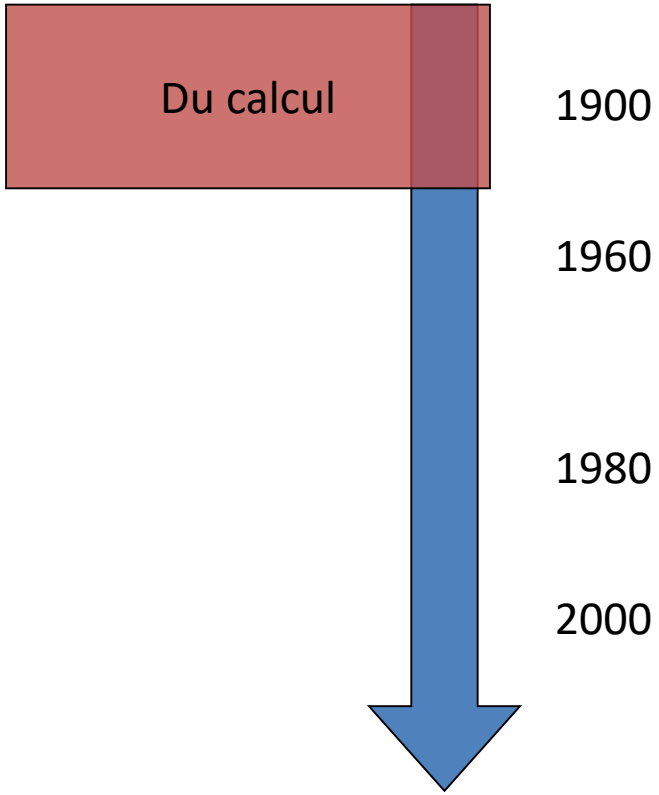
Un peu d'histoire



Un peu d'histoire

- L'intelligence artificielle c'est

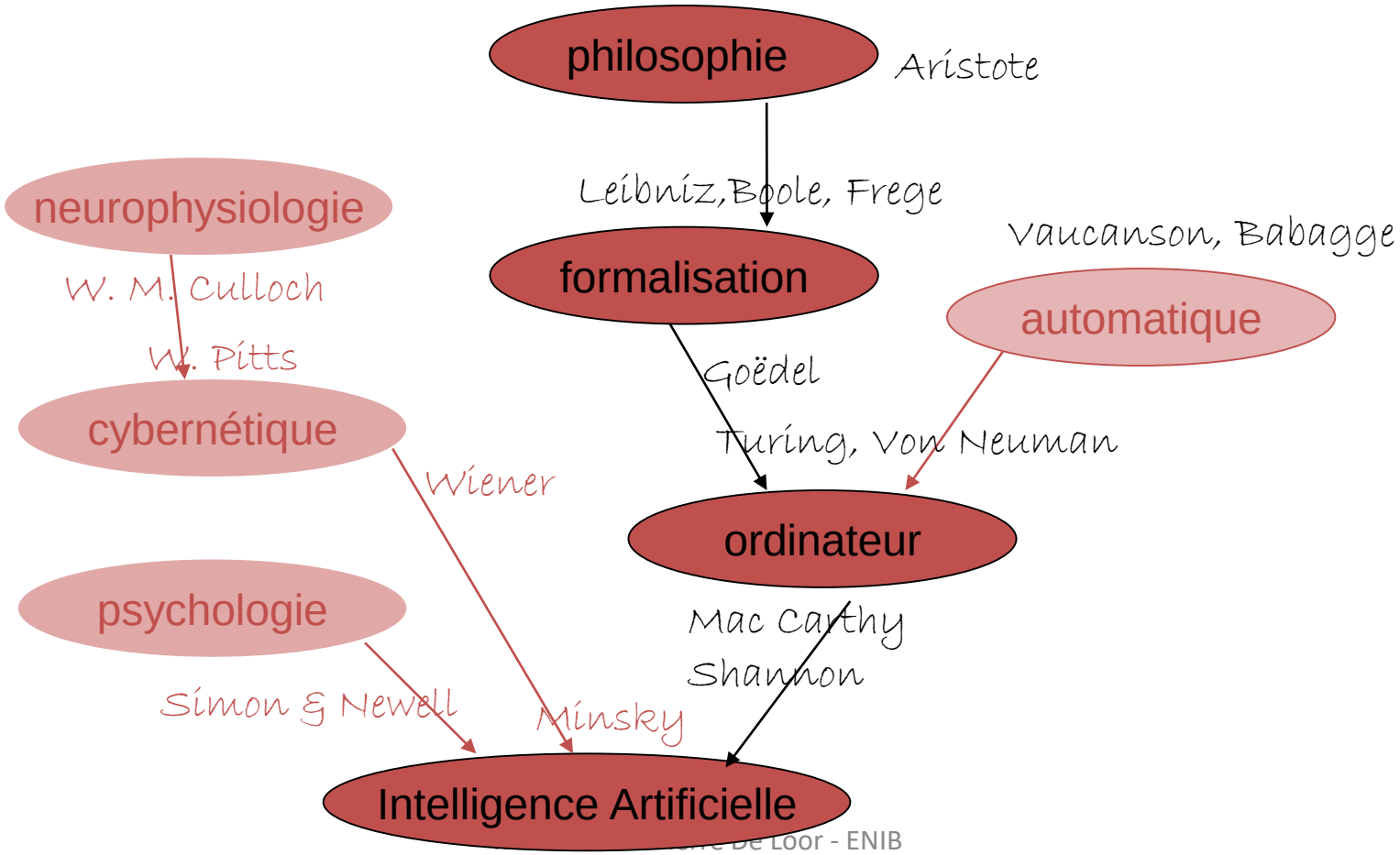




Du calcul …

Automate + algorithme + algèbre de Boole =
algorithme universel
=machines pensantes

la formalisation de la pensée



la logique

- Le syllogisme

aucune vulgarisation n 'est aisée

quelques travaux sont des vulgarisations

quelques travaux ne sont pas aisés

Aristote (-384 à -322)

… et sa formalisation

- L'algèbre de Boole

  Formulation mathématique de l'ensemble des lois de la pensée humaine (Boole 1854)

  – A et B c'est A*B
  – A implique B c'est ¬A+B

- Le calcul des prédicats
- Principia Mathématica

Frege (1848-1925), Whitehead et Russell (~1915)

un premier doute

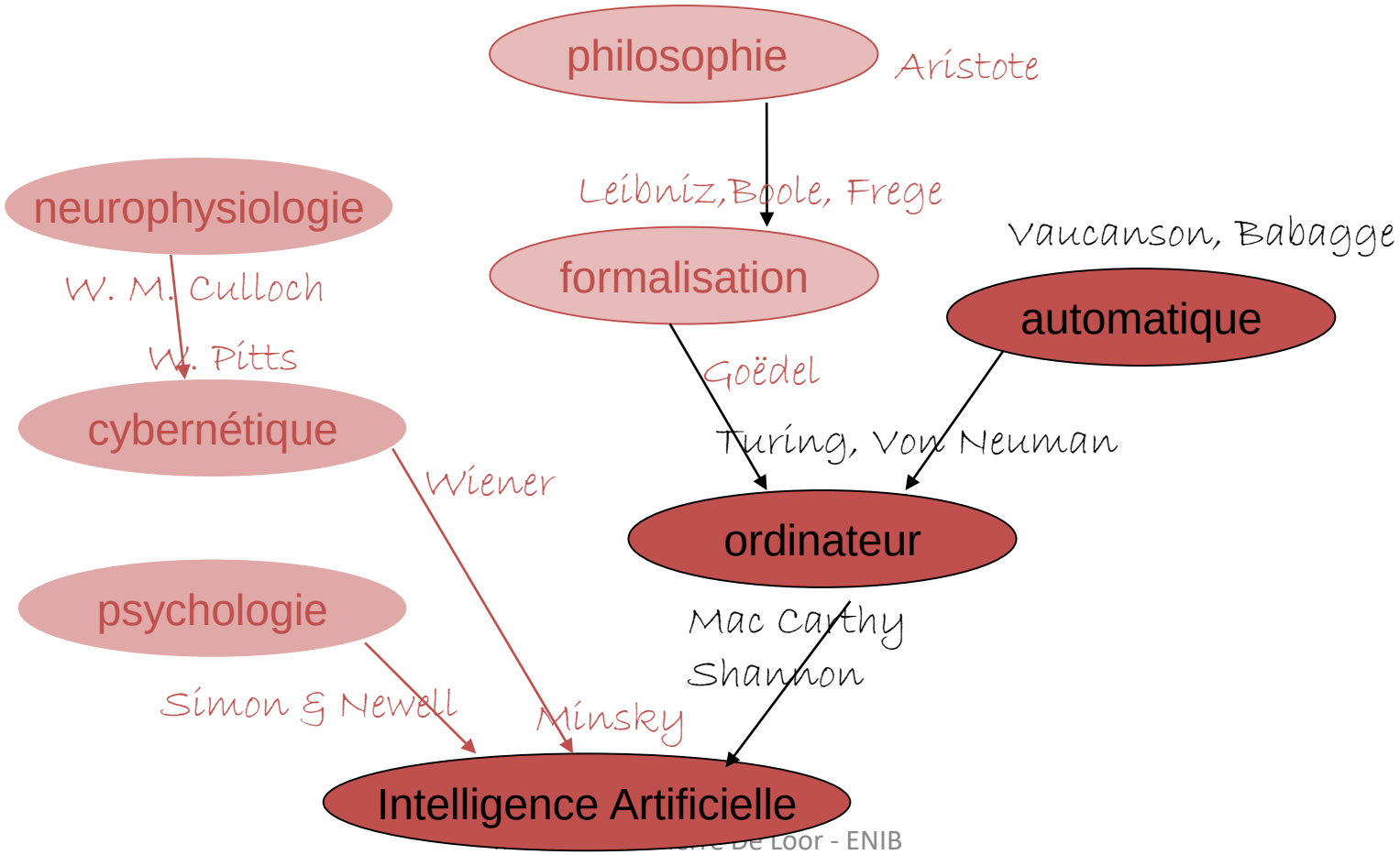
Existe-t-il un algorithme pour résoudre tout problème ?

(Leibniz 1646-1716)

Non

(Goëdel 1931)

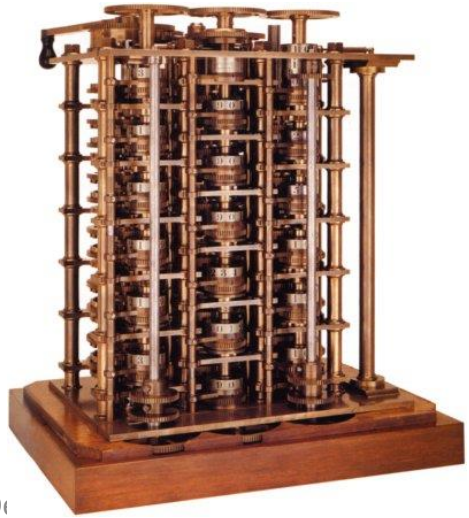
l'automatique



l'automatique

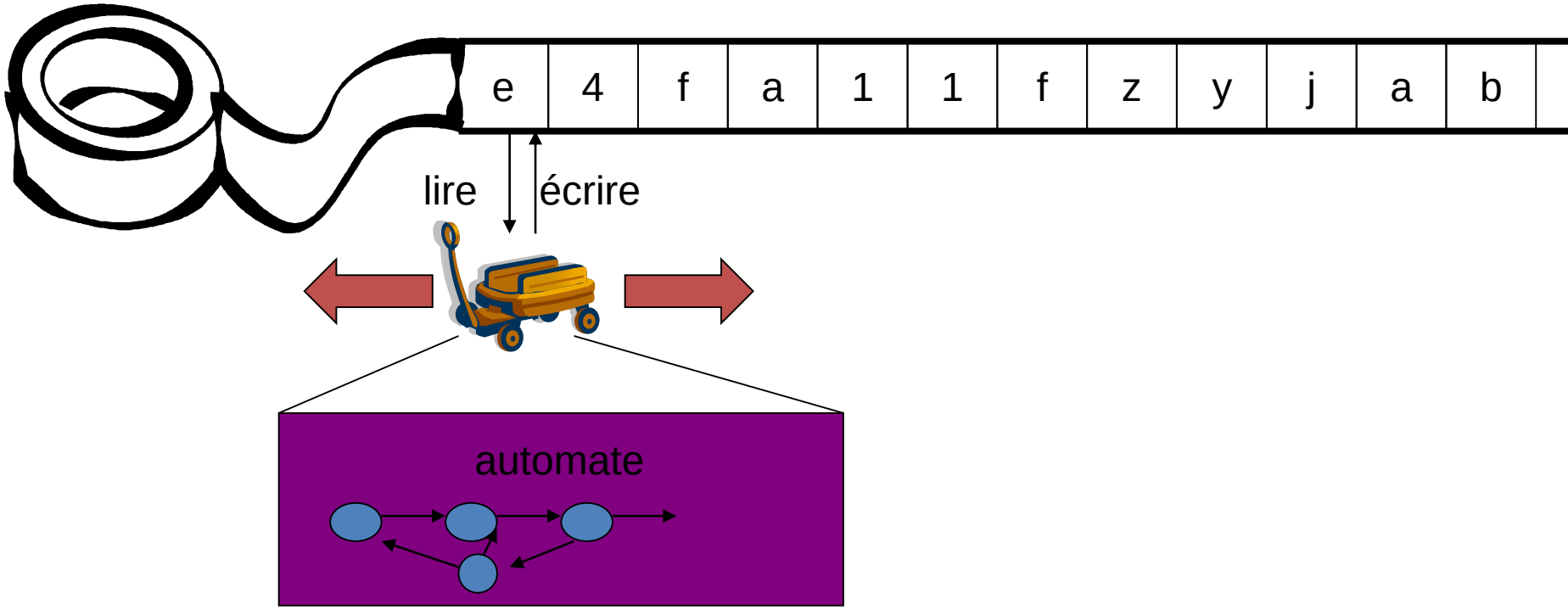
- Automates
- Calculatrices mécaniques
- Cartes perforées
- Machine analytique

Pascal, Leibniz, Vaucanson, Jacquart, Watts, Babbage



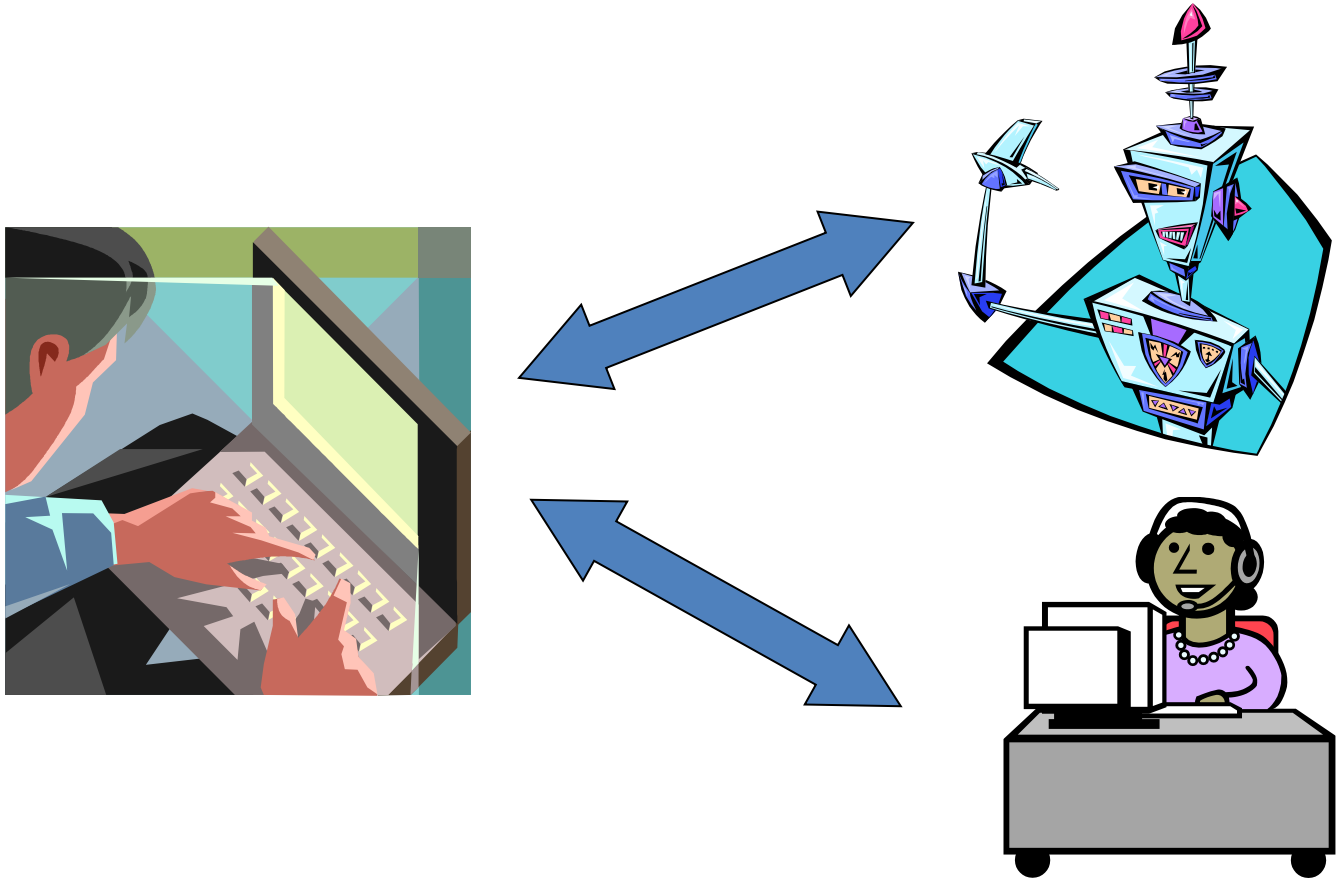
Vers l'ordinateur

- La machine de Turing (1936)



Équivalence Turing/Von Neuman

Le Test de Turing (1950)

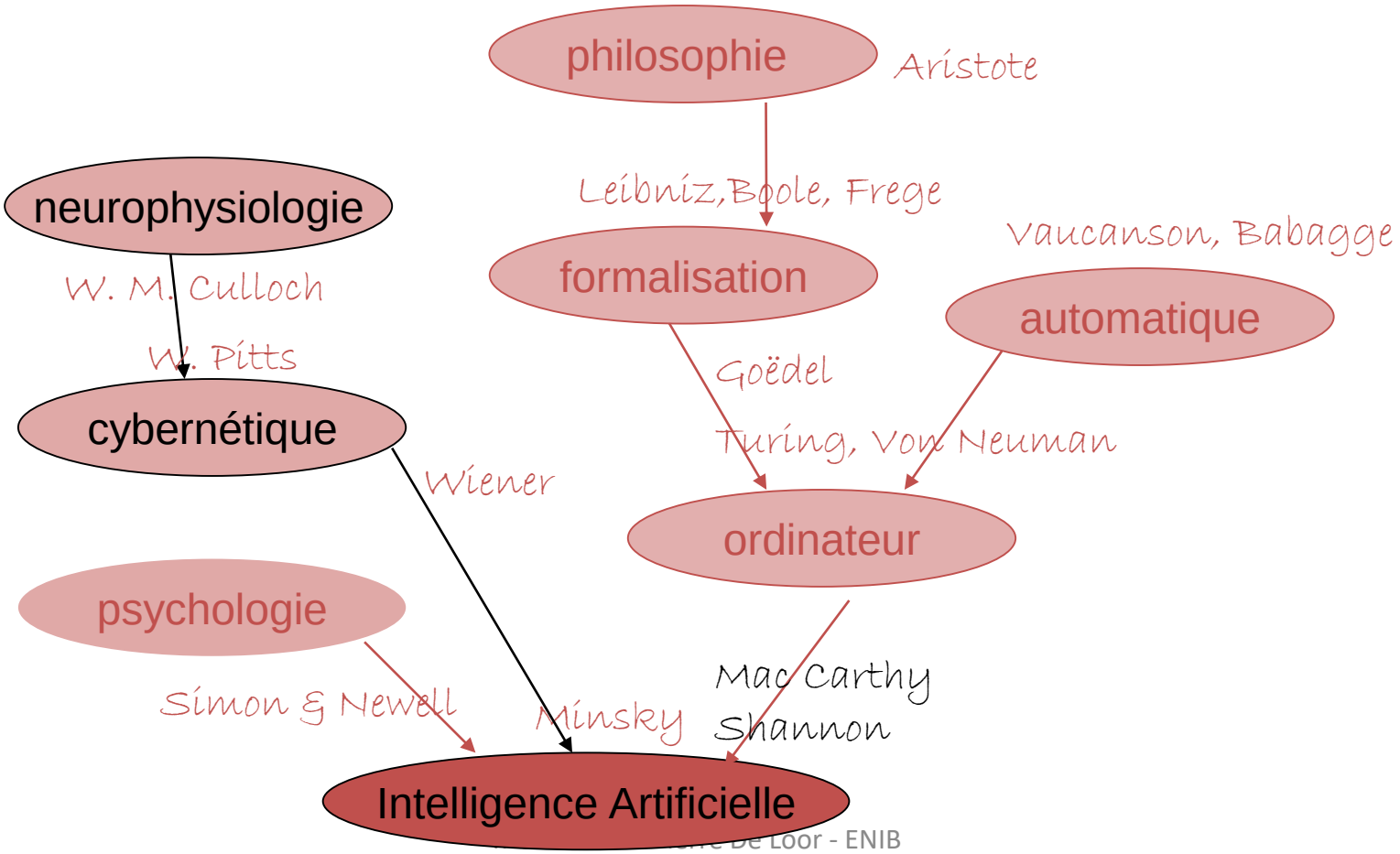


Un autre doute :

Est-il possible de faire une MT capable de dire si une autre MT s'arrête ou non ?

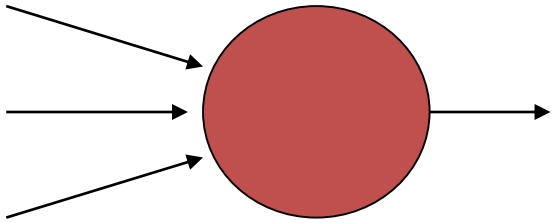
Non : indécidabilité de l'arrêt (Turing 1936).

La cybernétique

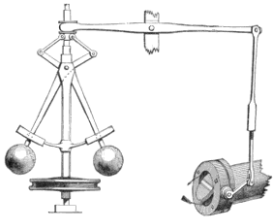


La Cybernétique

- Neurophysiologie + Mathématique = neurone artificiel (W.M. Culloh, W. Pitts 1943)



- Notion de système (Wiener)
- Principe d'Homeostasie
- Apprentissage



… et des polémiques

« vouloir reproduire le comportement des nerfs ne présente pas plus d'intérêt que de construire une voiture avec des jambes alors qu'on peut faire beaucoup mieux avec des roues (A. Turing) »

- Arrêt de l'approche neuronale
  - Succès de l'architecture de Von Neumann
  - Difficultés théoriques
  - Difficulté humaine

Naissance du terme

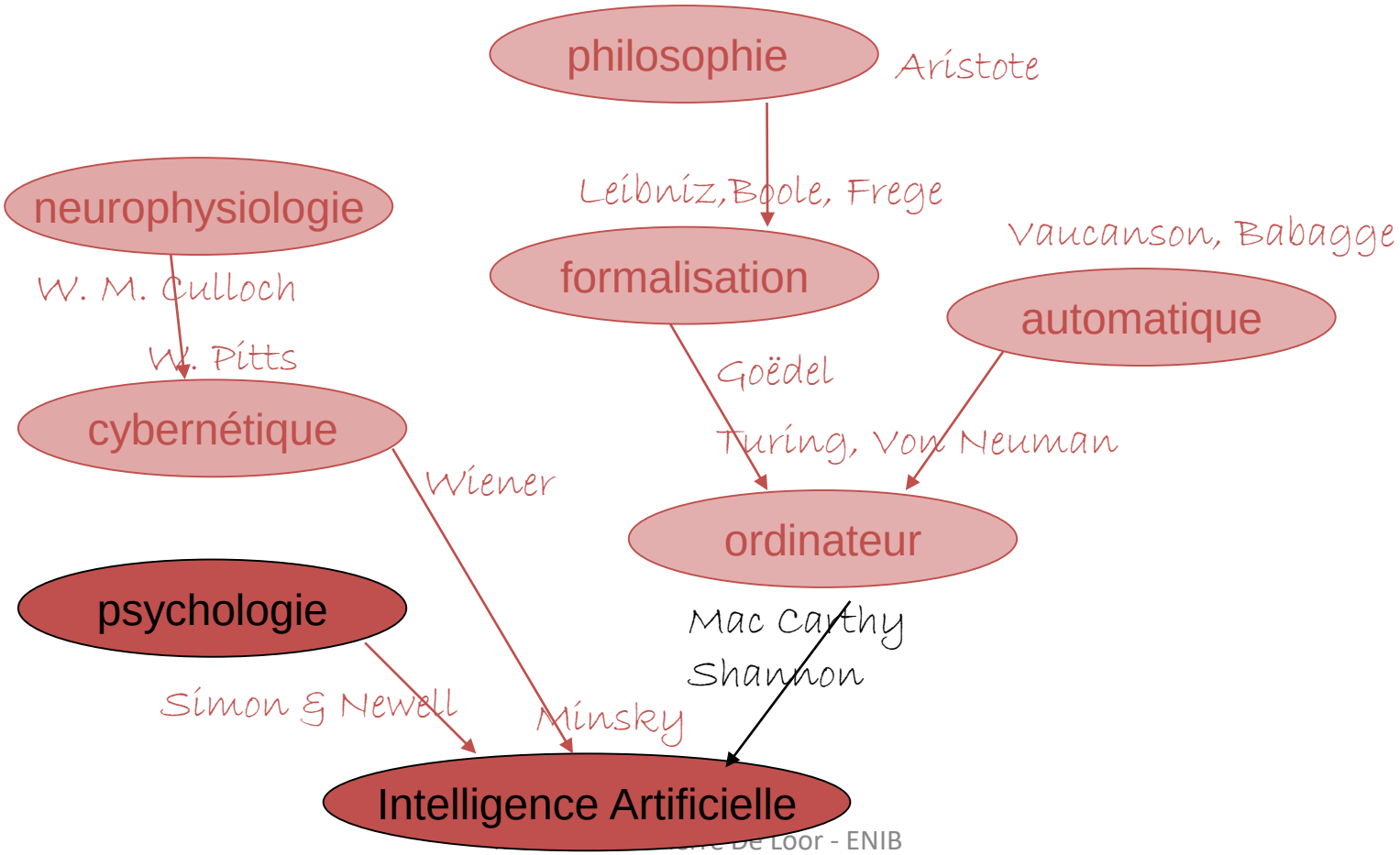
Intelligence Artificielle

- Conférence Dartmount 1956
  - Mac Carthy : langages
  - Shannon : automates
  - Minsky : critique de l'approche neuronale

- Deux nouveaux :

Herbert Simon et Allen Newell

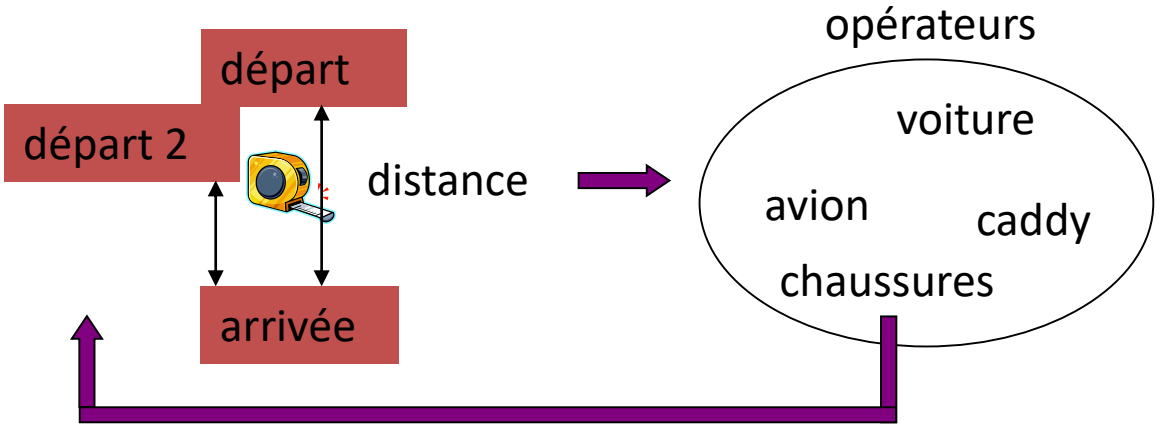
Introduction de la psychologie



introduction de la psychologie

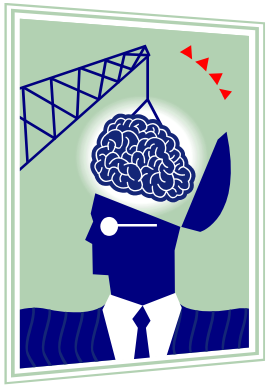
- Le Logic Theorist redémontre 38 des 52 théorèmes des « Principea Mathematica »

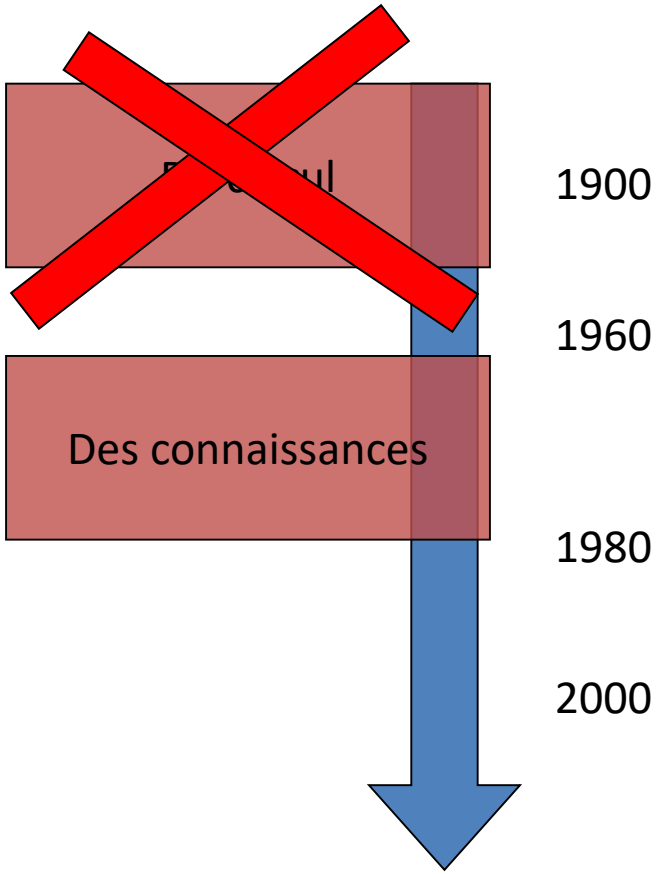
- General Problem Solver (GPS, 1959) :
  - Méthode générique : Analyse des fins et des moyens



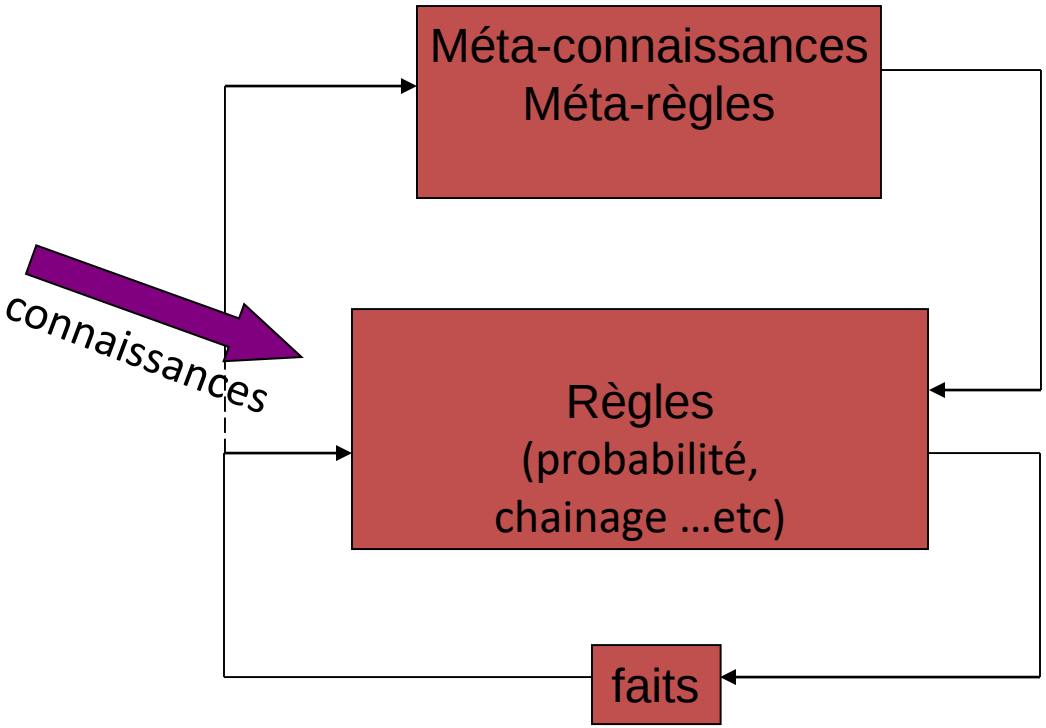
Histoire

- L’intelligence artificielle c’est





Les systèmes experts



Exemple : MYCIN (1970, 500 règles, probabilités)

Oui mais il y a DES Connaissances

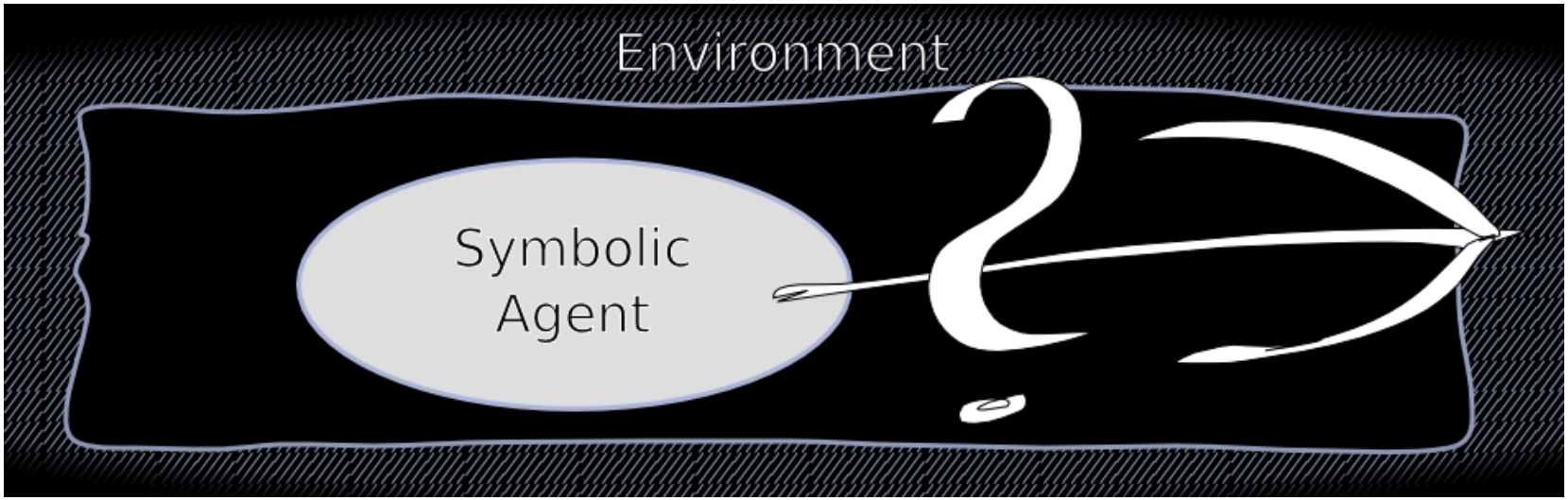
- Connaissance procédurale
  - Savoir faire, savoir comment
- Connaissance déclarative
  - Mémoire sémantique
    - je sais que
  - Mémoire épisodique
    - je me rappelle que
- Connaissance représentationnelle liée à l'action
  - Là je dois faire comme ça
- Connaissance catégorielle
  - La catégorie 'oiseau' est représentée par le moineau
  - La catégorie 'mammifère' est représentée par la vache
  - Canari -> oiseau
  - Autruche -> pas oiseau
  - Baleine -> poisson
- Schémas typiques
  - Aller faire ses courses
  - Conduite automobile

Oui mais, l'ordinateur ne pense pas

- [Searles 1980]

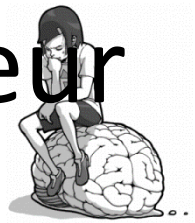
Symbol grounding problem [Harnad 1990]

- Comment la signification des symboles est ancrée dans quelque chose qui n'est rien d'autre que des symboles sans signification ?



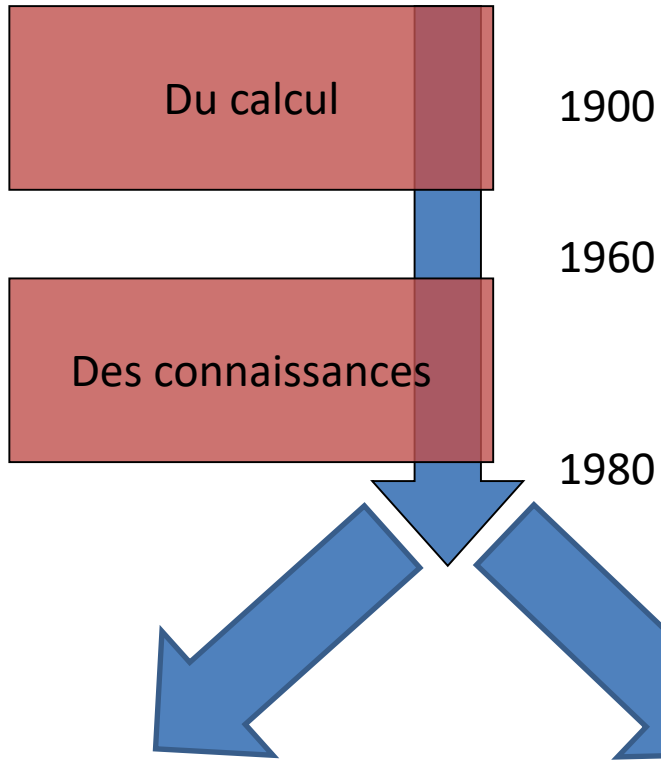
- Cognition is not the same as computation because computer do not transduce [Harnad 1990]

Le cerveau n’est pas un ordinateur



- Le cerveau n’est pas un ordinateur mais un “interacteur autonome” et il est impossible de séparer le cerveau de son environnement pour comprendre la connaissance
- Il n’y a pas de représentations pré-données “universelles”. Elles émergent.
- La cognition d’un individu ne peut être séparée de son histoire et de sa culture

L'IA la scission



IA « faible »
Informatique Avancée

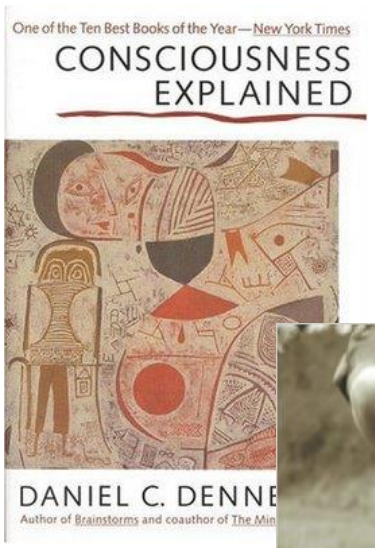
Programmation déclarative/ résolution de problèmes
Théorie des jeux
Apprentissage artificiel
Ingénierie des connaissances
Systèmes multiagents

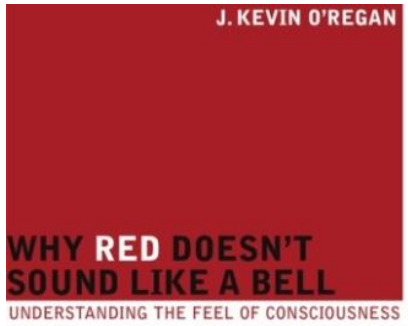
IA « FORTE »

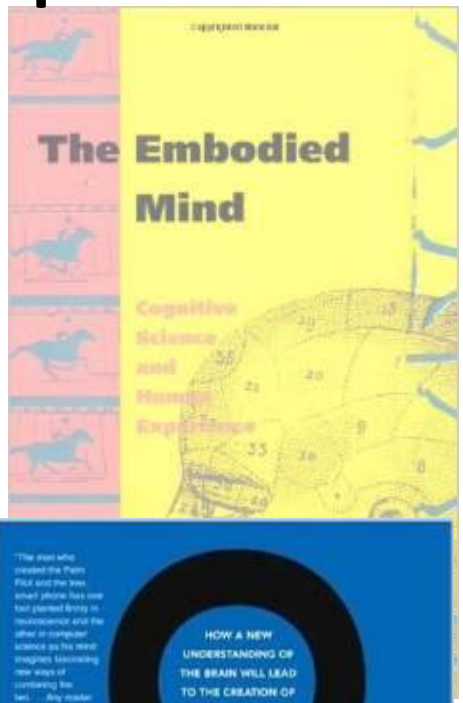
Sciences cognitives
Agentivité
Vie Artificielle
Conscience Artificielle

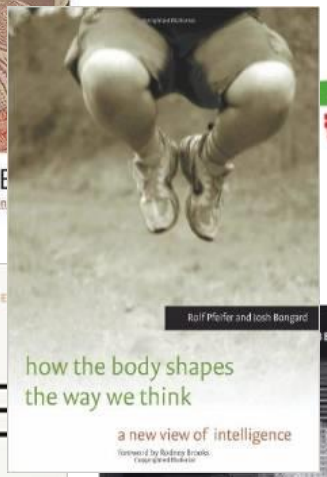
Simulation comportementale

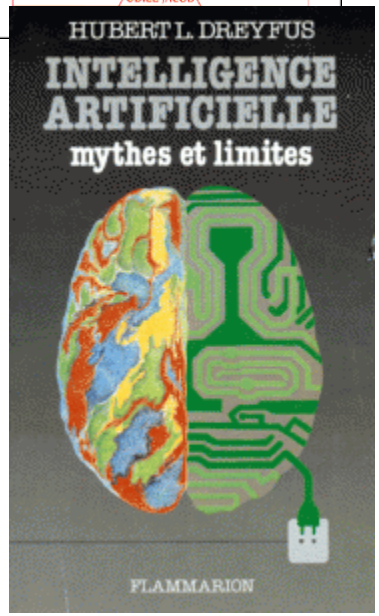
L'IA Forte et ses polémiques

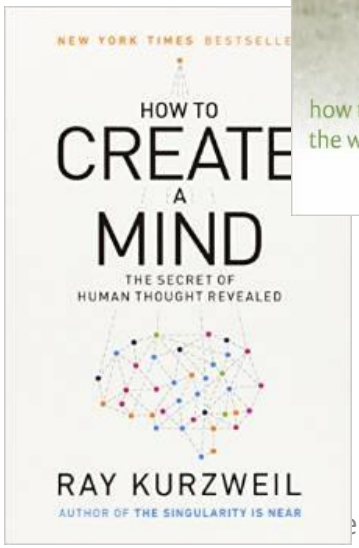


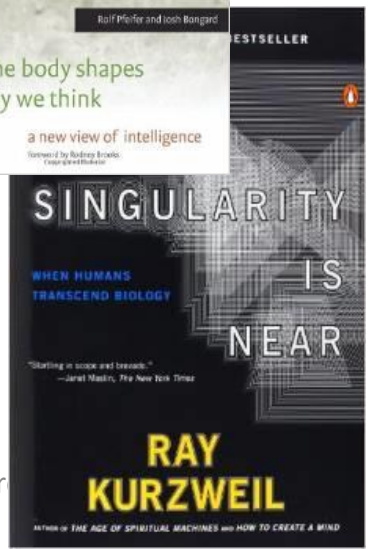


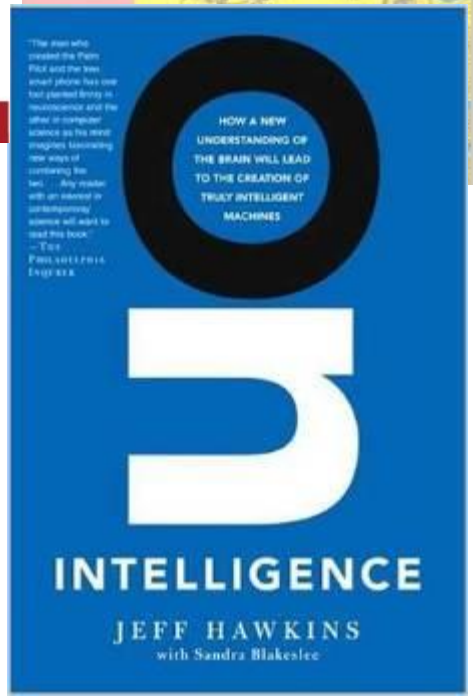










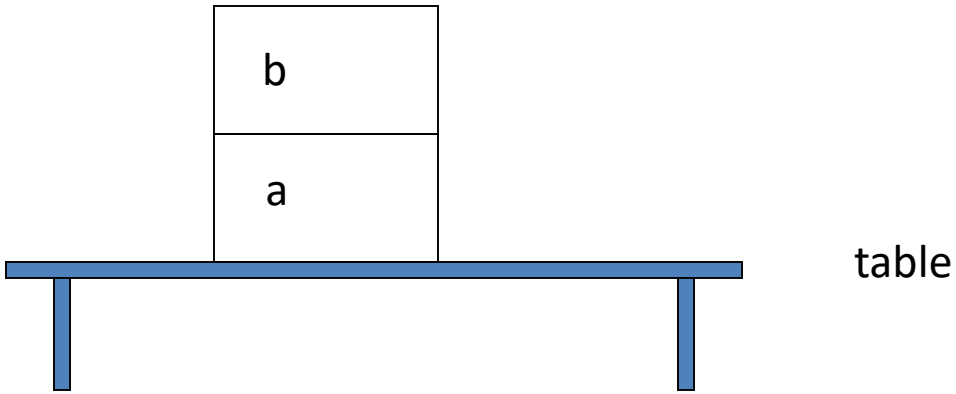


Question 1

- Comment fait-t-on un programme qui fait une déduction ?

- Réponse
  - Faits
  - Régles
  - Inférence
  - Langage PROLOG (Programmation Logique)

Démonstration automatique



Sachant que b est sur a et a est sur table, déduire que b est au-dessus de la table

Démonstration automatique

- Note : les variables s ’écrivent X, Y , …
- Axiomes :

  $\forall X \forall Y [sur(X,Y) \rightarrow auDessus(X,Y)]$

  $\forall X \forall Y \forall Z [auDessus(X,Y) \wedge auDessus(Y,Z) \rightarrow auDessus(X,Z)]$

- Peut s’écrire ainsi (forme normale)

  $\neg sur(U,V) \vee auDessus(U,V)$ (1)

  $\neg auDessus(X,Y) \vee \neg auDessus(Y,Z) \vee auDessus(X,Z)$ (2)

- Deux axiomes supplémentaires spécifiques à l ’exemple

  $sur(b,a)$ (3)

  $sur(a,table)$ (4)

- Démonstration : voir Annexe A.1

Un programme PROLOG

```
sur(a,table).
sur(b,a).
auDessus(X,Z):- sur (X,Z).
auDessus(X,Z):- auDessus(X,Y), auDessus(Y,Z).

Question à prolog
-? auDessus(b,table).

yes
```

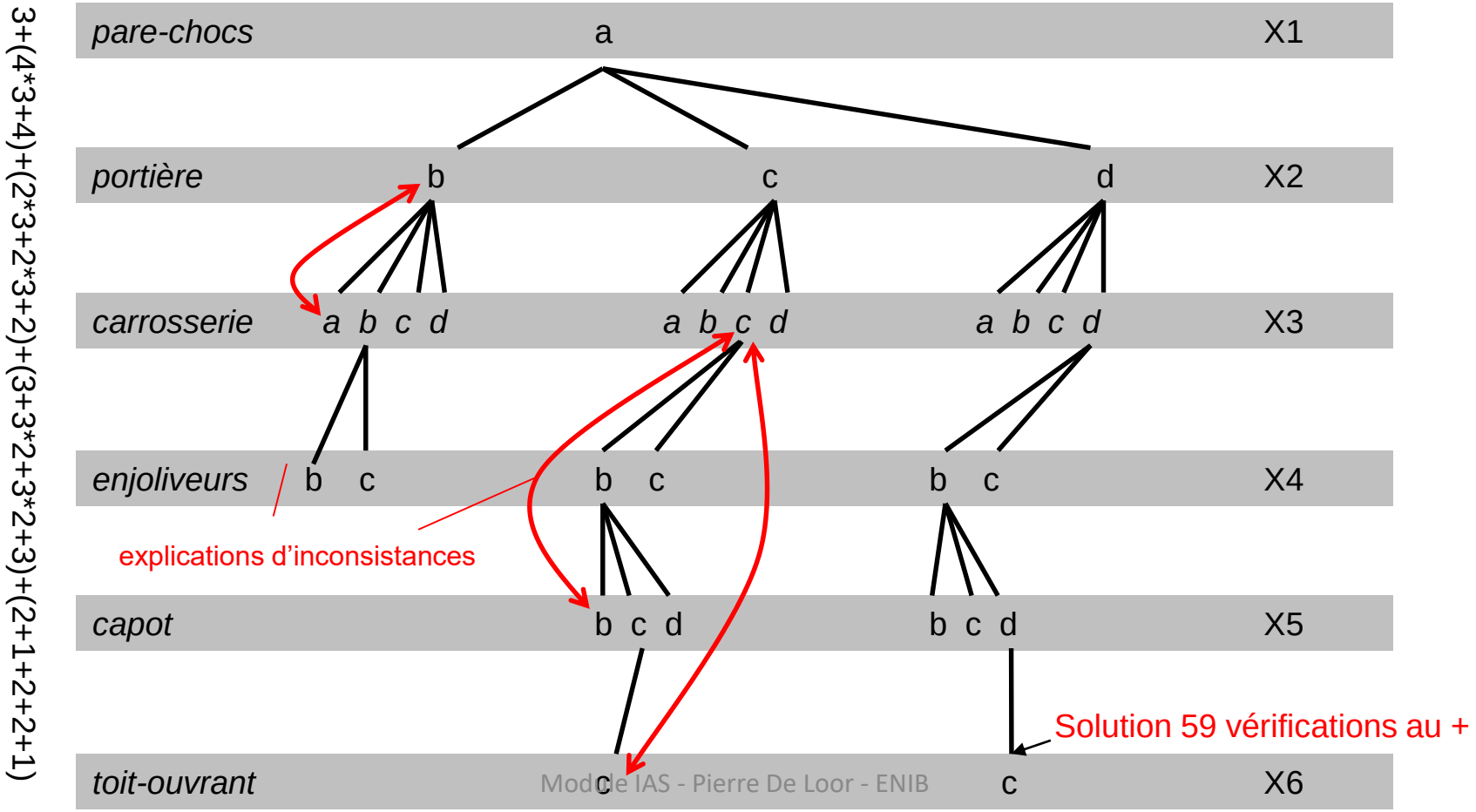
Question 2 : Comment fait-on un programme qui « trouve une solution »

- Fabrication d'une voiture
  - Les portières et le capot sont faits à Lille, mais le constructeur ne dispose que de peinture rose, rouge et noire
  - La carrosserie est à Hambourg, où l'on a de la peinture blanche, rose, rouge et noire
  - Les pare-chocs, faits à Palerme, sont toujours blancs
  - La bâche du toit ouvrant, qui est faite à Madrid, ne peut être que rouge
  - Les enjoliveurs sont faits à Athènes, où l'on a de la peinture rose et de la peinture rouge

  - La carrosserie doit être de la même couleur que les portières, les portières de la même couleur que le capot, le capot de la même couleur que la carrosserie
  - Les enjoliveurs, les parc chocs et le toit ouvrant doivent être plus clairs que la carrosserie

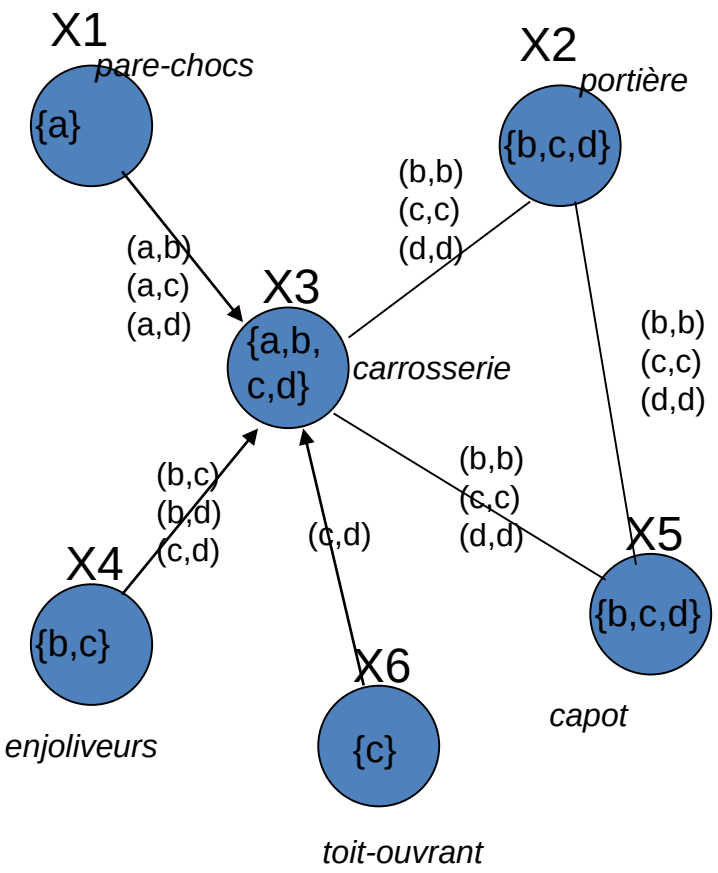
Algorithme du backtrack

- Tester avant de continuer à générer :
  - tester toutes les variables instanciées une à une



Les CSP …

a:blanc, b:rose, c:rouge, d:noir



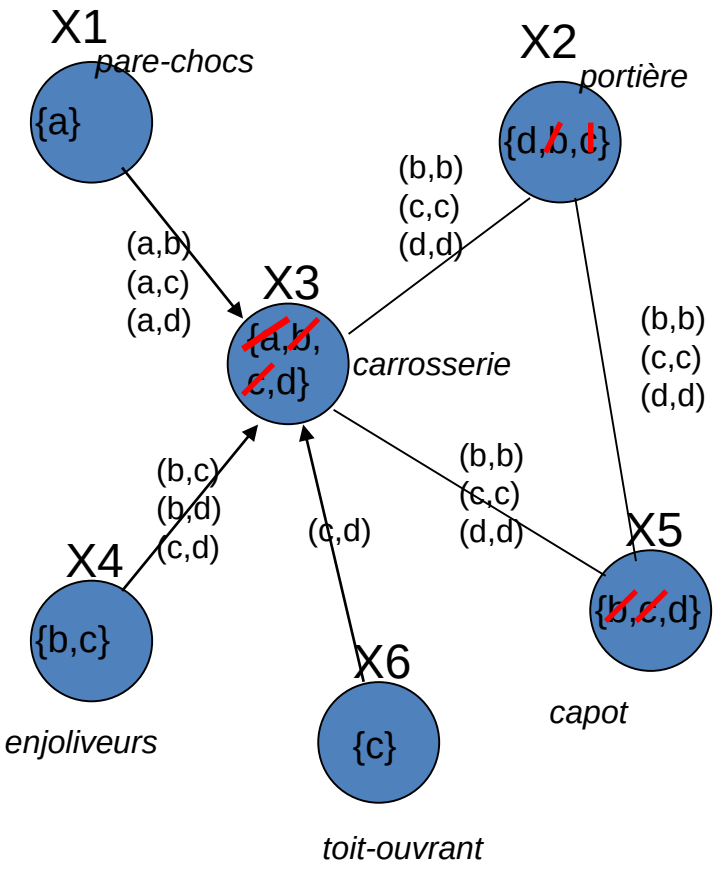
- Définition formelle
- P=(X,D,C)
- X={x1,x2,x3,x5,x6}
- D={(a),(a,b,c),(a,b,c,d),(b,c),(b,c,d),(c)}
- C={ (x1,x3):((a,b),(a,c),(a,d)),
  (x2,x3):((b,b),(c,c),(d,d))
  …}
- Solutions
  - S1={x1→a,x2→d,x3→d,x4→b,x5→d,x6→c}
  - S2={x1→a,x2→d,x3→d,x4→c,x5→d,x6→c}

Gestion de la complexité …

- Trouver une solution est NP-difficile
- Les algorithme de résolution de contraintes tentent de réduire cette complexité …

Exemple : le filtrage

- Arc-consistance de P



X1-X3 : (a,a) impossible

X2-X3 : pas de problèmes

X6-X3 : (c,b) et (c,c) impossibles

X2-X3 : (b,b) et (c,c) impossibles

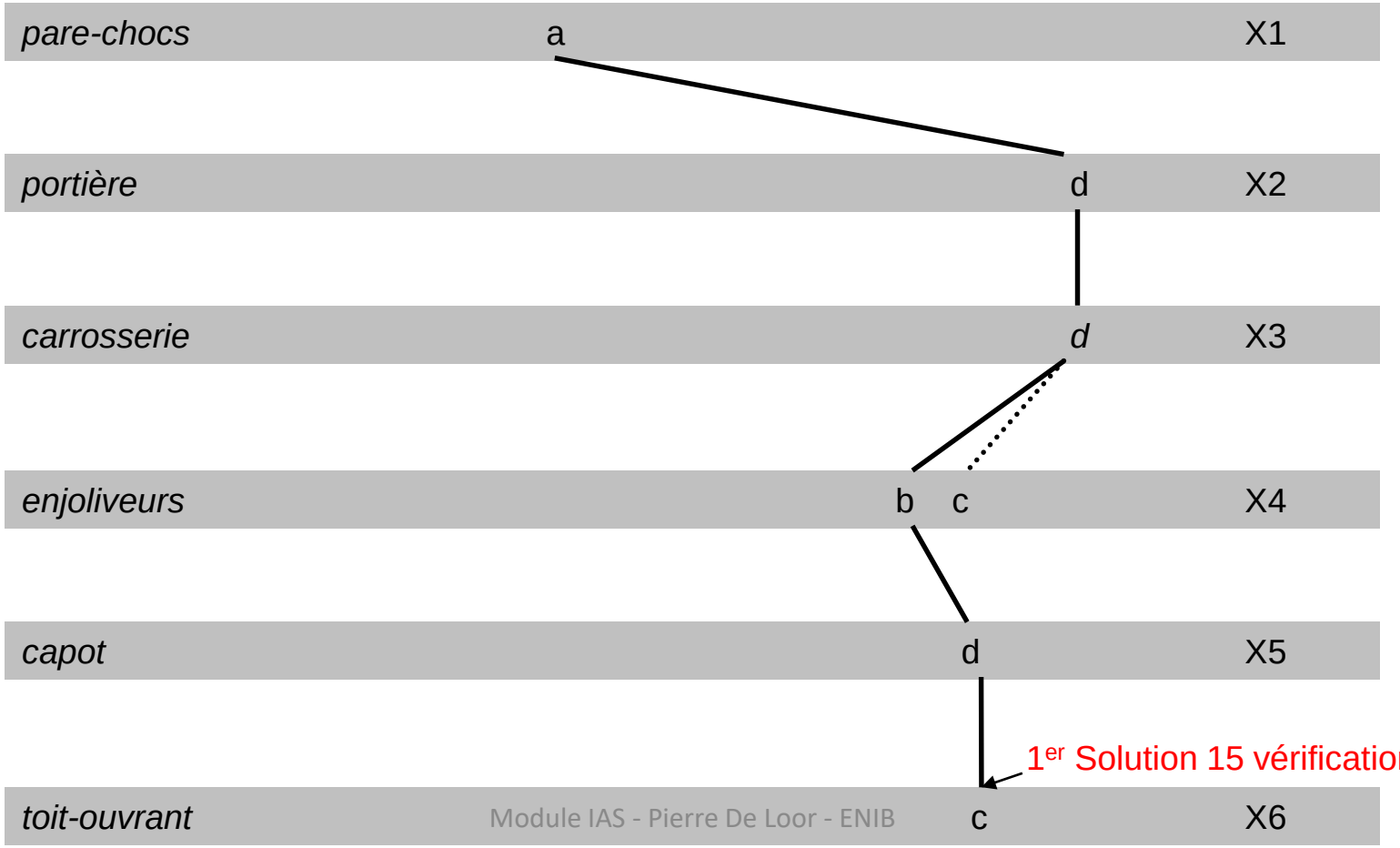
X3-X5 : (b,b) et (c,c) impossibles

Procédure AC3

P''

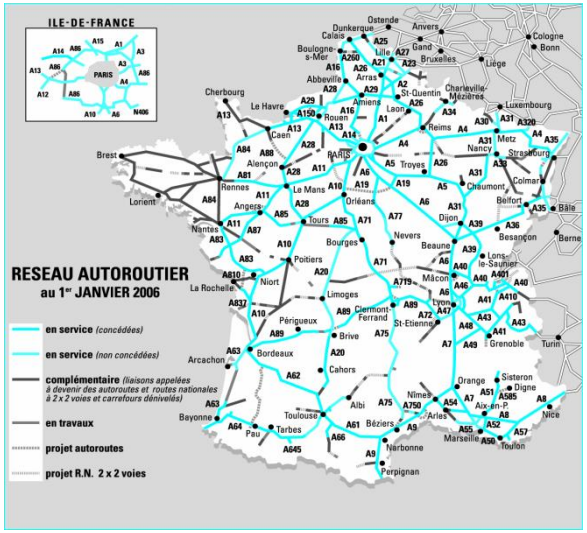
Backtrack sur P''

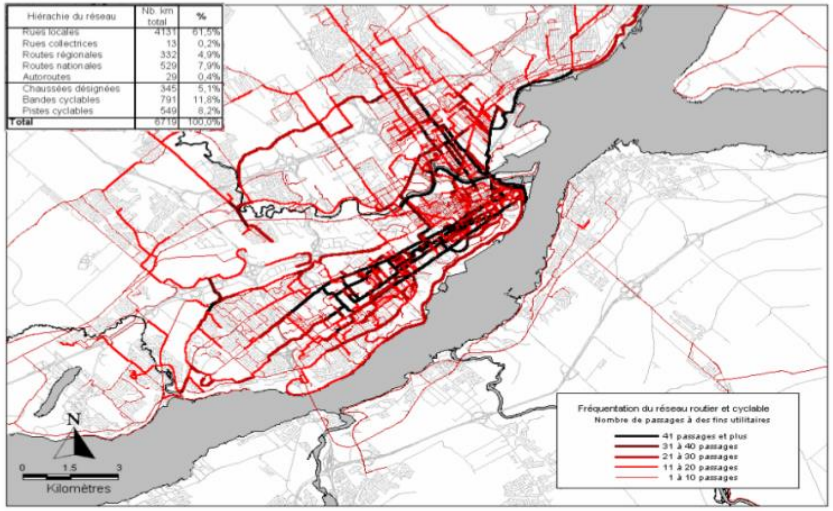
- P'' est globalement consistant
  - L'algorithme est glouton



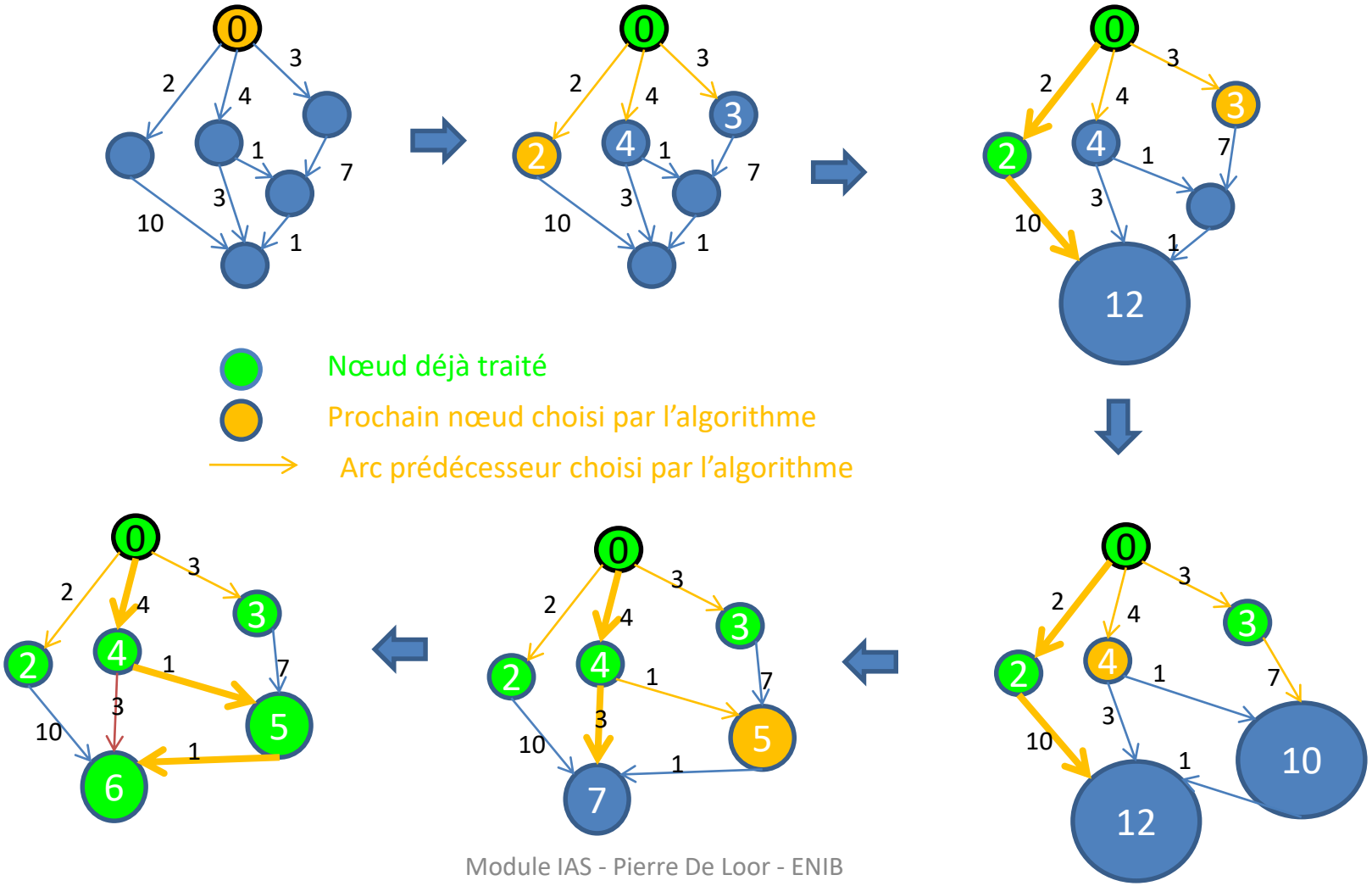
Question 3

- Comment fait-on un programme qui trouve le chemin le plus court ?





Un standard : L'algorithme de Djikstra (chemin le + court)

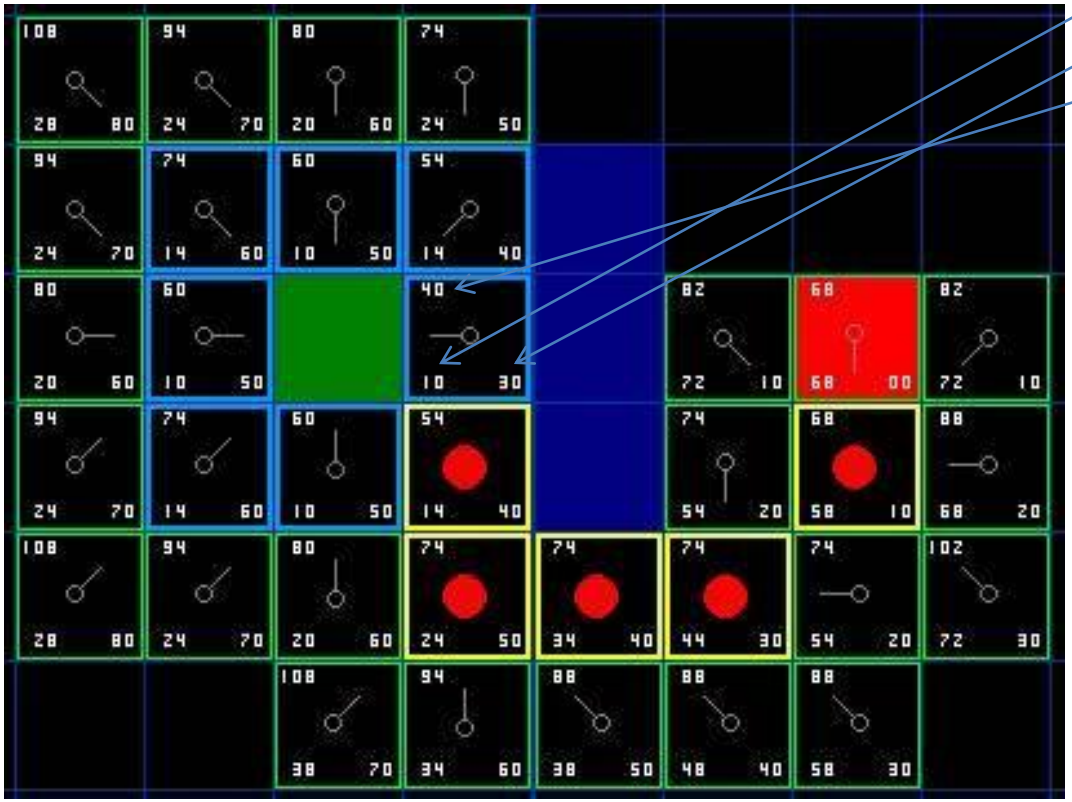


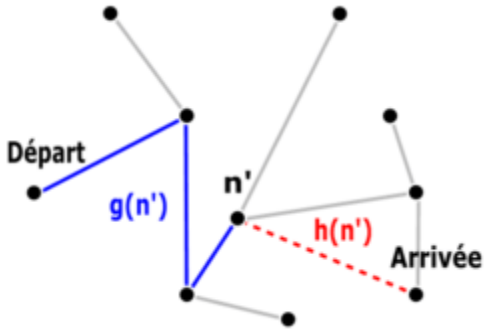
Notion d'heuristique

une **heuristique** est un algorithme qui fournit *rapidement* (en temps polynomial) une solution réalisable, pas nécessairement optimale, pour un problème d'optimisation (NP-Difficile).

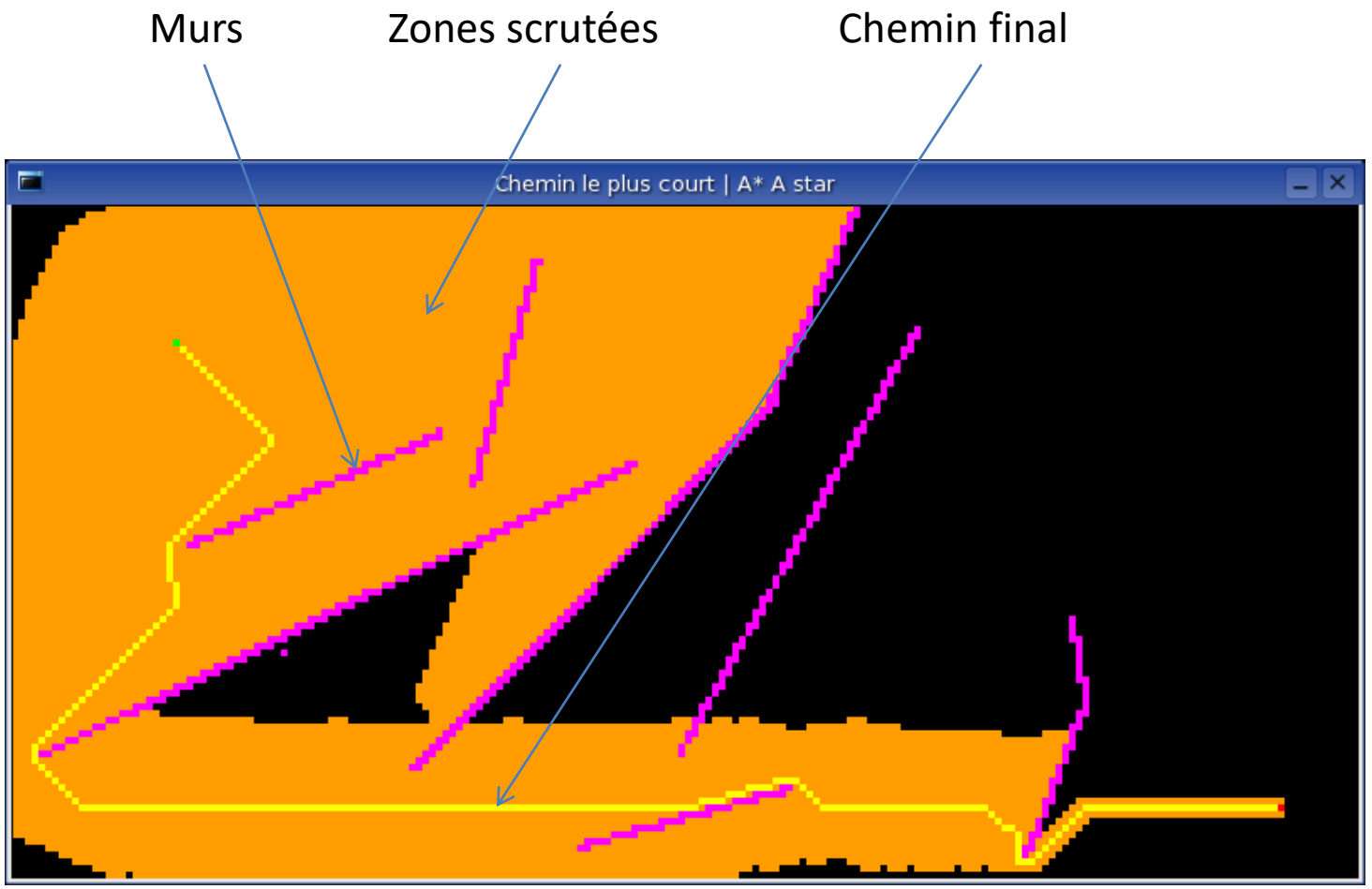
L'intérêt de l'heuristique étant que pour les problèmes NP-difficiles, les algorithmes exactes connus sont tous de complexité exponentielle et donc sans aucun intérêt **en pratique** (ni en théorie d'ailleurs).

Exemple : A*





A *


Murs
Zones scrutées
Chemin final
Chemin le plus court | A* A star

Autres exemples d'heuristiques

- Recuit simulé
- Méthodes tabous
- Algorithmes génétiques
- Seront Vues dans le module CRA

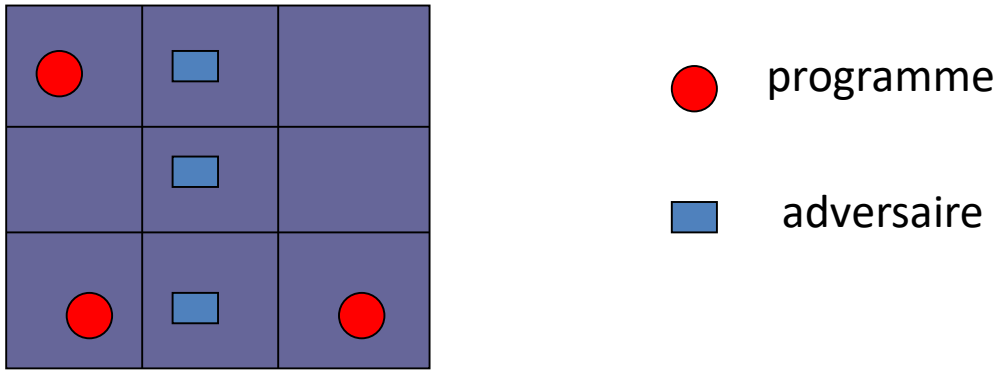
Question 4

- Comment fait-t-on un programme qui joue contre un adversaire ?

- La Théorie des jeux

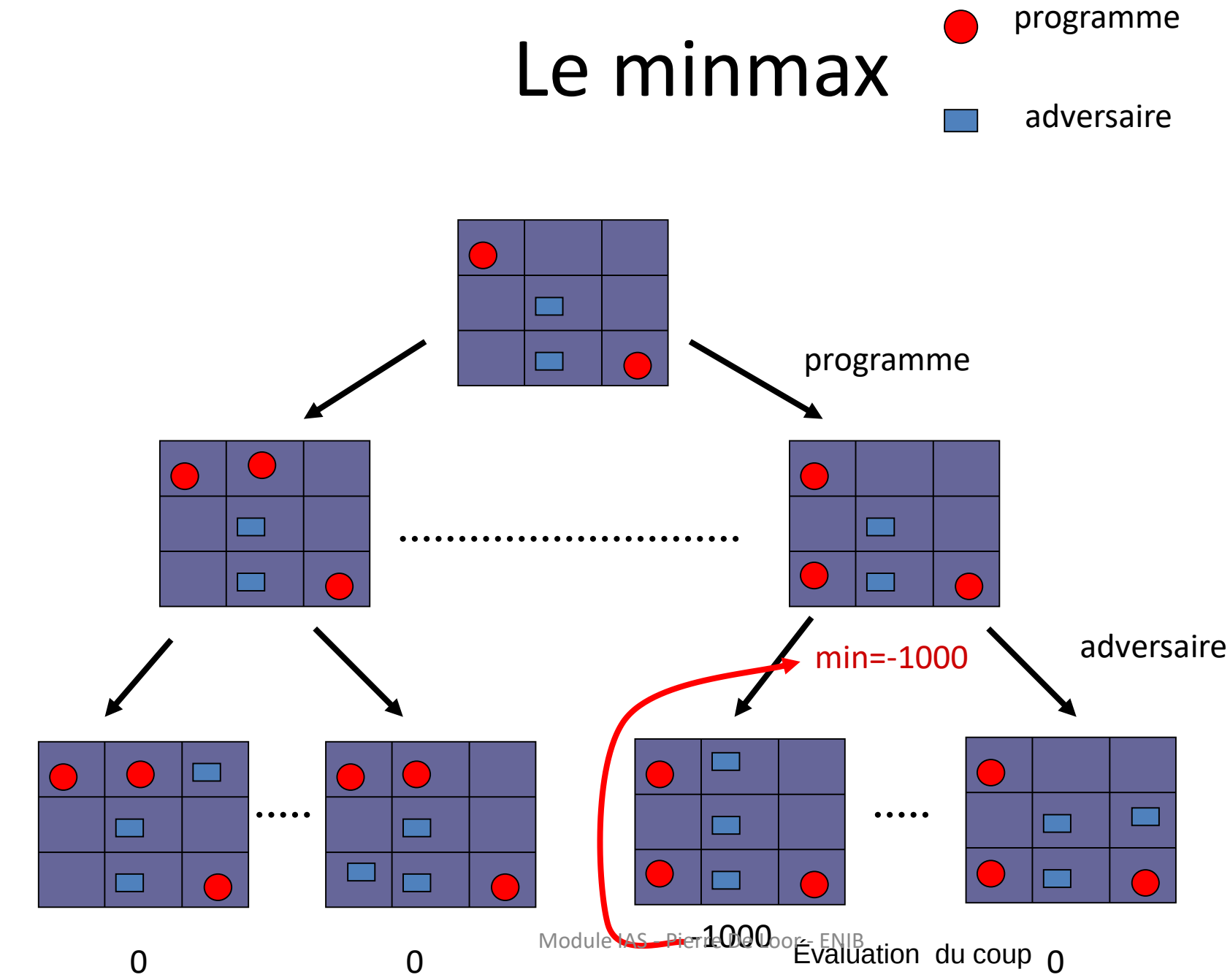
Jeux avec adversaire : minmax

- Von Neuman
- Evaluation numérique des coups possibles
- L'adversaire est « parfait » (joue au mieux)

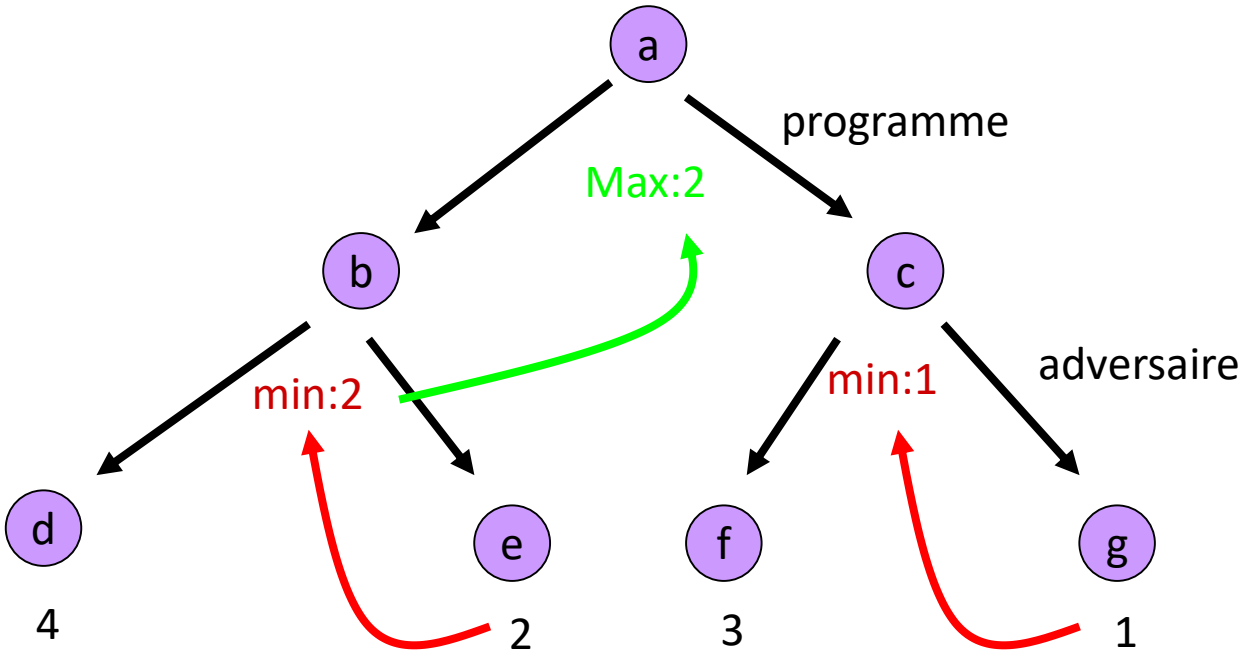


Exemple : pour le programme : -1000

Le minmax


programme
adversaire
programme
min=-1000
adversaire
1000
0
0
Évaluation du coup
0

Le minmax

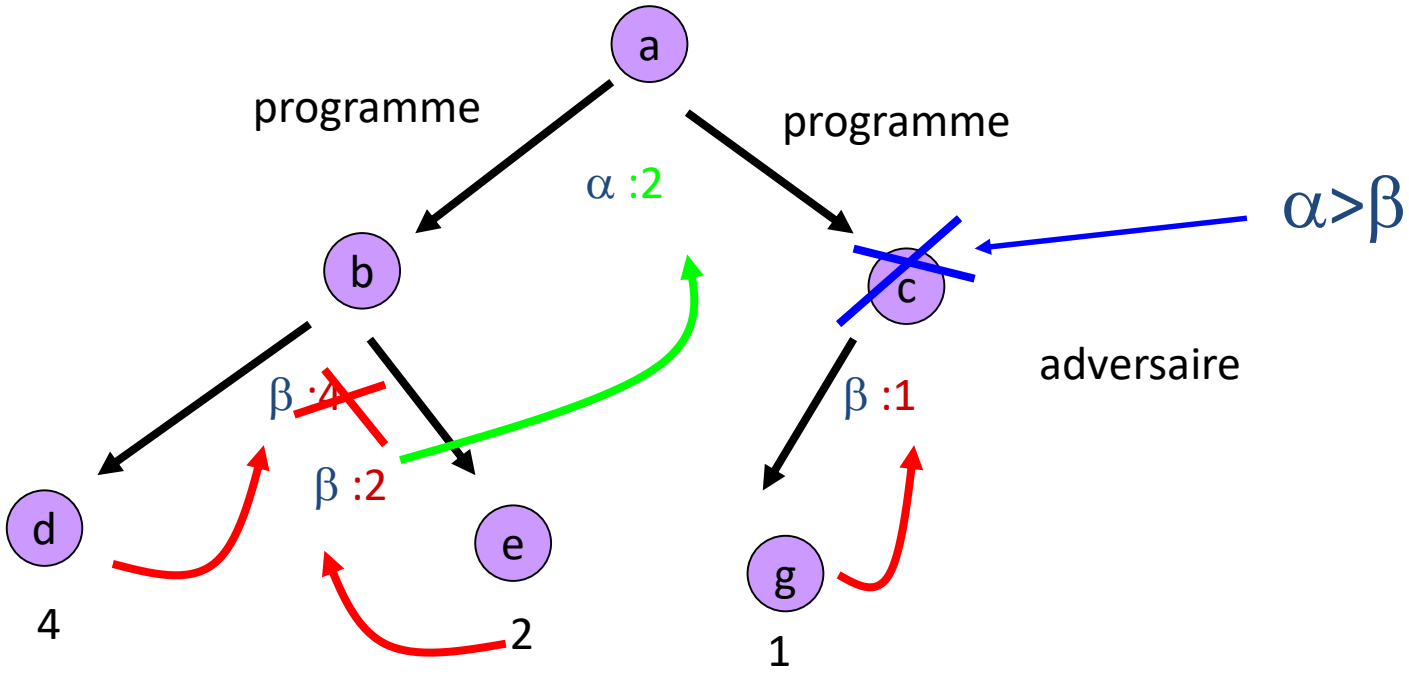


L'adversaire étant « parfait », je peux rapporter au mieux 2 points si je vais en b.
Généralisable à une profondeur quelconque

Amélioration : α-β

- Eviter de tracer toutes les branches du graphe.
- Passer par une recherche en profondeur
  - Remonter le min (β) et le Max (α) à chaque nouveau noeud
  - Comparer ces valeurs
  - Eliminer des nœuds avant de poursuivre

α-β



Le critère heuristique

- La fonction (le critère heuristique) qui définie le poids d'une situation est fondamentale.
- Exemple :
  - Jeux de dame : ratio du nb pions
  - Echecs :
    - Pondération des pièces selon leur « force »
    - Pondération des pièces selon leur position
  - Au GO :
    - Aucune fonction ne marche !!

Un challenge de l'IA pendant longtemps

- 8 coups à l'avance = 35 ^ 8 = 2 251 875 390 625 coups à calculer en moyenne (35 est le facteur de branche des échecs, nb coup possibles 10^{50})
- Deep Blue, 1996 (a battu Kasparov) : α-β de profondeur 6 à 12
- Deep Fritz, 2006 (tourne sur un PC) : 8 millions de coups à la seconde, profondeur moyenne de 16-17 coups
- Hydra : α-β amélioré (18 coups)

Résolution des jeux classiques

- Morpion : 10^3 , Résolu (à la main) : partie nulle si les joueurs jouent bien
- Puissance 4 : 10^{14} , Résolu en 1988 : le joueur qui commence a une stratégie gagnante
- Dames anglaises : 10^{20} , Résolu en 2007 : partie nulle si les joueurs jouent bien
- Échecs : 10^{50}, Non résolu. Les programmes sont meilleurs que les meilleurs humains
- Go : 10^{171}, Non résolu. Les programmes sont bien plus faibles que les meilleurs humains

Les méthodes de Monte Carlo au GO

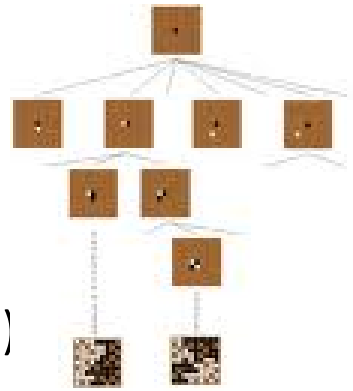
- Programmes pour un 6x6 et récemment un 13x13 (grille standart 19x19)
- Très récemment (2009), les méthodes de monté-carlo sont capables de battre certains maîtres.
- La méthode de Monté Carlo n'est pas une résolution, c'est une approche de simulation statistique qui évite le problème majeur du GO : il n'y a pas moyen d'évaluer la « qualité » d'une situation
- MoGo : un algorithme Français

UCB : Où l'estimation d'un coup

- Si il est difficile évaluer la qualité d'un coup, on va en faire une probabilité.
- L'algorithme UCB (Upper Confidence Bounds) : être optimiste devant l'incertain
- Plus on joue, plus on affine l'intervalle de confiance
- On est optimiste : c'est la borne max de cet intervalle qui est considérée comme la valeur (pour sélectionner le prochain bras à jouer)

Méthode de Monté-Carlo au go

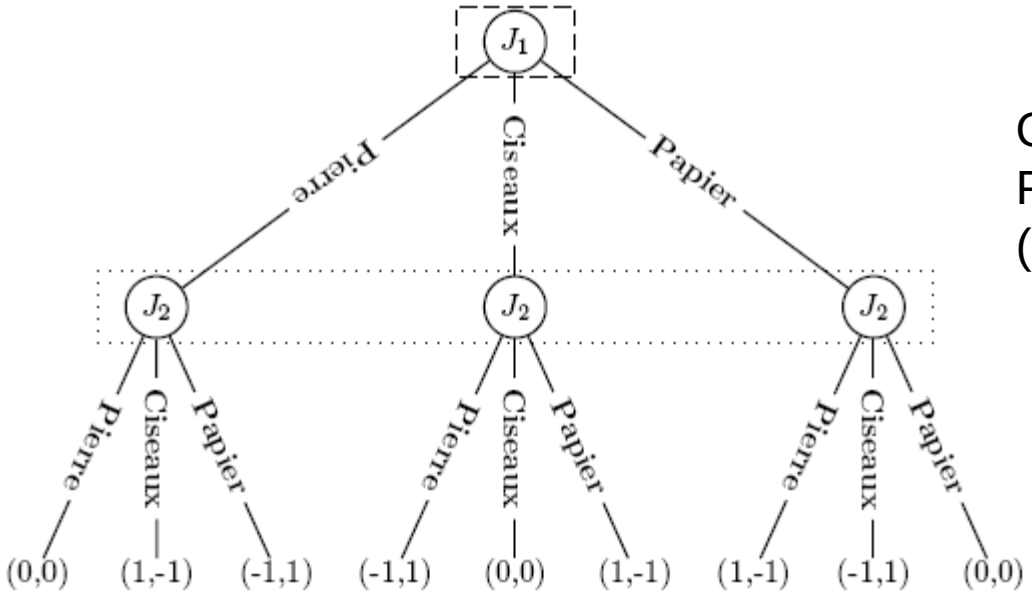
- Pour évaluer une position, finissons la partie !
- Choisir un coup C au hasard
- Faire N fois :
  - Faire **jusqu'à la fin de la partie « virtuelle »**
    - Choisir un coup suivant au hasard (ou selon la règle UCB)



- La valeur de C est la moyenne des parties gagnées sur les N parties évaluées.
- L'arbre est construit itérativement, la règle UCB « creuse » les meilleurs coups possibles

Stratégie et tactique

Etude théorique des jeux



Que vaut-il mieux jouer
Pour gagner plus
(ou perdre moins) ?

J_1 / J_2	Pierre	Ciseaux	Papier
Pierre	0 , 0	1 , -1	-1 , 1
Ciseaux	-1 , 1	0 , 0	1 , -1
Papier	1 , -1	- 1 , 1	0 , 0

J_1 / J_2	Pierre	Ciseaux	Papier
Pierre	0	1	-1
Ciseaux	-1	0	1
Papier	1	- 1	0

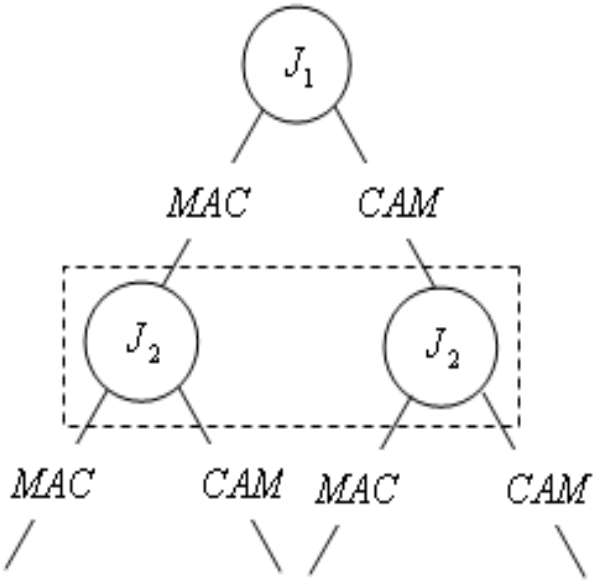
Jeux à somme nulle → Demi-matrice des gains

Jeux non strictement compétitifs

- L'entreprise J2 possède des MAC
- L'entreprise J1 possède des CAM
- Les deux entreprises ont intérêt à uniformiser leur parc informatique
- Changer de parc à un coût
- Quand l'une gagne, l'autre

Ne perd pas tout

J_1 / J_2	*CAM*	*MAC*
CAM	600 , 200	100 , 100
MAC	100 , 100	200 , 600



Jeu à « tactique prudente », équilibre de Nash

J_1 / J_2	b_1	b_2	b_3	b_4
a_1	5 , 5	6 , 4	0 , 10	4 , 6
a_2	1 , 9	7 , 3	5 , 5	6 , 4
a_3	6 , 4	7 , 3	7 , 3	8 , 1
a_4	4 , 6	8 , 1	0 , 10	2 , 8
a_5	3 , 7	5 , 5	9 , 0	0 , 10

Si J1 choisi a1, J2 a intérêt à jouer b3 et J1 gagne 0
Si J1 choisi a2, J2 à interêt à jouer b1 et J1 gagne 1
Etc …

Le choix prudent de J1 est donc a3 (il gagne au moins 6)
De même le choix prudent de J2 est b1 (gagne au moins 4)
Le choix a3/b1 est un équilibre de Nash : *aucun joueur ne modifie sa stratégie sans affaiblir sa position personnelle*

Equilibres de Nash

- Pour l'exemple des firmes J1 et J2, il y a deux points d'équilibres de Nash (CAM/CAM et MAC/MAC)
- Quelle solution sortira du jeux ??
- Impossible à dire ici, mais le fait est qu'il existe des stratégies « non perdantes »

J_1 / J_2	*CAM*	*MAC*
CAM	600 , 200	100 , 100
MAC	100 , 100	200 , 600

Ici pas d'équilibre de Nash

- La tactique prudente du joueur ligne est la ligne 2
- La tactique prudente du joueur colonne est la colonne 1

2	0
1	3

Les deux stratégies prudentes aboutissent à un point qui pourrait être amélioré pour le joueur ligne puisque si je joueur colonne prend la colonne 1, le joueur ligne a tout intérêt à prendre la ligne 1

Jeux répétitifs

- La guerre des sexes :
- L'homme préfère voir un film
- La femme un documentaire
- Mais ils préfèrent avant tout être ensemble

J_1 / J_2	*Doc*	*Com*
Doc	2 , 3	1, 1
Com	1 , 1	3 , 2

- Deux équilibres de Nash, impossible de prévoir l'issue du jeu
- Mais il est possible de mettre en place des stratégies si on répète ce jeux N fois (aller N fois de suite au cinéma)
- La stratégie au nième coup dépend du n-1ème coup
- C'est la théorie des jeux répétitifs

FIN DU PREMIER COURS

Chapitre 2

MACHINE LEARNING

Plan du chapitre 2

- Apprentissage de concepts
  - Concepts de l'apprentissage
  - Espace des versions
  - Arbres de décisions
- Apprentissage par renforcement
  - Le Q-Learning
  - Les systèmes de classeurs

Definition

A program learns a task (T) from experiences (E) if performance (P) to realize (T) improves after E. (Mitchell97)

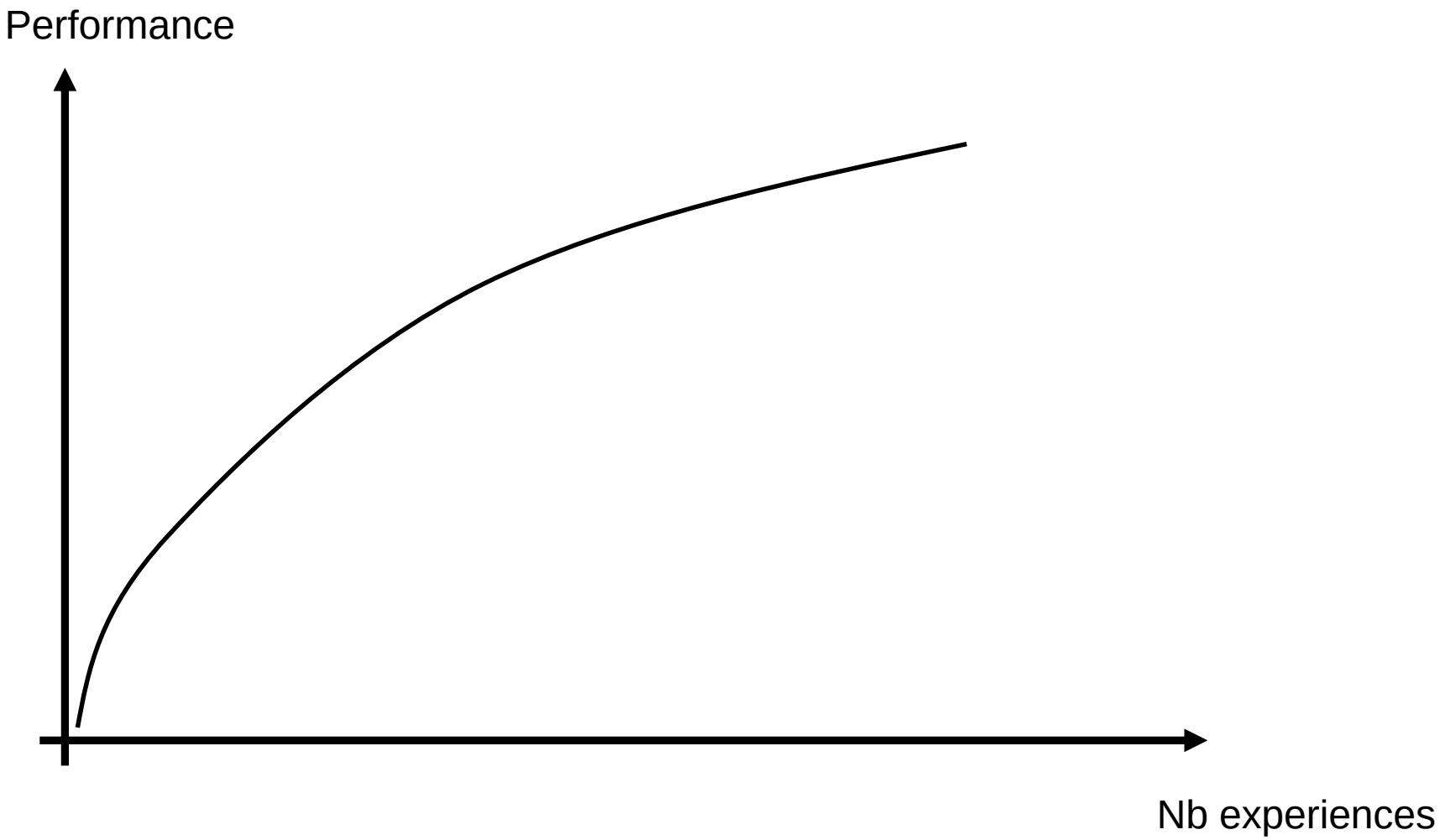
Il existe de nombreux algorithmes d'apprentissage car il existe de nombreux type de

Tâches(T)

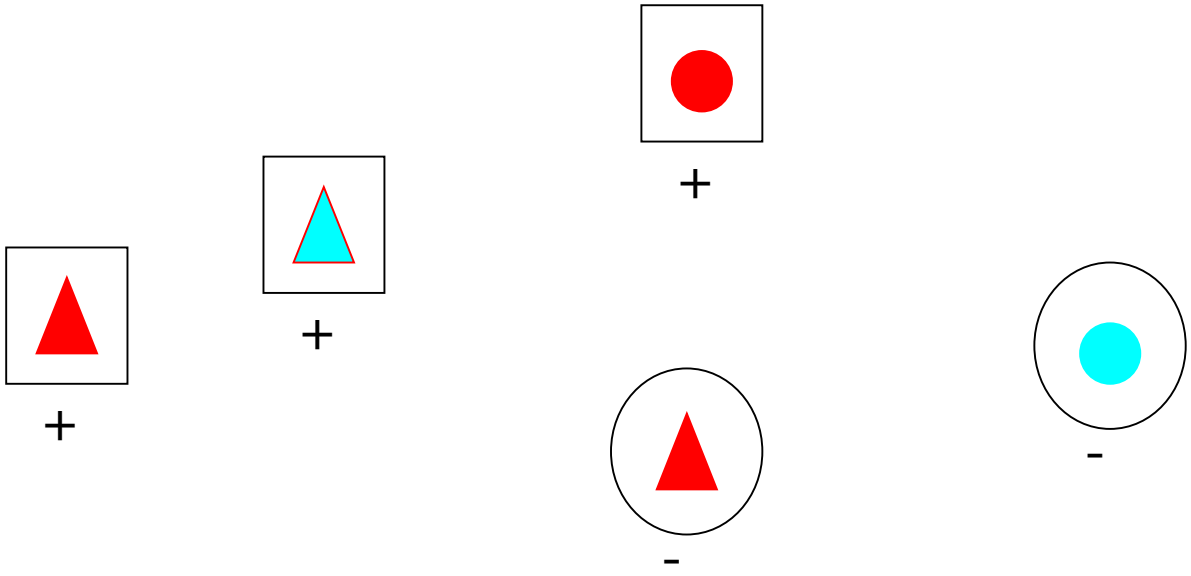
Expériences (protocol) (E)

Performances (P)

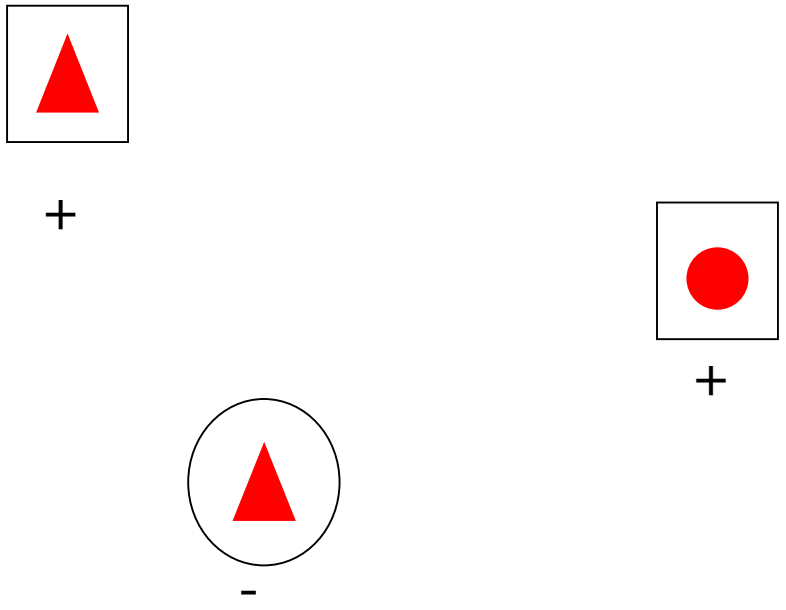
Première courbe



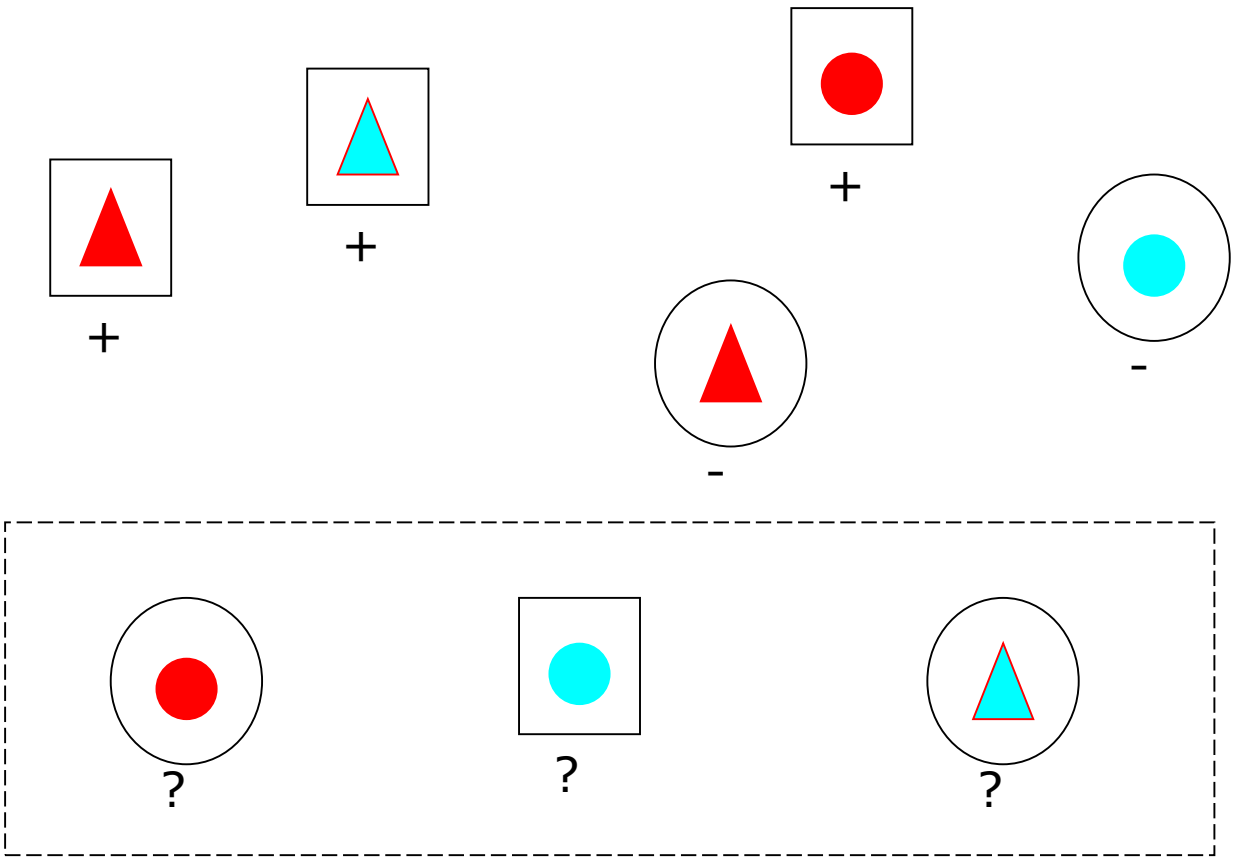
Un exemple



Reconnaitre

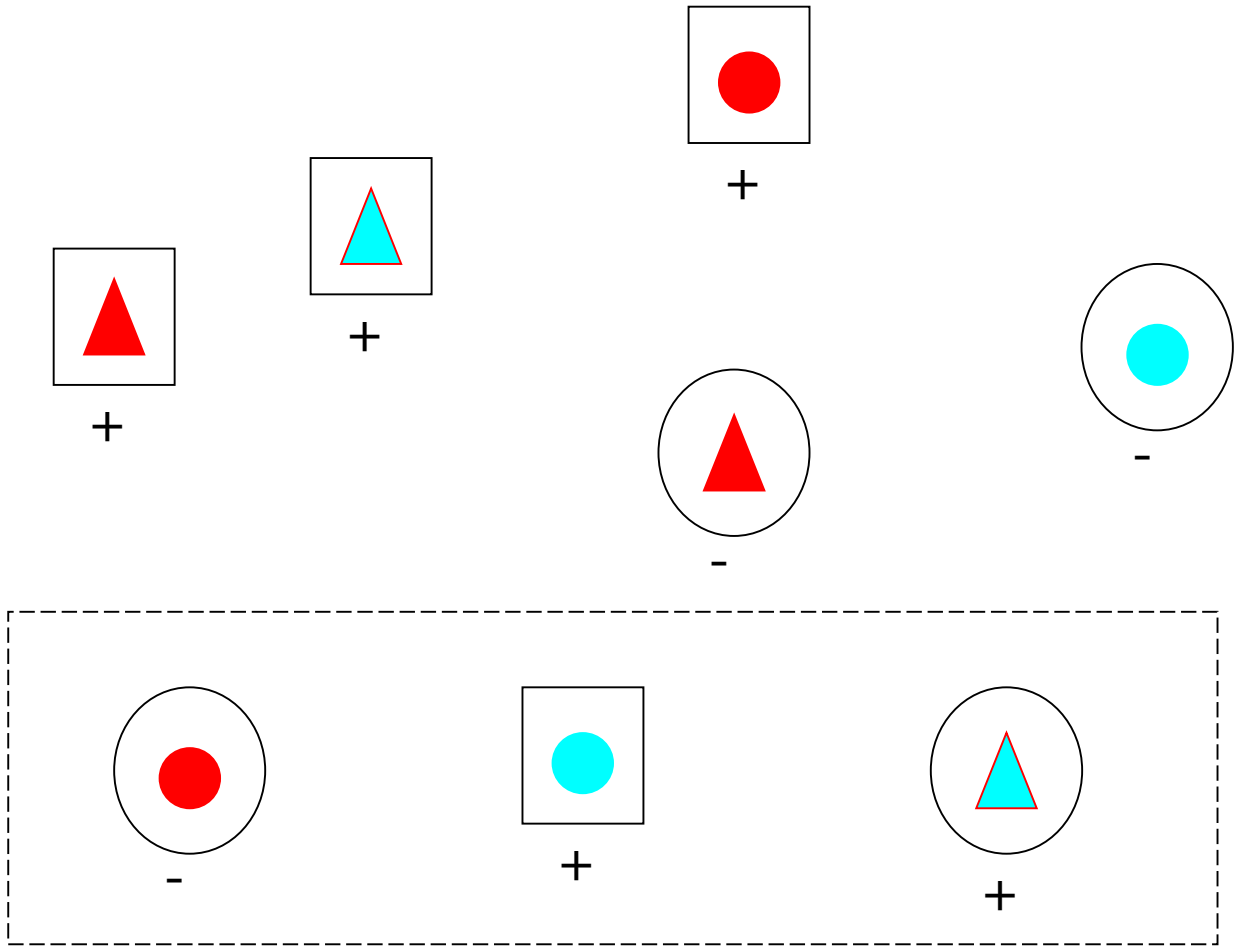


Généraliser

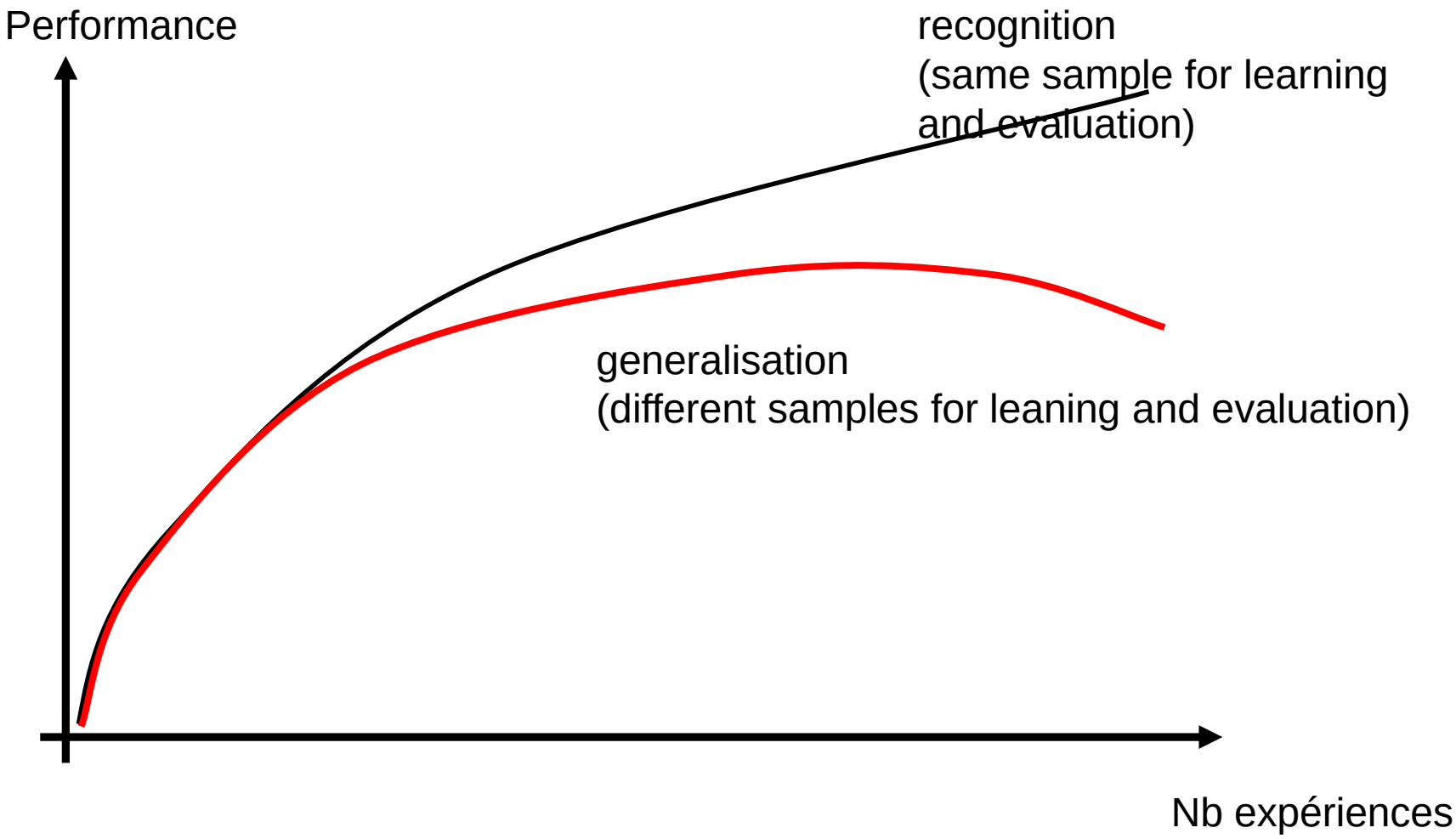


Les cadres carrés sont positifs ?

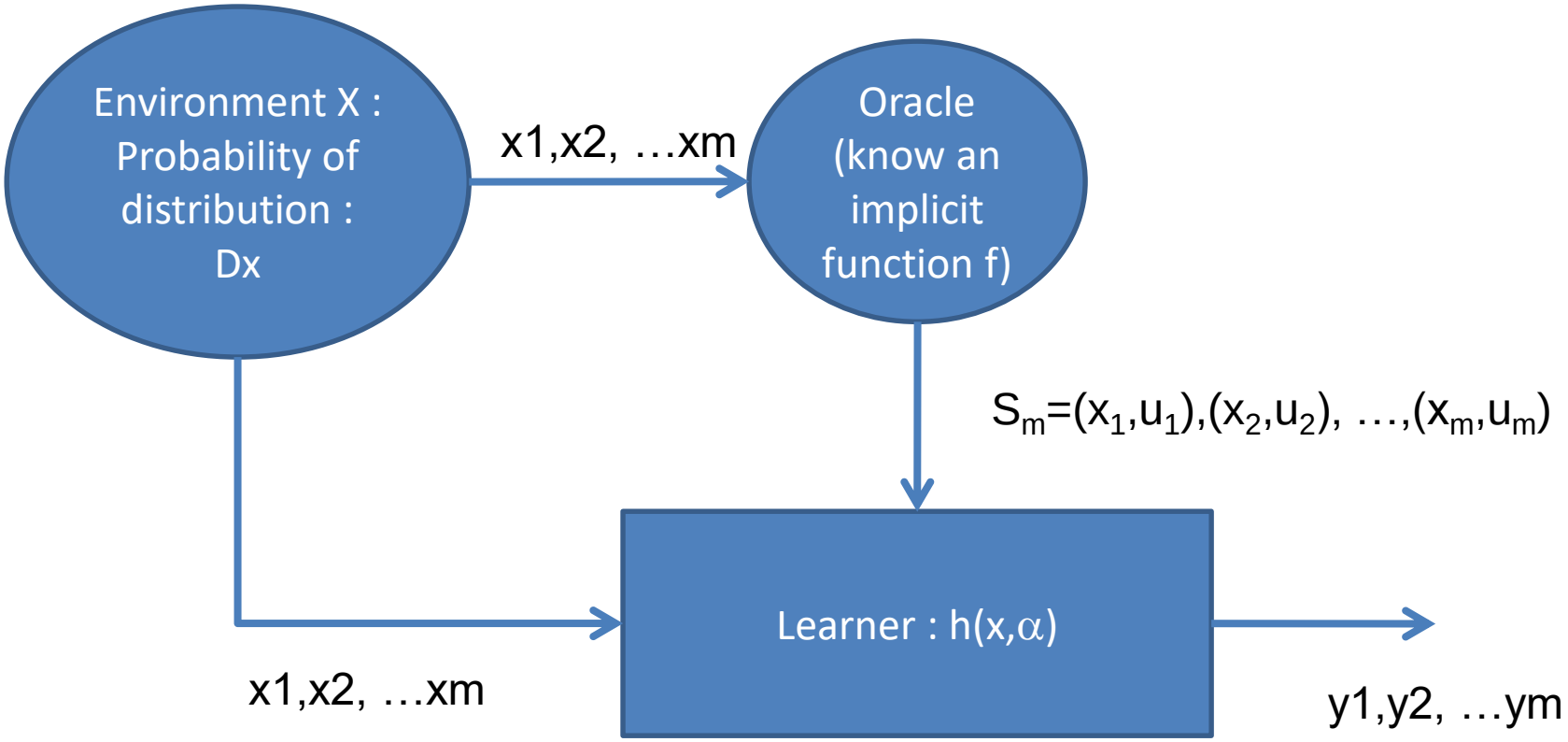
Problème !



Deuxième courbe



Induction

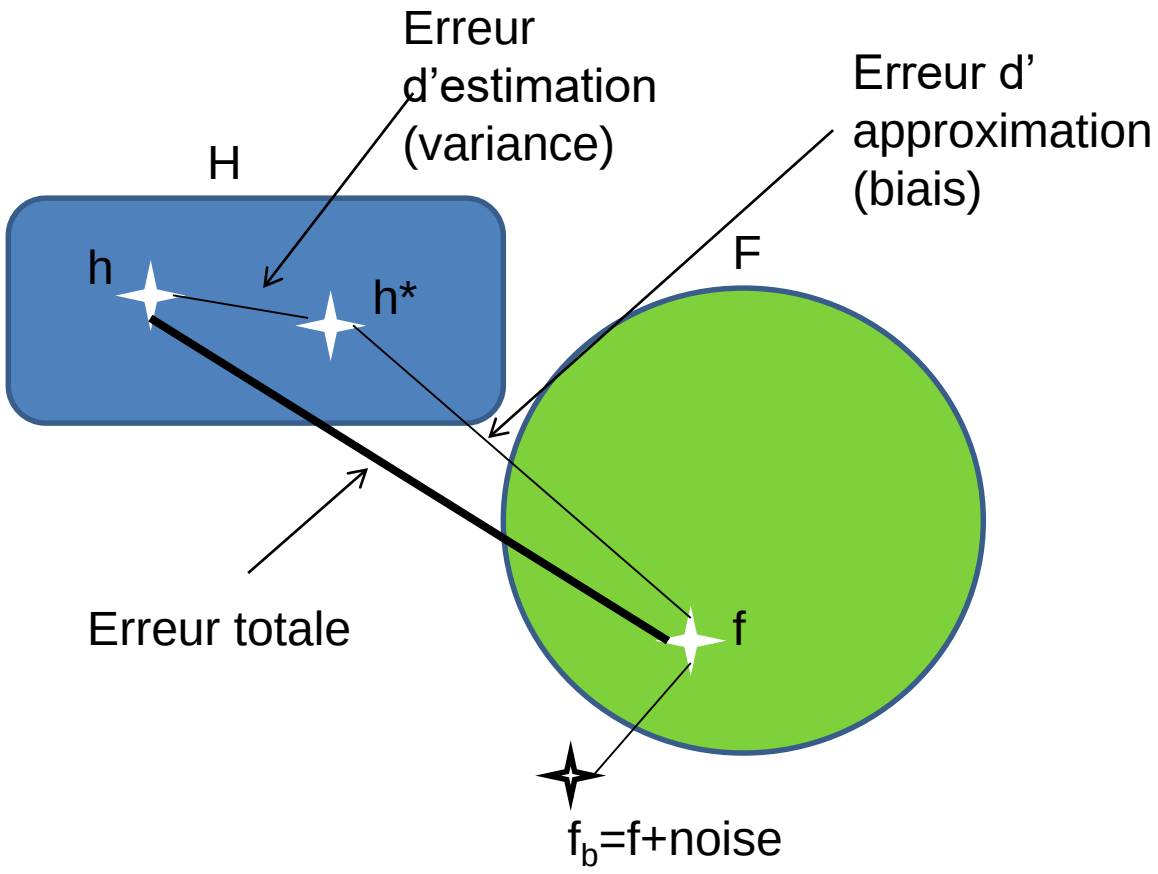


Le problème est de trouver une hypothèse h produisant les mêmes résultats que f (qui est inconnue), à partir d'un nombre limité d'exemples (x_i,u_i). h est une approximation de f.

Problèmes

- L’espace de représentation de f est inconnu et l’espace de représentation de h peut être différent (biais ou erreur de représentation)
- L’hypothèse optimal h* (la moins mauvaise hypothèse) n’est pas forcément découverte par l’algorithme (variance ou erreur d’estimation
- Les exemples peuvent être entâchés d’erreurs (bruit)

Compromis Biais/Variance



Minimiser le biais accroit la variance

Grandir H pour se rapprocher de f accroit la possibilité d'une erreur d'estimation.

Oui mais le biais est essentiel …

Réduire le biais est l'objectif principal de la plupart des algorithmes d'apprentissage

- Apprendre signifie que le modèle est inconnu
- Généraliser signifie « confondre » ou « regrouper » différents cas.
- Donc le biais est indispensable.

Protocoles

- Apprentissage supervisé ou en-ligne.
- Prediction (Cas particulier - météo) versus identification (Loi générale).
- Apprentissage actif: Le système propose une hypothèse ou une expérience.

Apprentissage de concepts

- Exemple : faire de la planche à voile

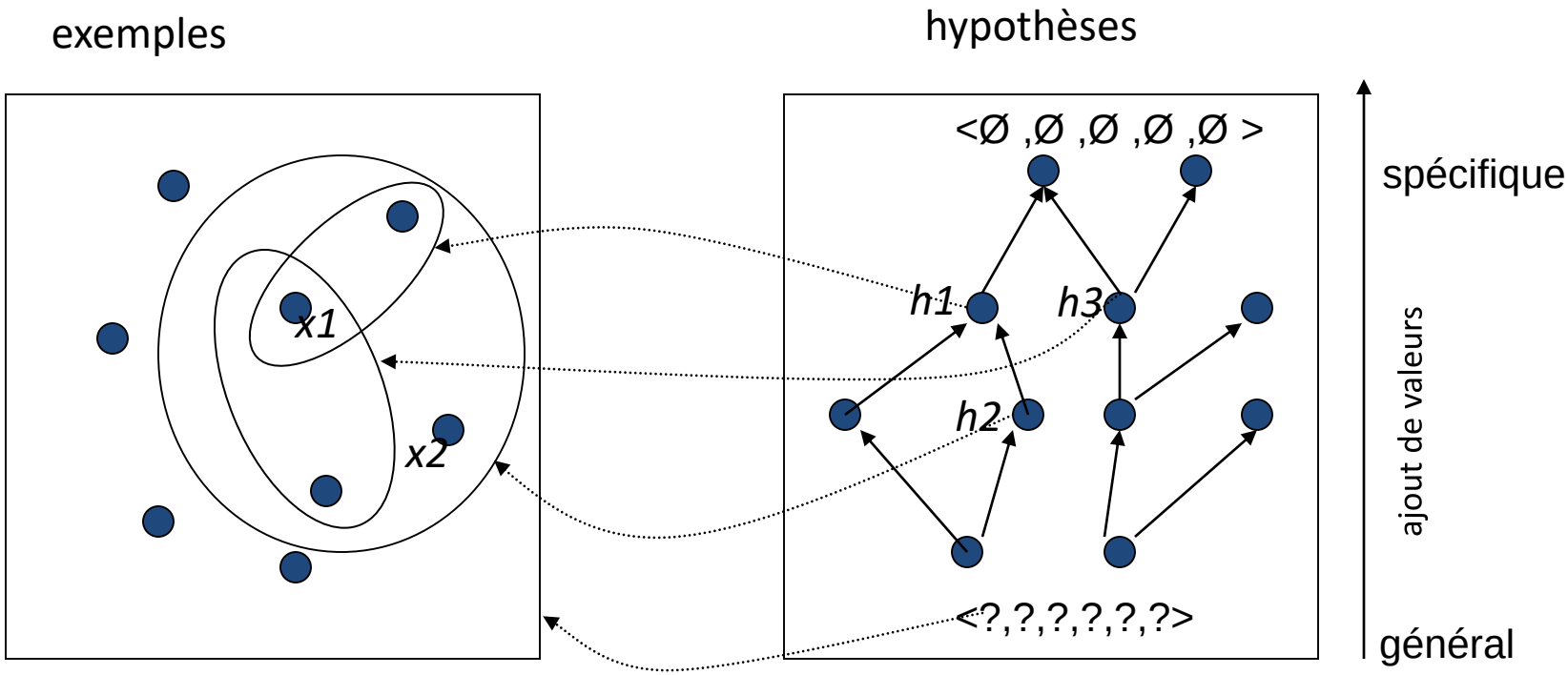
Ex	ciel	temp	humidité	vent	eau	prévision	planche
1	soleil	chaud	normale	fort	tiède	stable	oui (positif)
2	soleil	chaud	élevée	fort	tiède	stable	oui
3	pluie	froid	élevée	fort	tiède	variable	non (négatif)
4	soleil	chaud	élevée	fort	fraîche	variable	oui

Hypothèse d 'une fonction représentant le concept :
conjonction des valeurs des attributs.

Différentes hypothèses

- Quand il fait chaud et que la température est élevée
  - <?, chaud, élevée, ?, ?, ?>
- Hypothèse la plus générale (dans n 'importe quelle condition)
  - <?, ?, ?, ?, ?, ?>
- Hypothèse la plus restrictive (jamais)
  - <∅, ∅, ∅, ∅, ∅, ∅>
- Satisfaction d 'un exemple
  - une hypothèse h satisfait un exemple x si $x \subset h$
  - par commodité on dira aussi h(x) = positif

Classement « Spécifique - Général »



x1=<soleil,chaud,élevé,fort,fraîche,stable>

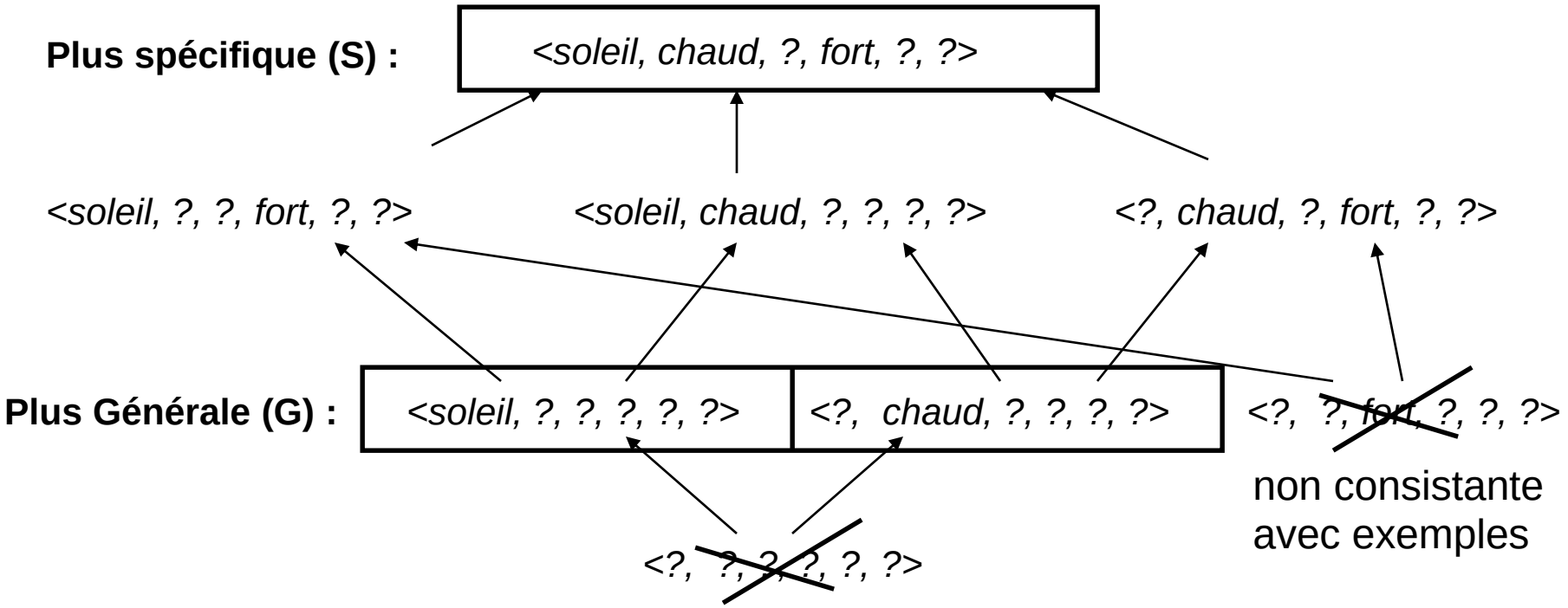
x2=<soleil,chaud,élevé,faible,tiède,stable>

h1=<soleil,?,?,fort,?,?>

h2=<soleil,?,?,?,?,?>

h3=<soleil,?,?,?,fraîche,?>

Représentation compacte de l ’espace des versions



3 hypothèses en représentent 6

Algorithme « Candidate-Elimination »

- Départ : toutes les hypothèses en 2

Plus spécifique (S) : <Ø,Ø,Ø,Ø,Ø,Ø>

Plus générale (G) : <?,?,?,?,?,?>

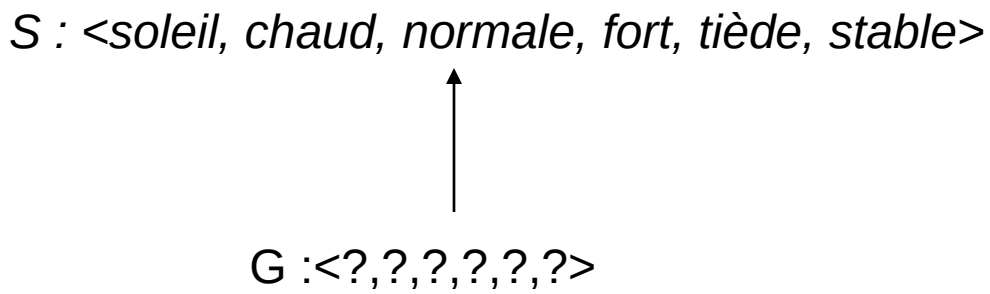
- Premier exemple :

<soleil, chaud, normale, fort, tiède, stable> ***positif***

S trop spécifique

Algorithme « Candidate-Elimination »

- Transformation :



- Deuxième exemple :

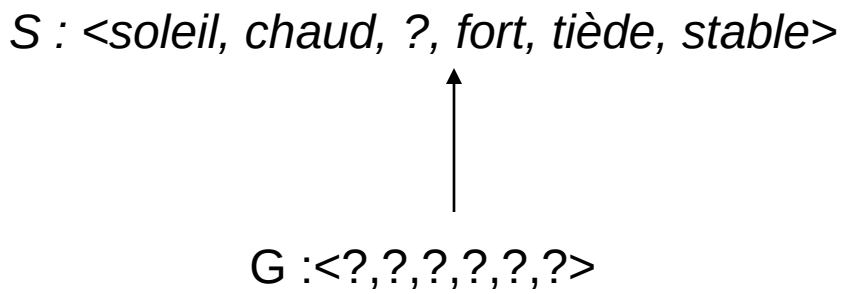
  <soleil, chaud, élevée, fort, tiède, stable> ***positif***

  S trop spécifique

  à cause de « normale»

Algorithme « Candidate-Elimination »

- Transformation



- Troisième exemple :

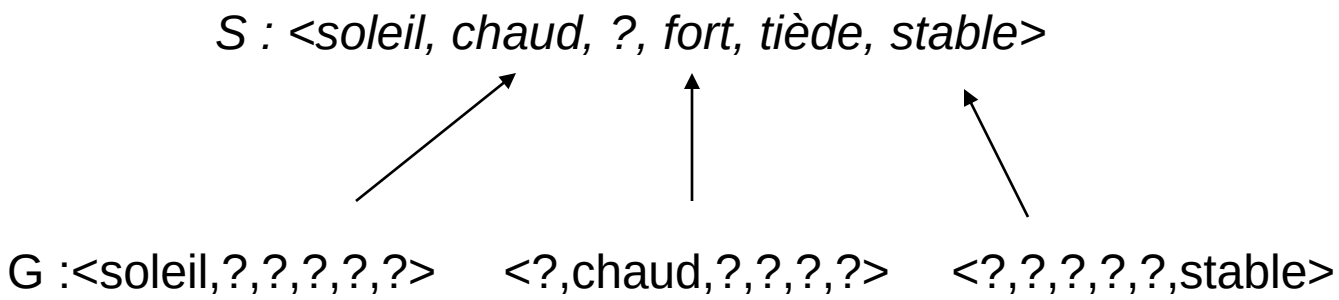
  <pluie, froid, élevée, fort, tiède, variable> ***négatif***

  G trop générale

  à cause de : pluie ou froid ou variable (ou les 3)

Algorithme « Candidate-Elimination »

- Transformation



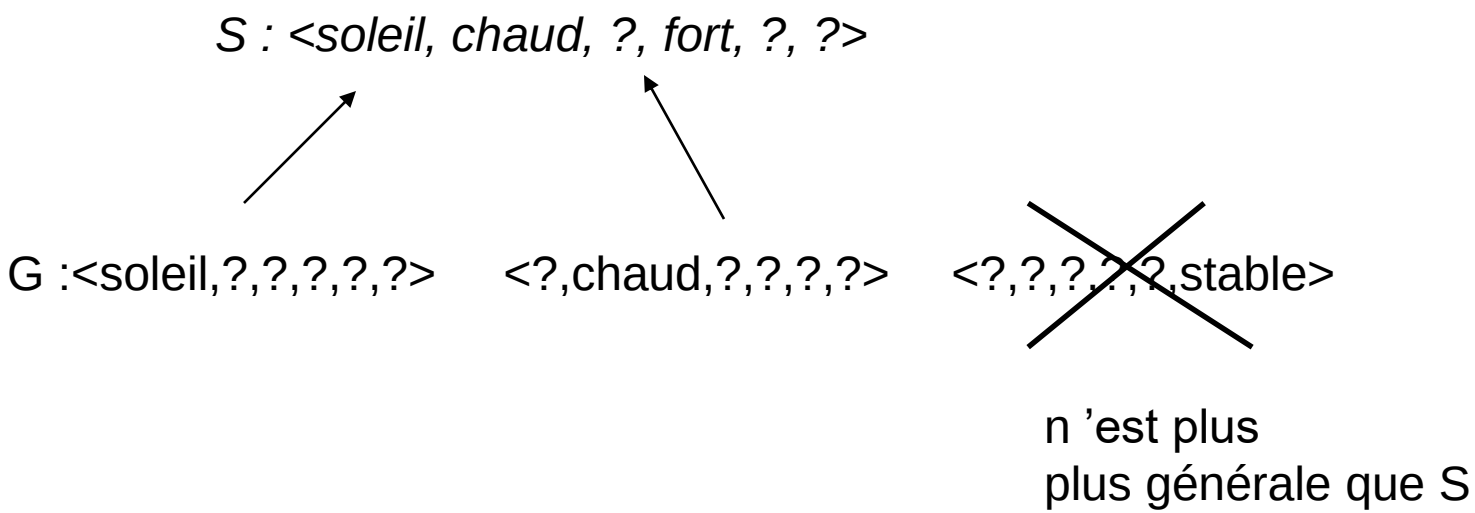
- Quatrième exemple :

  <soleil, chaud, élevée, fort, fraîche, variable> ***positif***

  S trop spécifique à cause de « tiède » et « stable »

Algorithme « Candidate-Elimination »

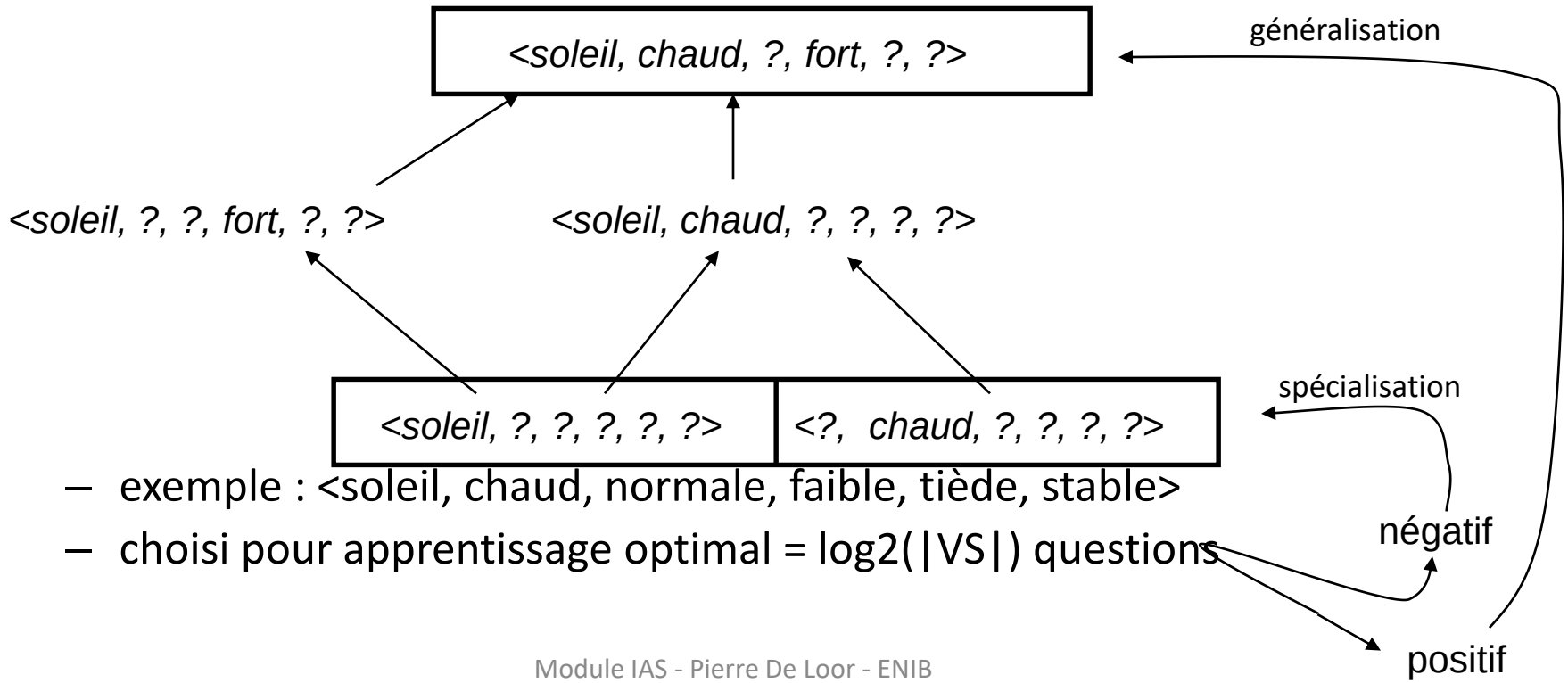
- Transformation



- Représente 6 hypothèses consistantes

Algorithme « Candidate-Elimination »

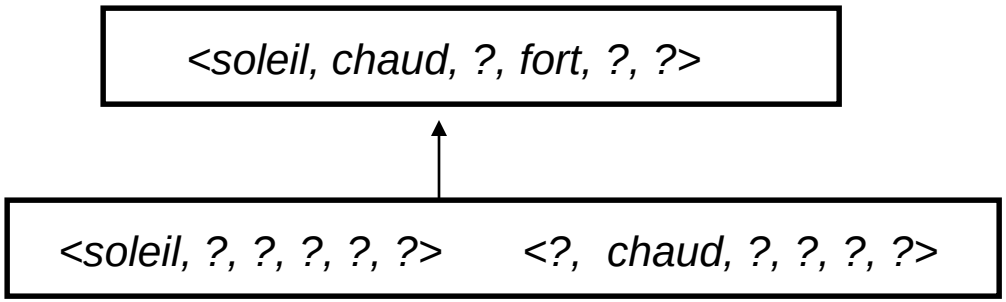
- Poser une question pour apprendre
  - qui puisse être positive ou négative par rapport aux hypothèses actuelles



  - exemple : <soleil, chaud, normale, faible, tiède, stable>
  - choisi pour apprentissage optimal = log2(|VS|) questions

Algorithme « Candidate-Elimination »

- Exploiter l ’apprentissage



<soleil, chaud, normale, fort, fraîche, variable > : ***positif***

<pluie, froid, normale, faible, tiède, stable > : ***négatif***

<soleil, chaud, normale, faible, tiède, stable > : ***3 positives / 3 négatives ??***

<soleil, froid, normale, fort, tiède, stable > : ***2 positives / 4 négatives : plutôt négatif ?***

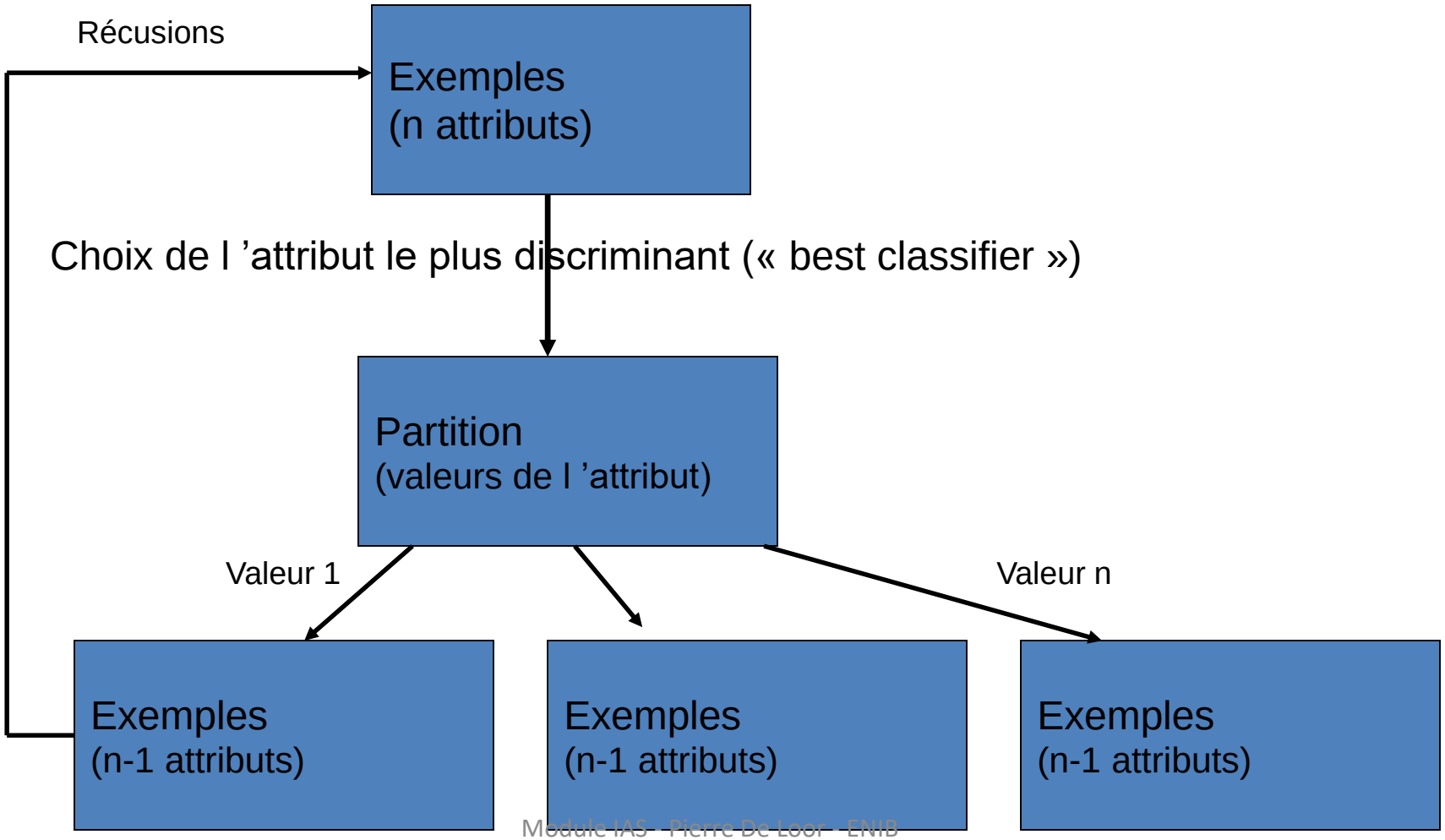
Notion de complexité

- 6 variables possédant 2 ou 3 valeurs
  - 96 combinaisons (973 hypothèses possibles)
- introduction des disjonctions et négations
  - $2^{96} \approx 10^{28}$ concepts possibles
- algo avec disjonctions ?
  - Impraticable
  - ne reclasse que les exemples donnés
  - toute question voit nb positifs = nb négatifs
- biais : $973 \sim 10^{28}$

Apprendre un arbre

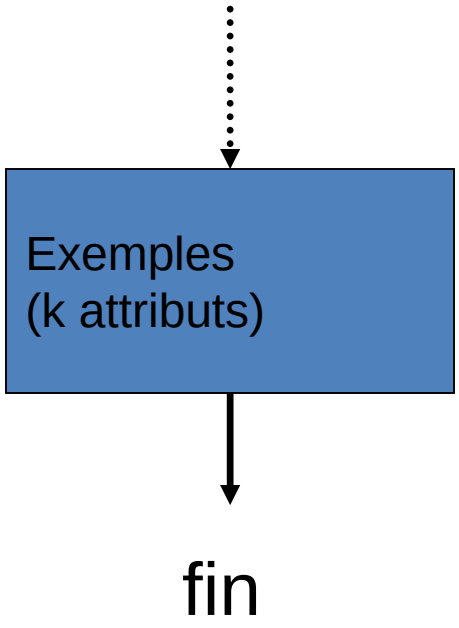
- Qui permet de classer les exemples
- En acceptant des erreurs dans les exemples
- Acceptant des exemples incomplets
- Qui soit le plus petit possible
- Qui soit le plus discriminant possible
  - Choix de l ’ordre de test des variables
  - algorithme ID3 : notion d ’entropie

ID3

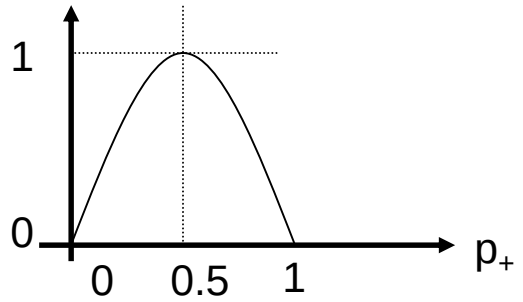

Récusions
Exemples
(n attributs)
Choix de l ’attribut le plus discriminant (« best classifier »)
Partition
(valeurs de l ’attribut)
Valeur 1
Valeur n
Exemples
(n-1 attributs)
Exemples
(n-1 attributs)
Exemples
(n-1 attributs)

ID3

- Fin d ’une branche

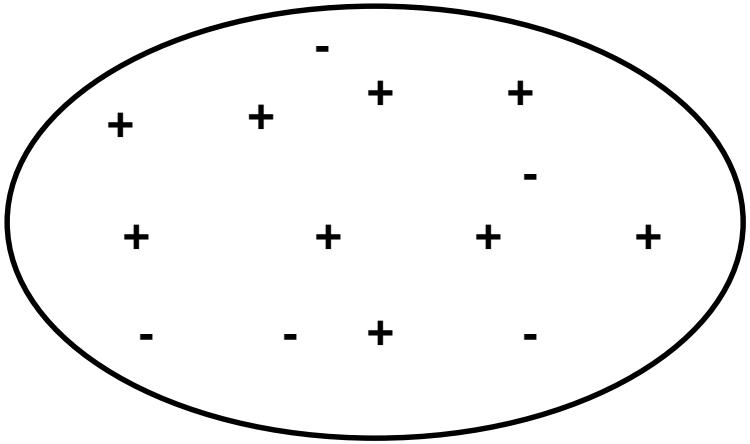


Entropie



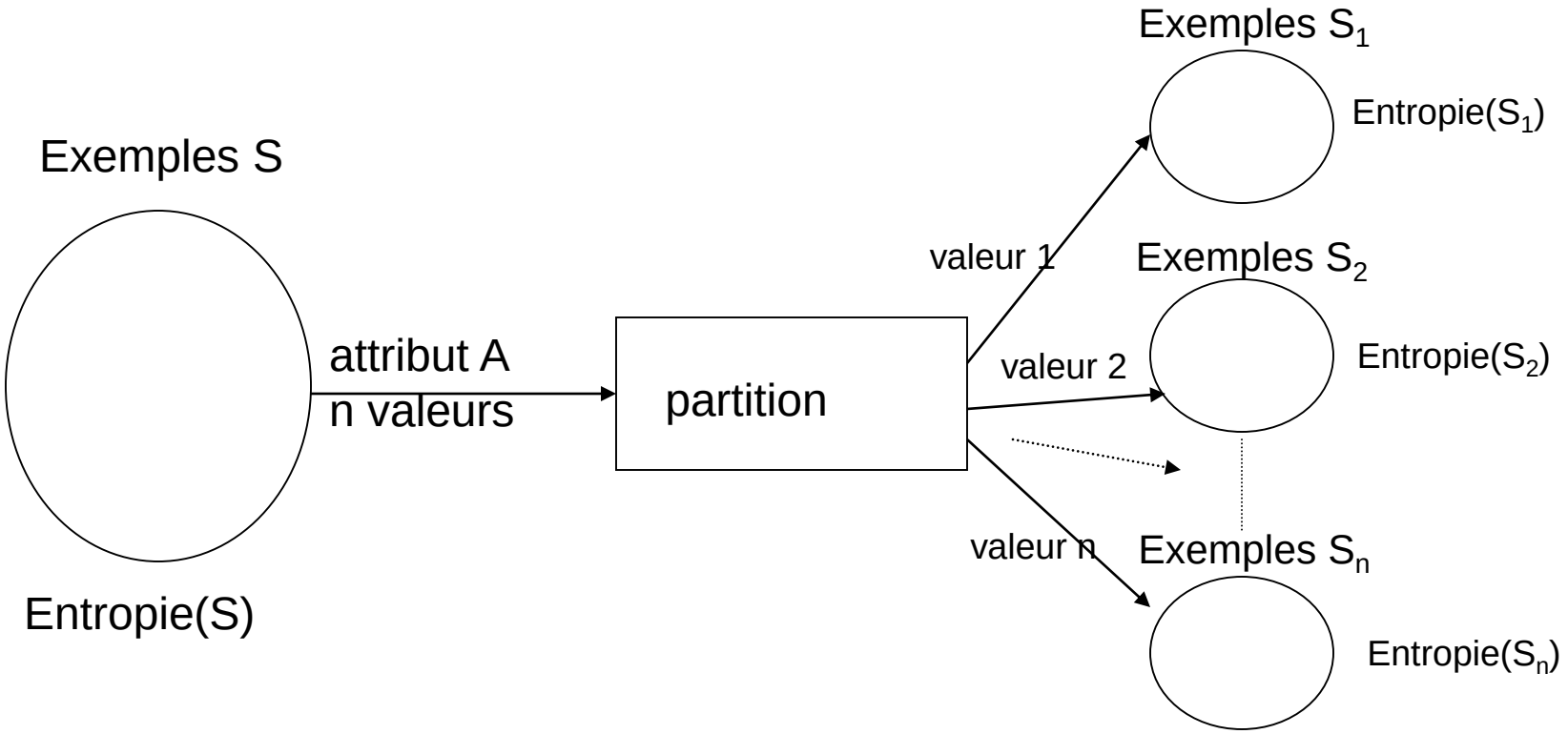
- Si il y a autant de + que de -
  - Entropie(S) = 1
- Si tous les exemples ont la même valeur
  - Entropie(S) = 0
- Généralisation à c valeurs possibles
  - Entropie(S) = $\sum_{i=1}^{c} -p_i \log2(p_i)$

Entropie : exemple



Entropie : $-(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940$

Gain d ’entropie dû à un attribut A

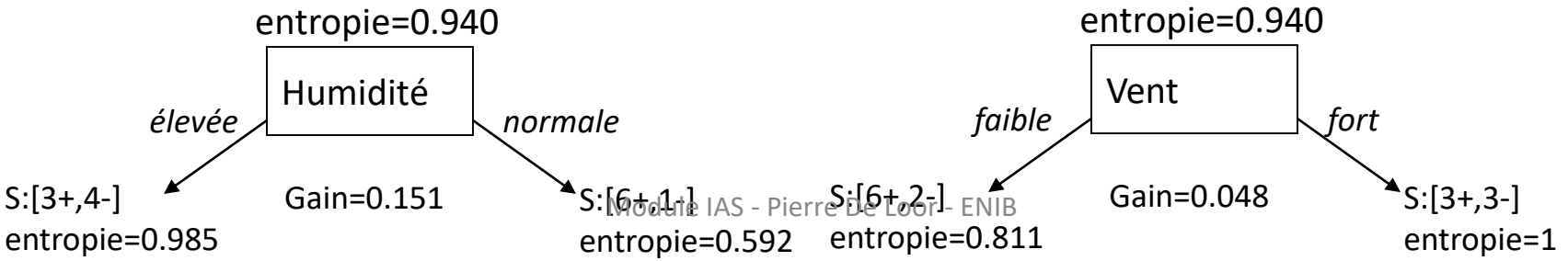


$$Gain(S,A) = Entropie(S) - \sum_{v=1}^{n} \frac{|S_v|}{|S|} \times Entropie(S_v)$$

Gain d ’entropie : exemple

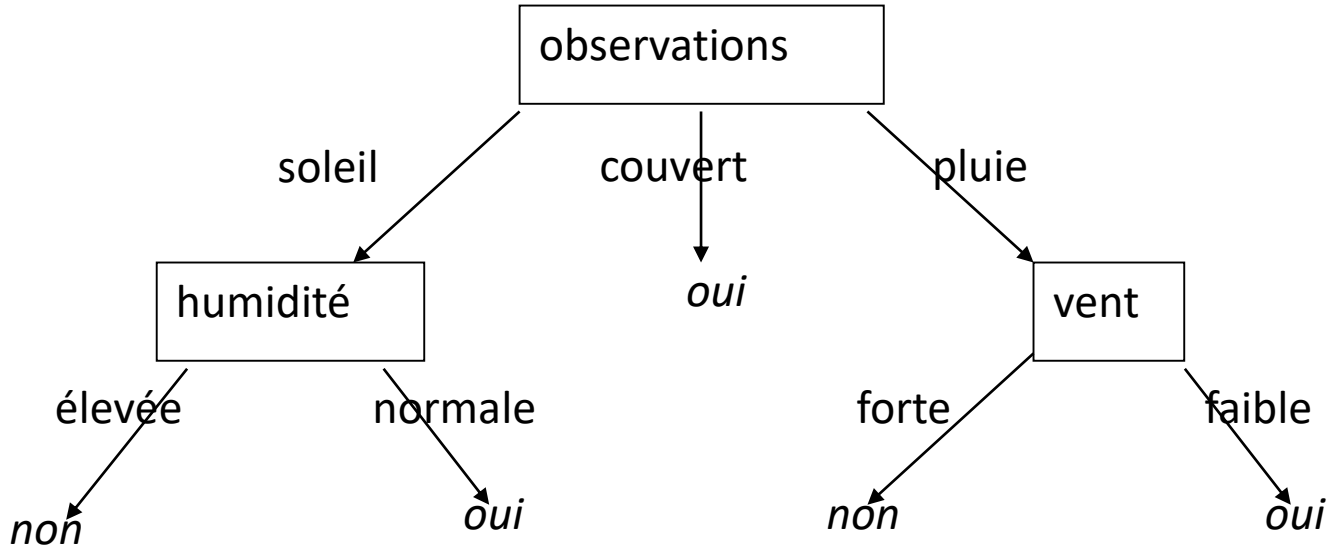
S:[9+,5-]
entropie=0.940

Jours	Observations	Température	Humidité	Vent	Tennis
1	soleil	élevée	élevée	faible	non
2	soleil	élevée	élevée	fort	non
3	couvert	élevée	élevée	faible	oui
4	pluie	moyenne	élevée	faible	oui
5	pluie	tiède	normale	faible	oui
6	pluie	tiède	normale	forte	non
7	couvert	tiède	normale	forte	oui
8	soleil	moyenne	élevée	faible	non
9	soleil	tiède	normale	faible	oui
10	pluie	moyenne	normale	faible	oui
11	soleil	moyenne	normale	fort	oui
12	couvert	moyenne	élevée	fort	oui
13	couvert	élevée	normale	faible	oui
14	pluie	moyenne	élevée	fort	non



Exemple de bruit

- Tennis
- Avec les 14 premiers exemples (ID3).



- Exemple **erroné** supplémentaire : que ce passe-t-il ?
  <obs:soleil,temp:élevée,hum:normale,vent:fort,tennis:non>

Sur-apprentissage

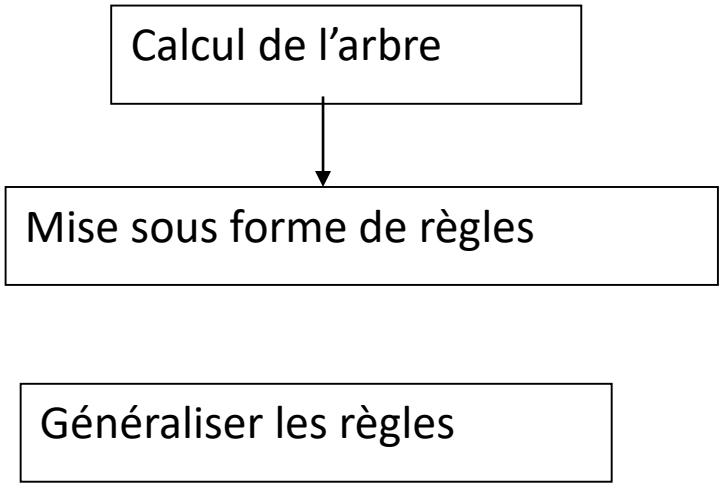
- Comment limiter la taille de l'arbre ?
  - Arrêter au bout d'un certain temps
    - Combien ?
    - Toutes les branches de même longueur ?
  - Elager les branches si elles semblent biaisées

Sur-Apprentissage

- Prendre un ensemble d'exemples pour tester l'apprentissage.
- L'impact des nœuds de l'arbre est mesuré sur cet ensemble (qui n'est pas appris)
- Si l'impact est négatif, le nœud est élagué.

C4.5

- Heuristique
- Protège ID3 contre le biais



Apprentissage par renforcement

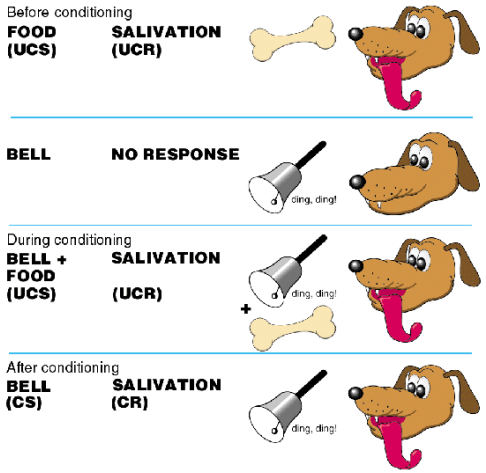
Apprendre en simulant :

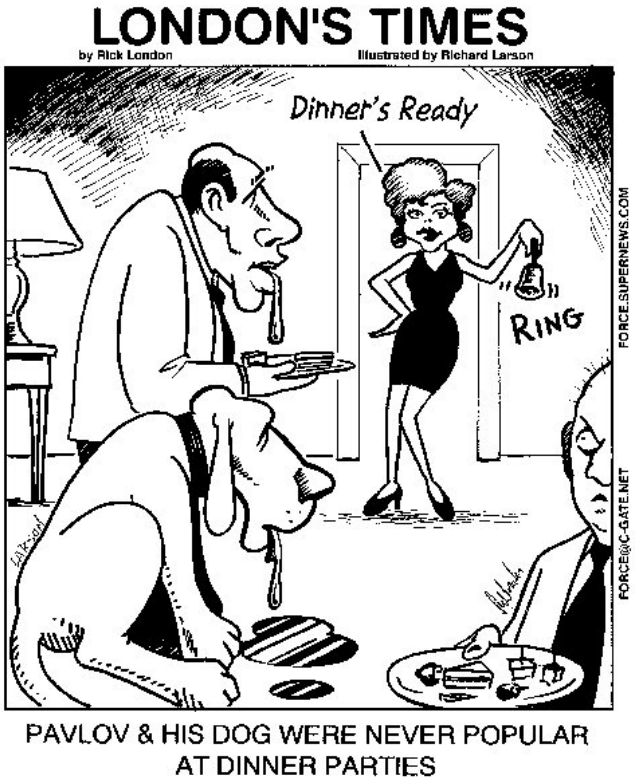
Apprentissage par renforcement

- Apprentissage d'interactions avec un environnement
- Environnement pouvant être dynamique
- Pas d'exemples "tout fait"
- Exploration / Exploitation

Les origines

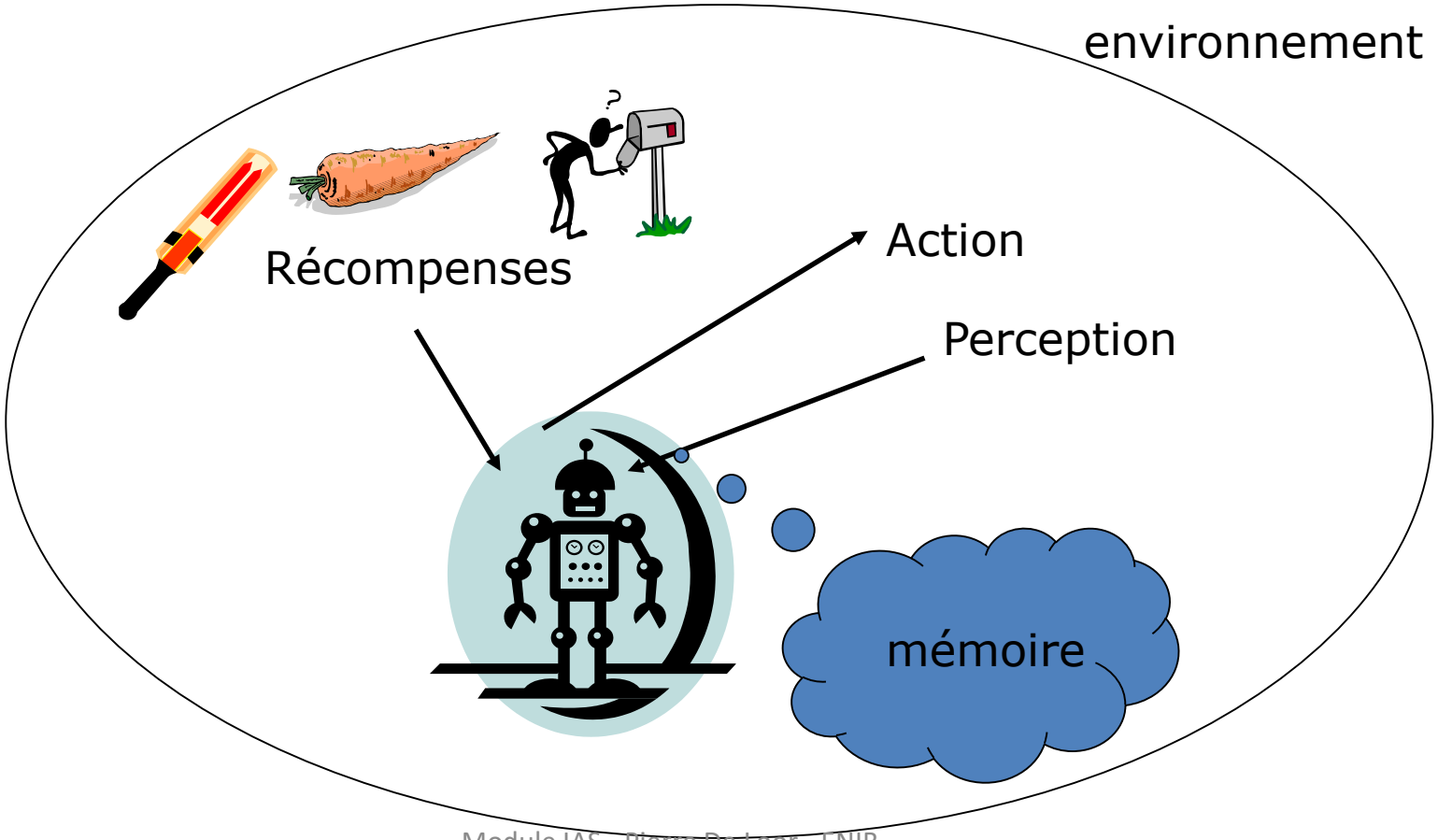
- Le conditionnement …
  - [Thorndike 1911],
  - [Pavlov 1927],
  - [Skinner 1938],
  - [Tolman 1930]



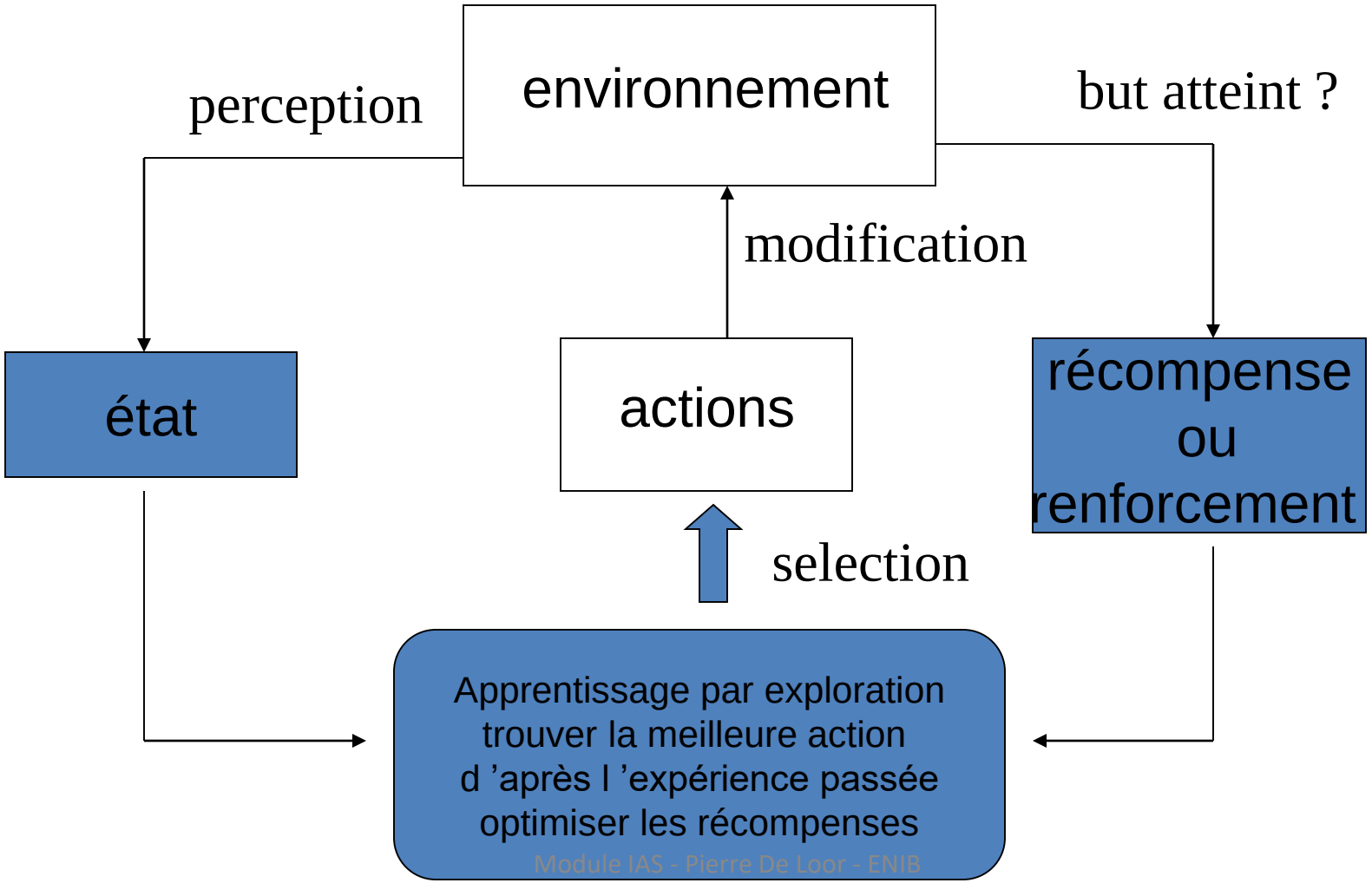


Principe

Apprentissage artificiel par renforcement:
La carotte, le bâton et le silence



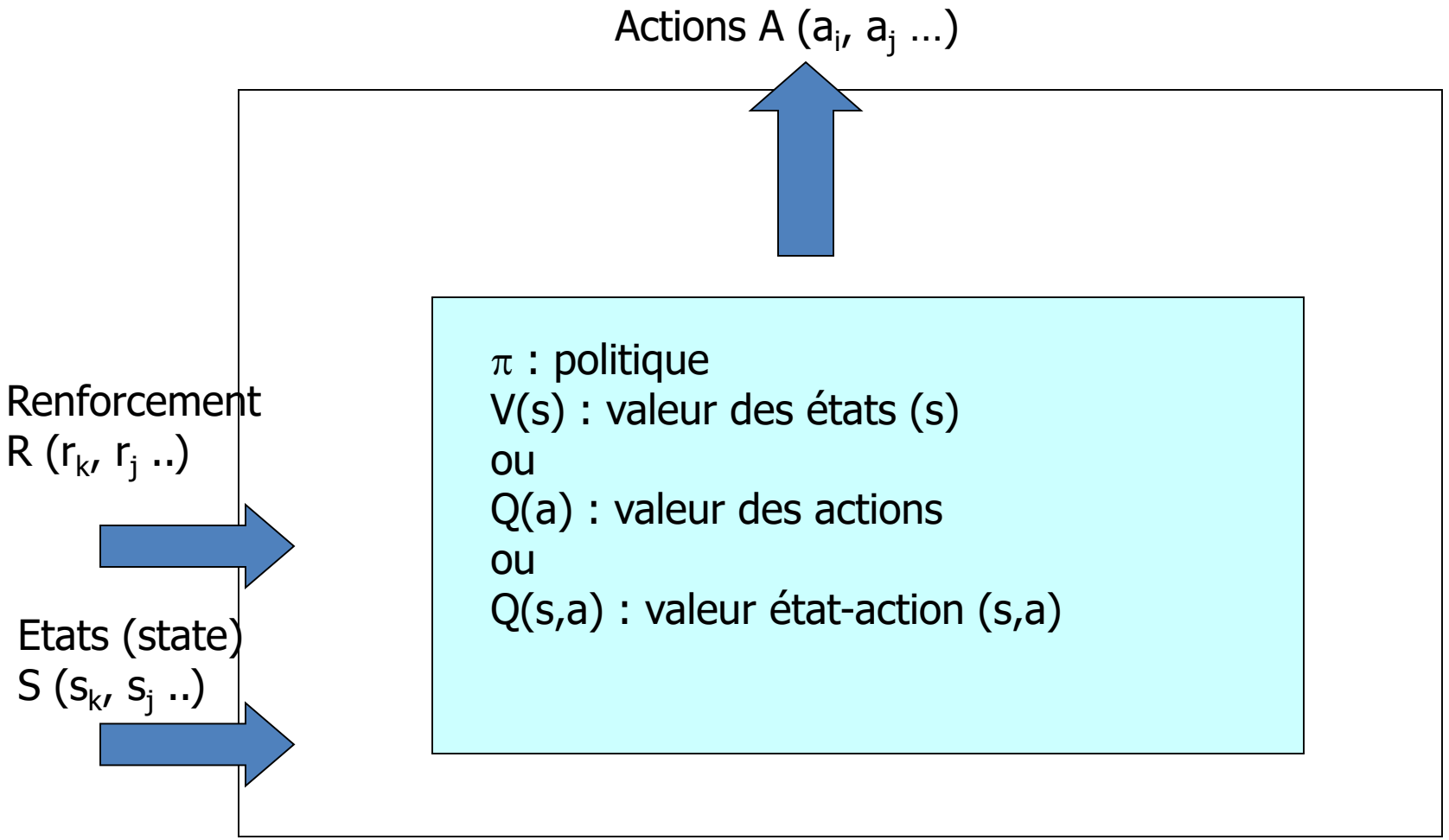
Principe


environnement
perception
but atteint ?
modification
état
actions
récompense
ou
renforcement
selection
Apprentissage par exploration
trouver la meilleure action
d 'après l 'expérience passée
optimiser les récompenses

Bien comprendre le renforcement

- But atteint = récompense
- mauvaise solution = punition
- Entre les deux = rien
    - c'est à l'algorithme de trouver "comment" atteindre le but, pas à nous de lui dire "chaud" ou "froid" car il va apprendre à atteindre un sous objectif.
    - Il n'est pas nécessaire (ni souhaitable) de mesurer le degrès de satisfaction du but, ou d'imaginer les étapes intermédiaires

Variables importantes

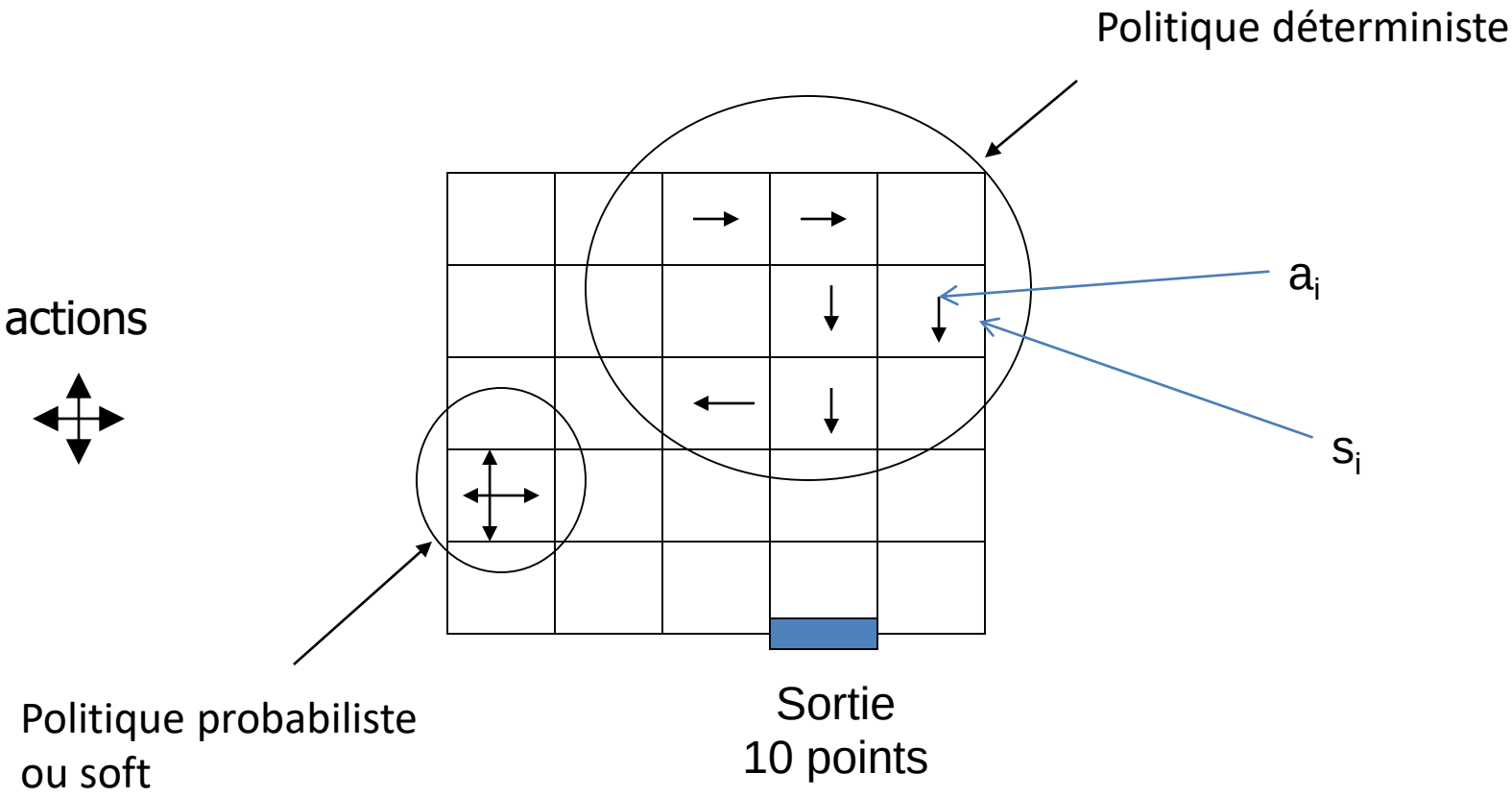


π : Politique(s)

- Caractérise ce qu'il faut faire ? s->a
- probabiliste :
  - π(s,a) : probabilité de choix (dure/permissive)
- déterministe :
  - π(s) : action, pas une probabilité
- π^* optimale : permet d 'obtenir le plus de gains
  - compromis court/long terme , infini ? épisode ?
  - C 'est ce que l 'on cherche à apprendre

Politique π

- exemple



V : Valeur des états (s->$\Re$)

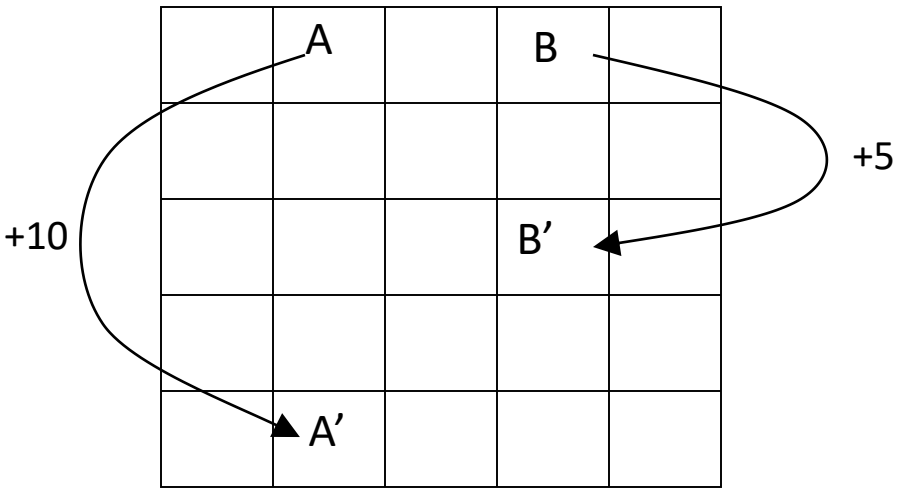
États :	s_1	s_2	s_3	…
Valeur	V(S1)	V(S2)	V(S3)	

- V(s) Gains que je peux espérer en s
  - sur n coups (épisodes) : V(s) est borné
    - Reward =r_{t+1}+ r_{t+2} + … + r_{t+n}
  - à l ’infini : pondération -> remise γ
    - Reward = r_{t+1} + γr_{t+2} + $\gamma^2 r_{t+3}$ +…
- V* : valeur optimale (inconnue)
- V^{π} : valeur pour la politique π

Autre Exemple

- L ’environnement

actions



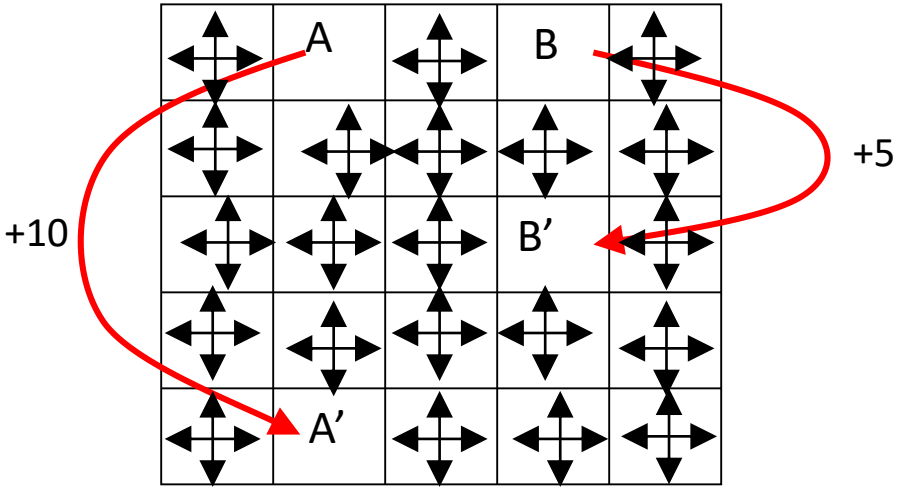
Renforcements
- bords : -1

- A->A ’ : 10
- B->B ’ : 5
(pour toute action)

- ailleurs : 0

Gains des états V^{π} pour une politique π

- politique π : probabilité de choix des 4 directions identique partout $\pi(s_i,\text{droite}) = \pi(s_i,\text{haut}) = \ldots = 0.25$
- Horizon infini, remise : γ=0.9

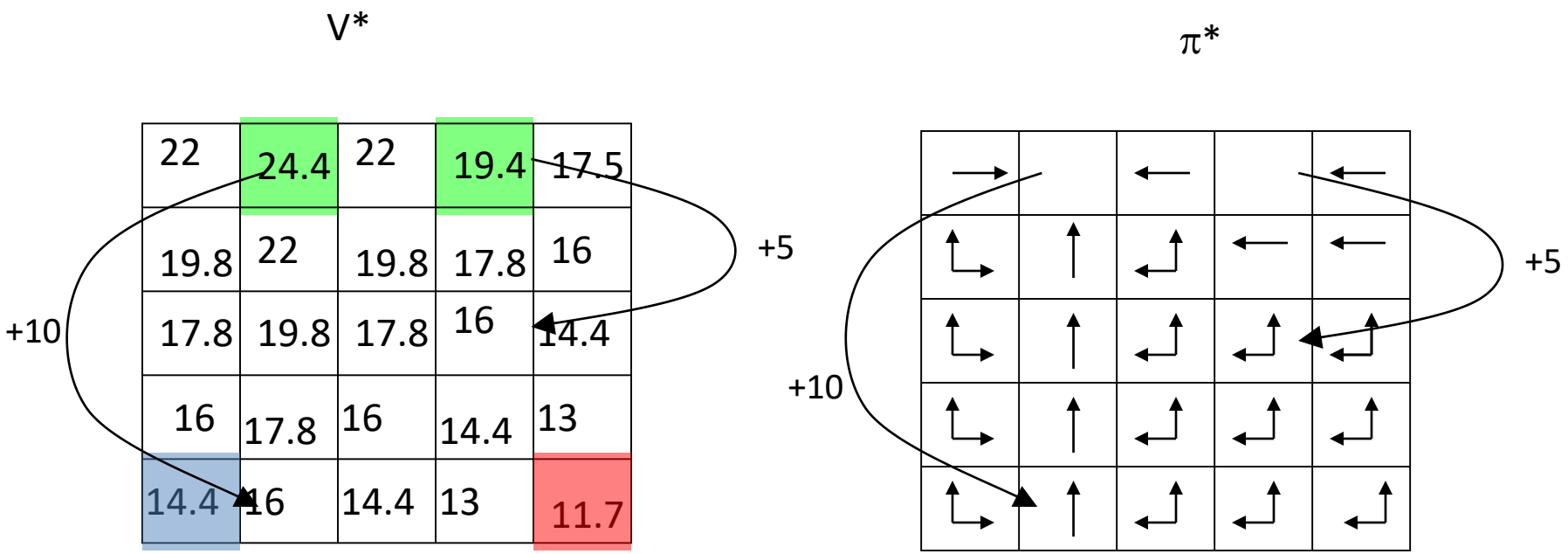


V^{π}

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2

Gain et politique optimale

- V* : gains optimums pour la politique optimale π^*



Plusieurs actions (Q(s,a))

- Association état/action
- Politique π

États :	s_1	s_2	s_3	…
Actions : a1 a2 a3 …	$Q_{(s1,a1)}$ … …	$Q_{(a1,s2)}$	$Q_{(a3,s3)}$	…

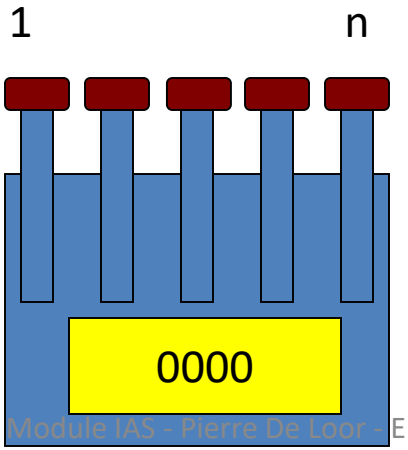
Q (s,a)

- Comme V mais …
  - sur une action : Q(a) : a -> $\Re$
  - sur un couple état/action : Q(s,a) : (s,a) -> $\Re$
  - Q(s,a) : gains espérés en exécutant a à partir de s.
- Q* valeur optimale (inconnue)
- $Q^{\pi}(s,a)$ valeur pour une politique π

Le problème de l'exploration

Bandit Manchot

- pas d'états (ou plutôt un seul)
  - pas de V
- n bras : n actions $a_1 \ldots a_n$ -- > $Q(a_i)$
- chaque bras permet de gagner une somme r distribuée aléatoirement (proba inconnue)



Bandit Manchot

- Explorer : jouer au pif
- Exploiter : jouer le meilleur bras estimé
  - Valeur d'une action a (bras) : Q(a) (= Q*(a))
    - Reflète la moyenne des gains obtenue par a
  - Valeur estimée d'une action après qu'elle soit choisie k_a fois :

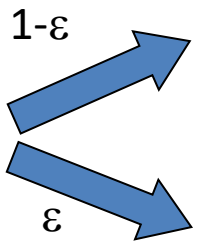
$$Q_t(a) = \frac{r1+r2+\ldots\ldots+rk_a}{k_a}$$

  - si $k_a \rightarrow \infty$ alors $Q_t(a) \rightarrow Q(a)$

Bien apprendre : explorer/exploiter

- Taux d'exploration : ε (ε-Greedy methods)
- ε variable aléatoire ($0 \leq \varepsilon \leq 1$)

probabilité



Jouer le meilleur « greedy action »
$\max_a(Q(a))$

Explorer

- valeur de ε ?
  - ↗ apprentissage lent, prudent, performant à long terme
    versus
  - ↘ apprentissage rapide mais moins performant en moyenne

  – si ε est toujours > 0 on parle de politique ε-soft

Explorations plus fines

- Softmax
  - Dépend de la valeur des actions $Q_t(a_i)$
  - plus une action est “valable” plus elle a de chance d’être choisie (distribution de Boltzman, fixer la température ?)

probabilité
de choisir « a »
parmi n actions

$$\frac{e^{Q_t(a)/\tau}}{\sum_{b=1}^{n} e^{Q_t(b)/\tau}}$$

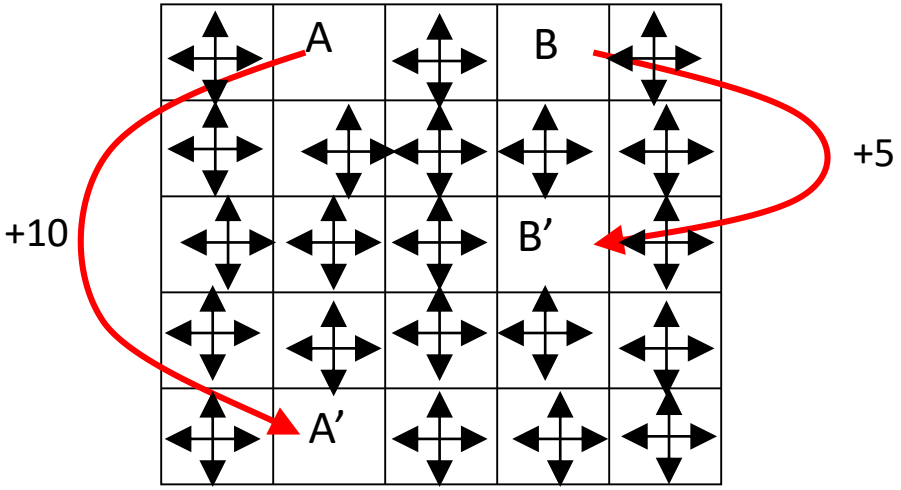
$\tau \to 0$: greedy

$\tau \to \infty$: equiprobable

Le problème de l'apprentissage

Gains des états V^{π} pour une politique π

- $Q_t(a) = \frac{r1+r2+\ldots\ldots+rk_a}{k}$
- Mais quand k-> ∞ (Horizon infini) , il faut une version récursive/incrémentale



3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2

Il est possible d'obtenir une définition incrémentale Q(a)

$$Q_t(a) = \frac{r_1 + r_2 + \ldots + r_{ka}}{ka}$$

$$Q_{k+1} = \frac{\sum_{i=1}^{k+1} r_i}{k+1}$$

$$Q_{k+1} = \frac{\sum_{i=1}^{k} r_i + r_{t+1}}{k+1}$$

Q_k

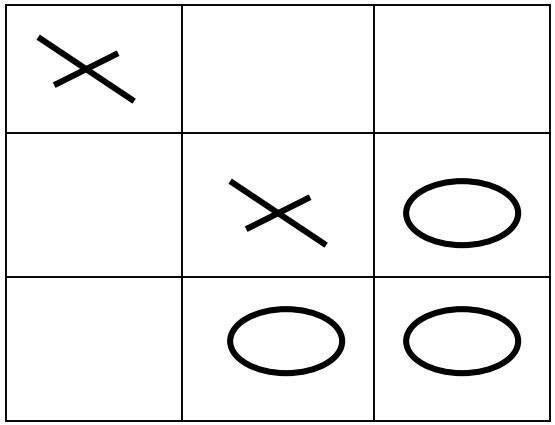
$$Q_{k+1} = Q_k + \frac{1}{k+1} * [r_{k+1} - Q_k]$$

Voir annexes

$$Q_{k+1} = Q_k + \alpha.[erreur]$$

Apprentissage par différence temporelle : Exemple : Tic Tac Toe

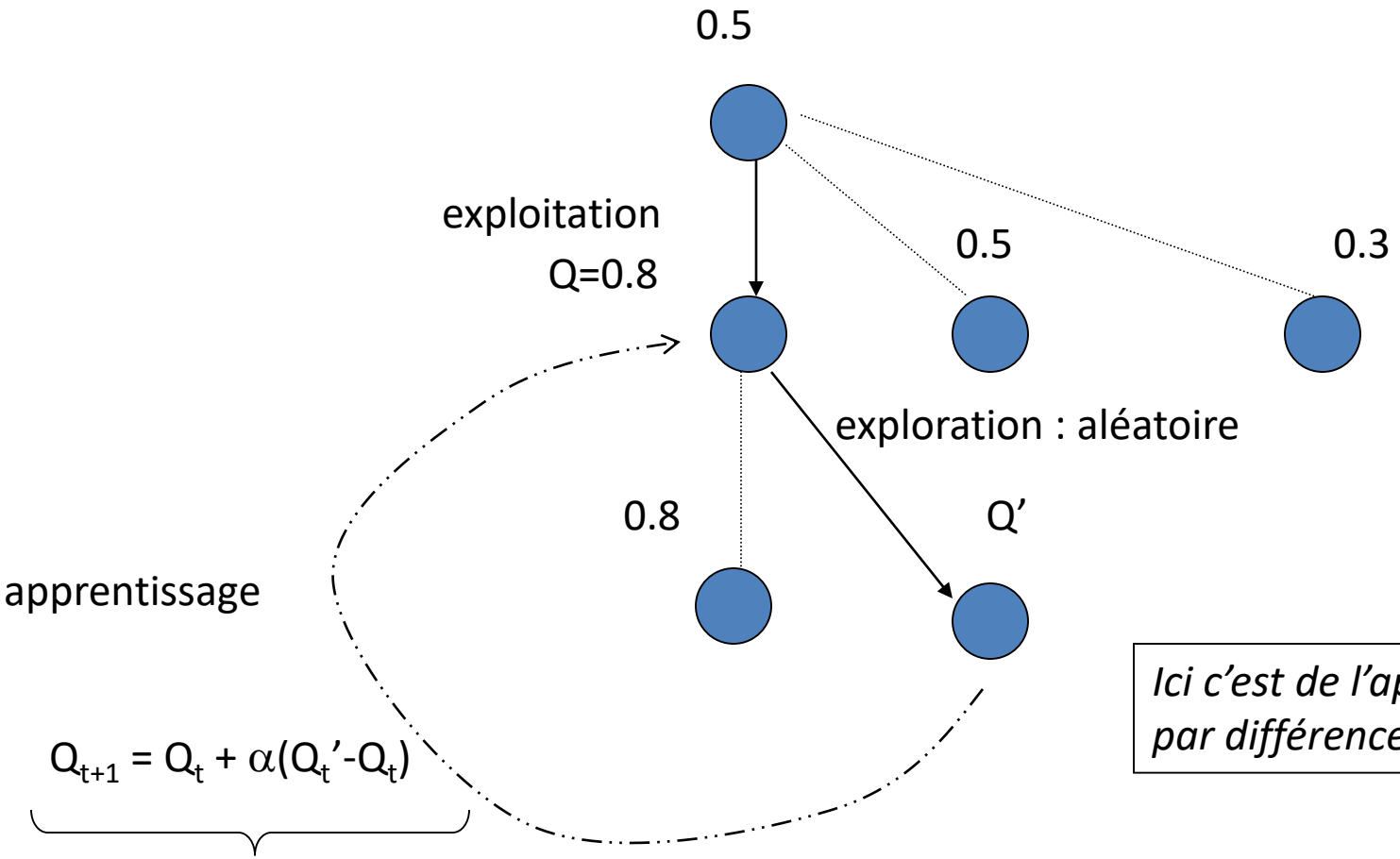
- Joueur parfait ne perd jamais
- Joueur imparfait “pas logique”
  - modèle parfais du joueur (minmax) exclu



Apprentissage par différence temporelle : Exemple : Tic Tac Toe

- Pondérer les états du jeu
  - valeur de l'état = récompense
  - pondération initiale :
    - 1 pour trois croix alignées
    - 0 pour trois rond alignés
    - 0.5 ou aléatoire pour les autres
- exploiter : *aller vers les états de grande valeur*
- explorer : *partir au hasard*
- apprendre : *faire remonter un partie des valeurs de l 'état courant vers les états précédents*

Apprentissage par différence temporelle : Exemple : Tic Tac Toe



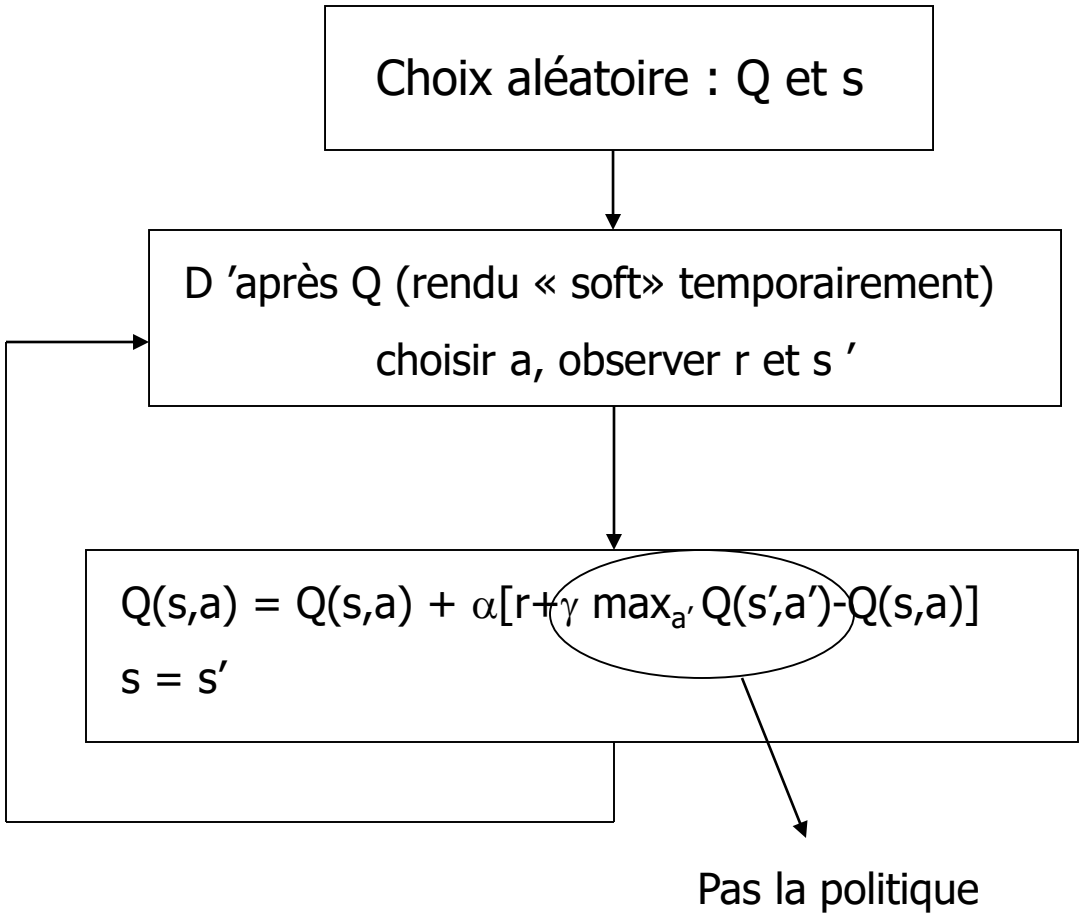
Ici c'est de l'apprentissage par différence temporelle

Bootstrap : l 'estimation de Q dépend de l 'estimation de Q'

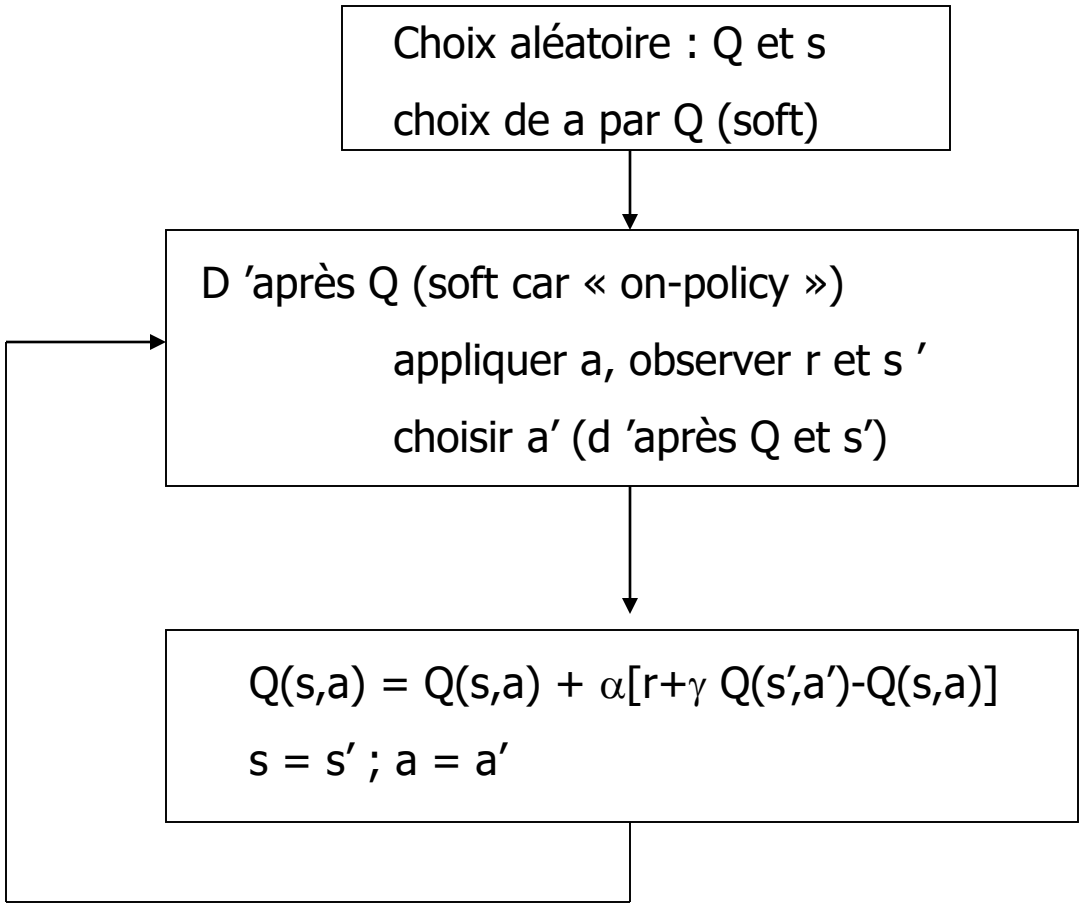
Q-learning : off-policy

- Pendant une simulation à chaque pas :
- $Q(s_t,a_t) = Q(s_t,a_t)+\alpha*[r_{t+1}+\gamma*max_a Q(s_{t+1},a)-Q(s_t,a_t)]$
- max_a recherche l ’action a qui permet d ’obtenir le meilleur Q
- indépendant de la politique que l ’on « joue »

Q-learning : off-Policy



Sarsa : On-Policy (s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})
Apprentisage de Q(s,a)

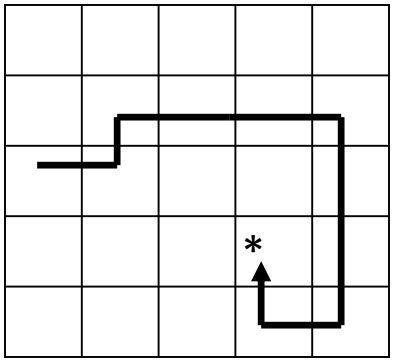


Contrôle du système

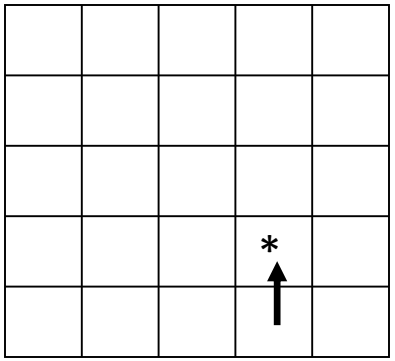
Amélioration de la politique

Sarsa(λ) sur un labyrinthe

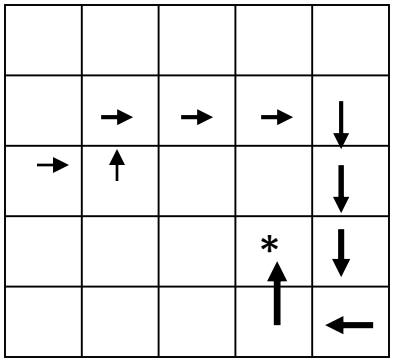
- Les premiers pas sont utilisés pour évaluer Q



chemin



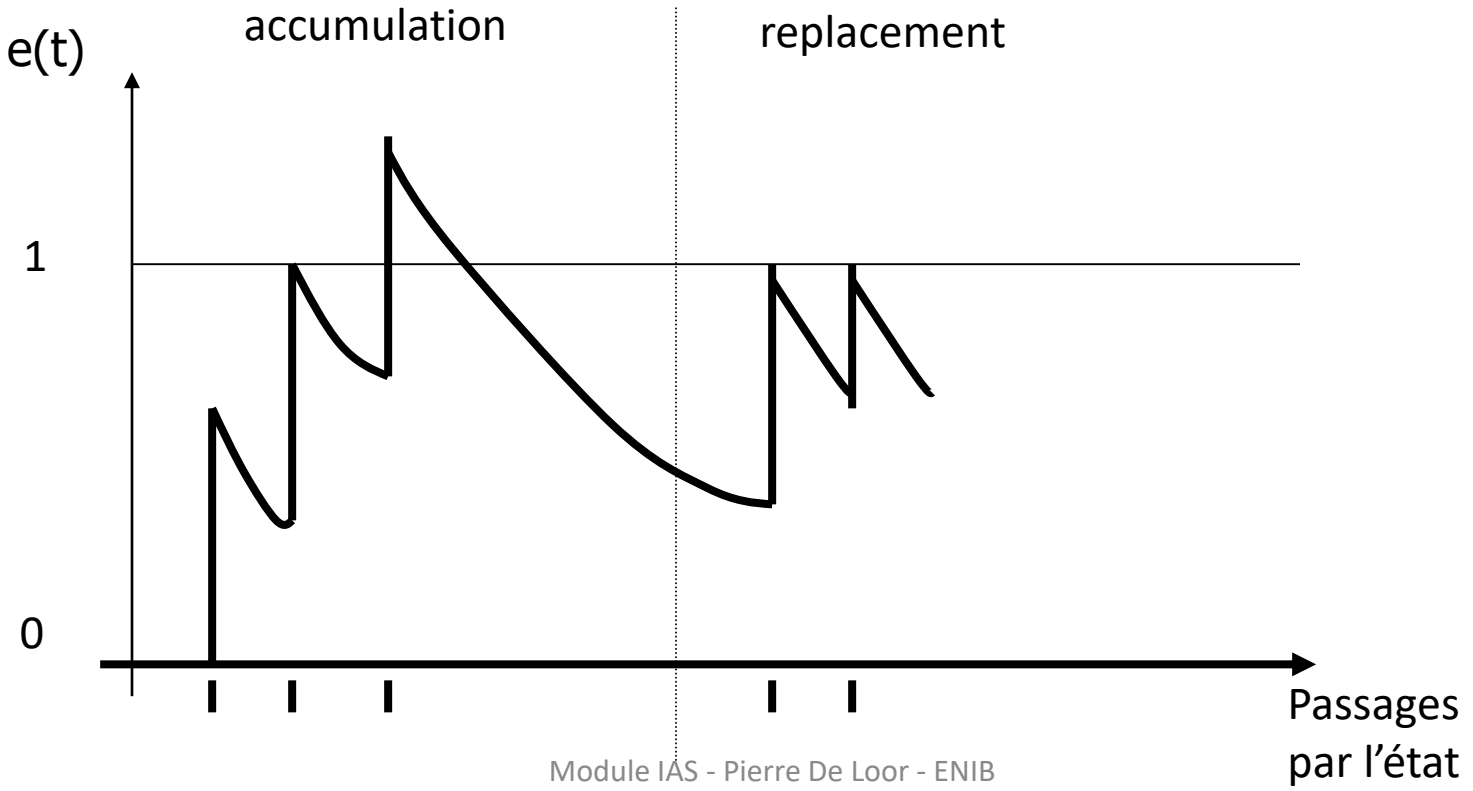
Sarsa



Sarsa(λ)

e(t) trace d ’elligibilité d ’un état (ou état-action)

- Sorte de facteur d’oubli
- Deux possibilités : accumulation ou replacement



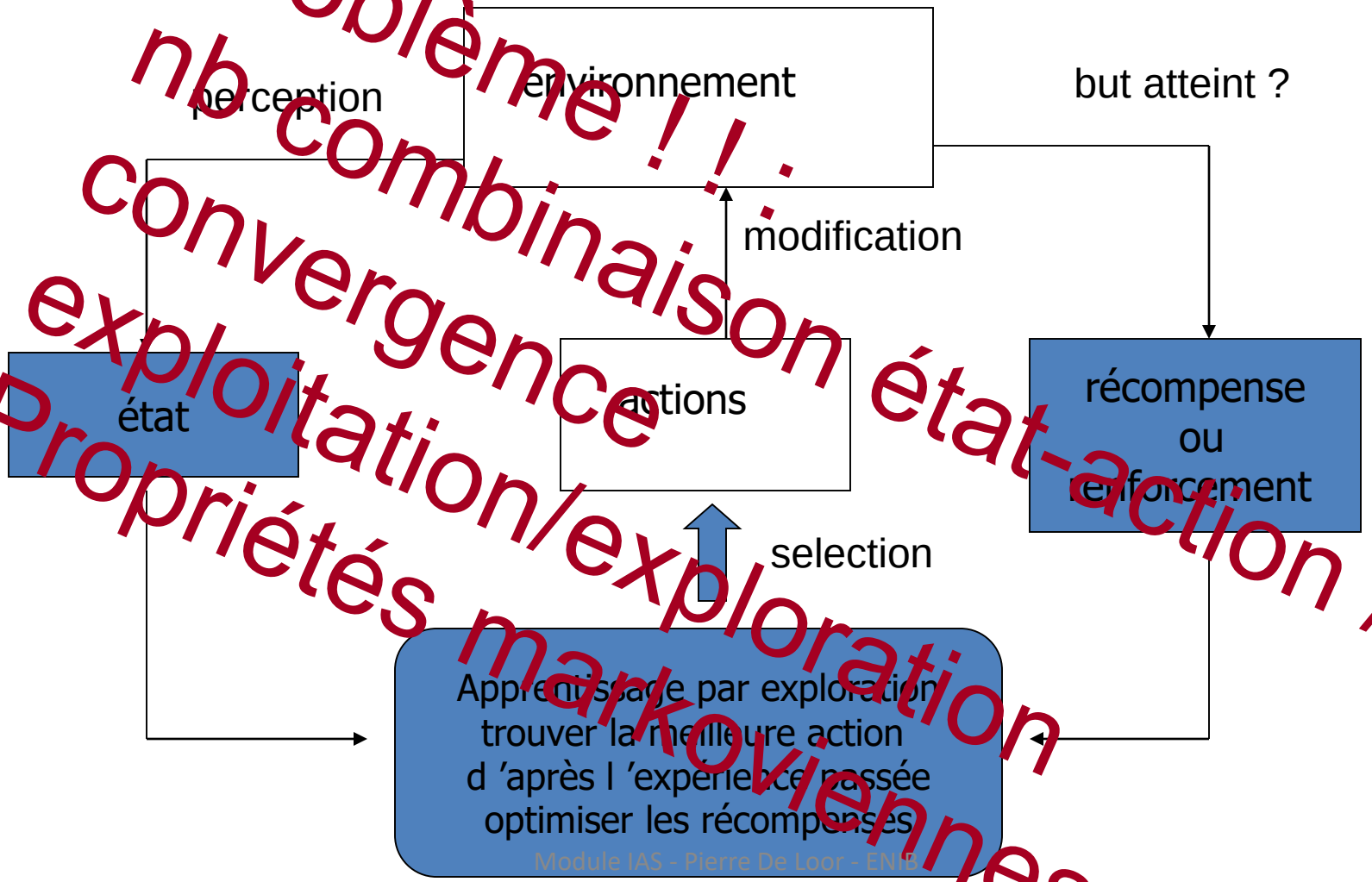
SARSA(λ) : l ’algo complet (en cadeau gratuit)

Initialize Q(s,a) arbitrarily and e(s,a)=0, for all s,a
Repeat (for each episode)
    Initialize s,a
    Repeat (for each step of episode)
        Take action a, observe r, s’
        Choose a’ from s’ using policy derived from Q (e.g., ε-greedy)
        $\delta \leftarrow r + \gamma Q(s,a') - Q(s,a)$
        $e(s,a) \leftarrow e(s,a)+1$ ← accumulation
        For all s, a:
            $Q(s,a) \leftarrow Q(s,a) + \alpha\ \delta\ e(s,a)$ ← Ça c’est intéressant
            $e(s,a) \leftarrow \gamma\lambda e(s,a)$
        $s \leftarrow s'$; $a \leftarrow a'$
    until s is terminal

Le problème d'une approche markovienne

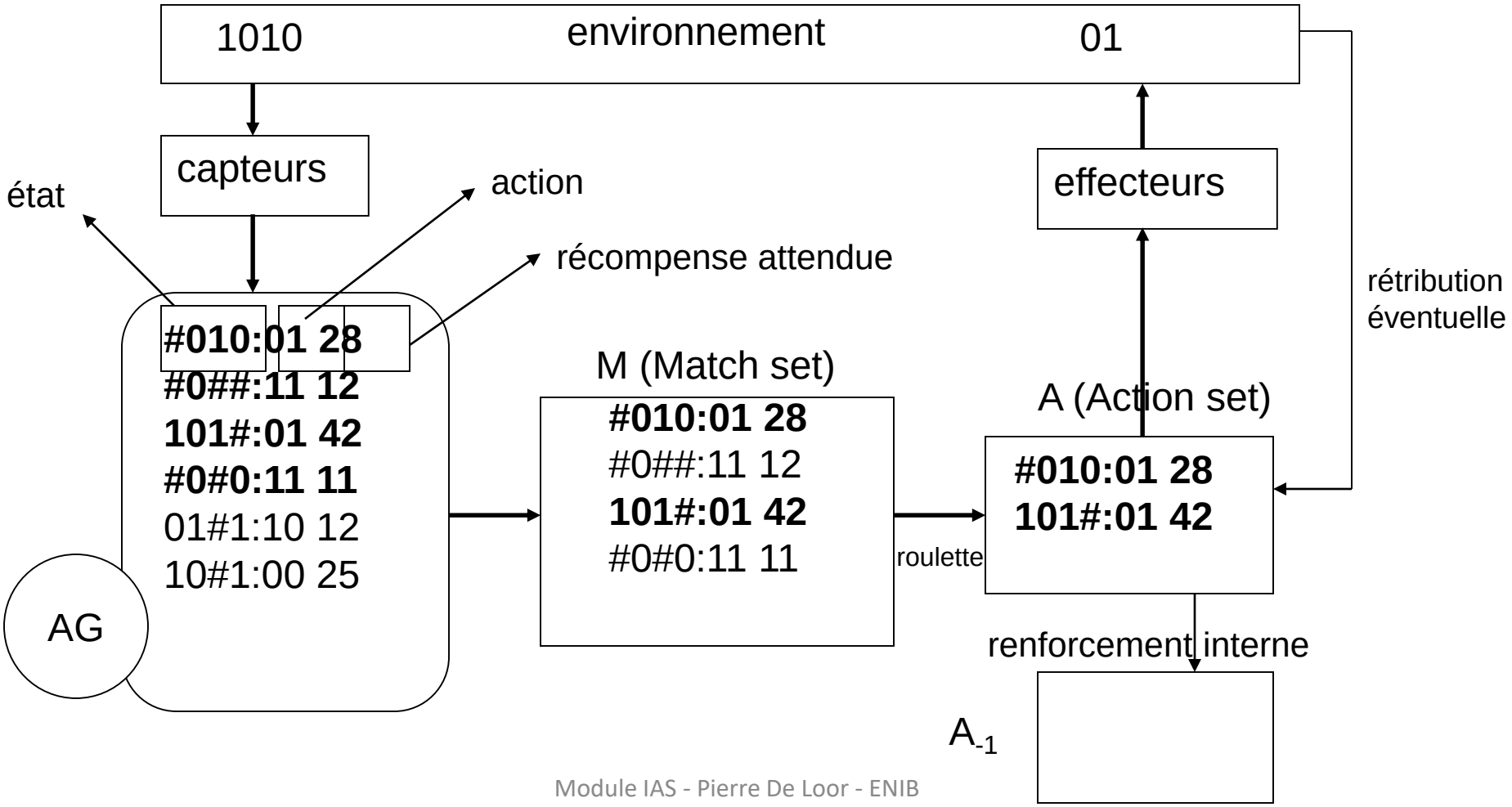
- Chaque état :
  - représentent le passé de façon condensé
  - suffisant pour prendre les décisions
  - propriété Markovienne
    - le choix de la prochaine action à exécuter dépend juste de l'état présent
  - pas d'historique
  - exemple : situation d'un jeu d'echec

Apprentissage par renforcement

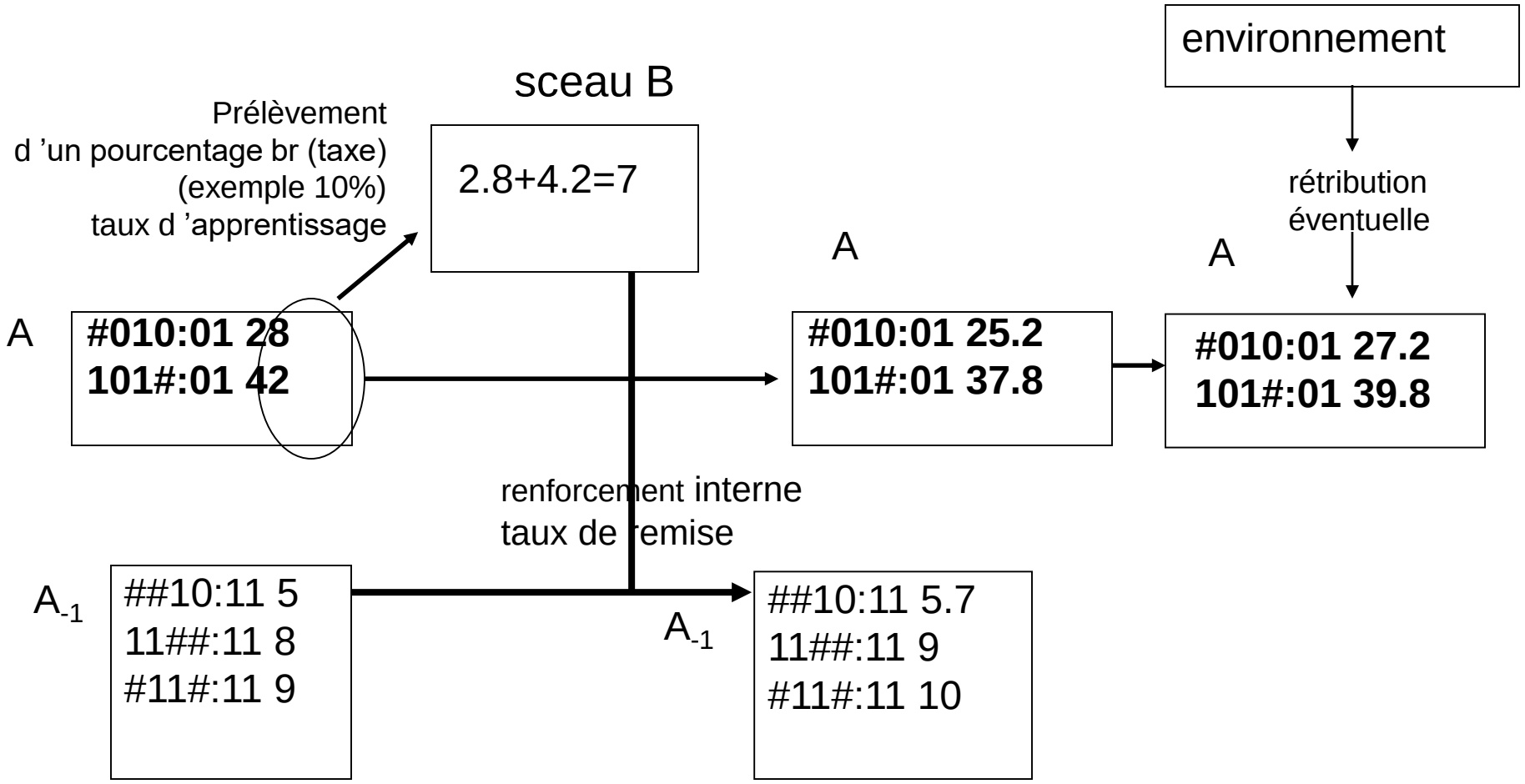


Problème ! ! .
nb combinaison état-action /
convergence
exploitation/exploration
Propriétés markoviennes

Système de classeurs



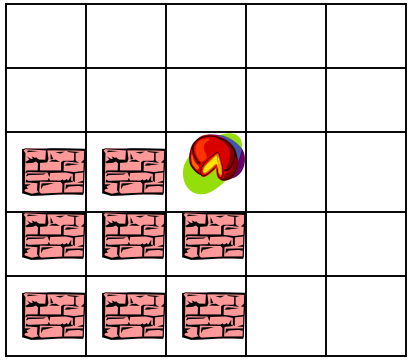
Renforcement des ZCS : Algorithm « Bucket Brigade »

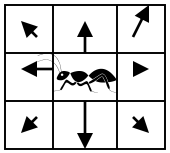


Variante pour convergence : diminution des poids des classifieurs de M-A

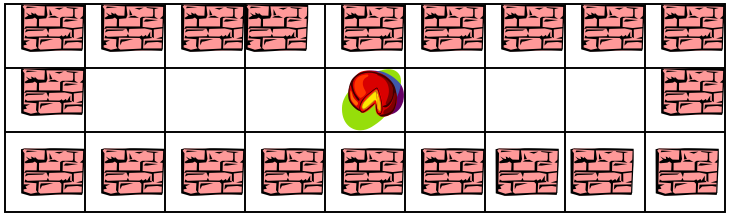
ZCS

- privilégie les règles rapportant plus
- élimine les autres (oubli = surprise = « covering » = pif)
- quelle que soit leur « précision »
- n ’apprend que des environnements markovien





Perception états

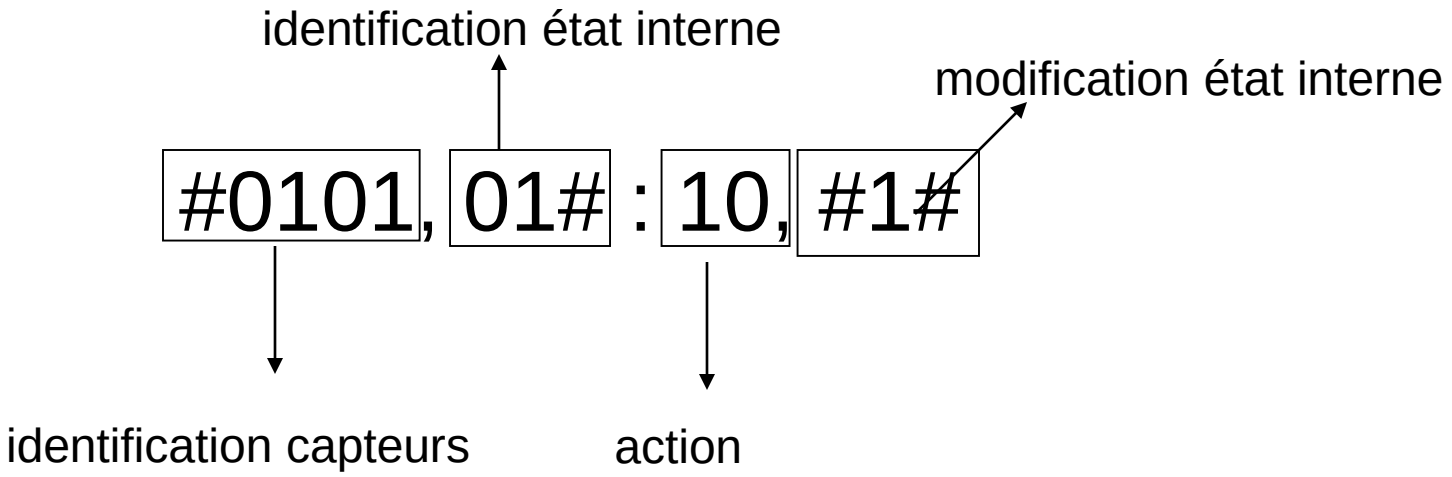


Woods1 : markovien

Woods100 : non markovien

Environnement non markovien : renforcer l ’historique : ZCSM

- Codage d ’une règle



Préserver les classifieurs précis : XCS

- Privilégier les règles les plus « préçises »
- Une règle #010:01
  - + 3 attributs
    - prédiction (de rétribution) : utilisé pour la sélection (M->A)
    - erreur de prédiction
    - fitness (qualité/précision) : utilisé par l ’AG pour la reproduction (faite sur A (niche))
  - mise à jour : Q-learning

Systèmes de classifieurs XCS

- Apprentissage
  - Prédiction : pour toutes les régles de l ’ensemble M (Match set)

$$P(a_i) = \frac{\sum P_i . F_i}{\sum F_i}$$

  - Définition de l ’ensemble A (roulette ou autre d ’après $P(a_i)$)
  - Application de l ’action et réception de la rétribution P
  - Pour tous les éléments de A, mise à jour de la prédiction :

    $p = p + \beta.(P-p)$ [$0<\beta<1$: taux d ’apprentissage]
  - Mise à jour de l ’erreur de prédiction :
    - $e = e + \beta.(|P-p|-e)$
  - Mise à jour de la fitness :
    - $F = F + \beta.(k-F)$ [k fonction décroissante de e (e↘ : F↗)]

Applications

- Jeux vidéos
- Robotique
- Finances / Systèmes experts

Références

- [Cor2002] Antoine Cornuéjols et Laurent Miclet, **Apprentissage artificiel, concepts et algorithmes**, éditions Eyrolles, 2002.
- [Ger2002] P. Gérard, **Systèmes de classeurs : étude de l'apprentissage latent**, thèse de doctorat de l'université de Paris 6.
- « Machine Learning » *Tom Mitchell, McGraw-Hill,* international edition, 1997.
- [pes99] Peshkin, Leonid and Meuleau, Nicolas and Kaelbling, Leslie P., **Learning Policies with External Memory,ml99**, editor : Bratko, I. and Dzeroski, S, pp 307-314, 1999.
- [Ria2000] **Algorithmes d 'apprentissages et applications**, revue d 'intelligence artificielle, volume 14, n°3, édition hermes, 2000
- Cédric Sanza, **Evolution d 'entités virtuelles cooperatives par système de classifieurs**. These de doctorat de l 'université de Toulouse, 2001.
- [Sut2000]R.S.Sutton and A.G.Barto, **Reinforcement Learning,** MIT Press, 2000.
- [Ski38] B.F. Skinner, **The Behavior of organisms**. Appleon Century Croft, New York, 1938.
- [Thor11] Thorndike, **Animal Intelligence**, 1911.
- [Tol30] E.C. Tolman et C.H. Honzik. **Insight in Rats**. University of California Publications in Psychology, 4: 215-232, 1930.
- [Whi91] Whitehead and Ballard, **Learning to Perceive and Act by Trial and Error**, Machine Learning, 7, 45-83, 1991.
- **Apprentissage automatique et évolution artificielle**, revue extraction des connaissances et apprentissage, Volume1, n°3, éditions hermes, 2001.
- Algorithmes d 'apprentissages et applications, revue d 'intelligence artificielle, volume 14, n°3, édition hermes, 2000.

Aspect théorique

- Apprentissage optimal
  - MDP (Markov Decision Process), equation de Bellman
  - Connaissance de la dynamique de l ’environnement.

$\mathcal{P}_{ss'}^{a}$ Probabilité de passage de s à s’ par l ’action a

$\mathcal{R}_{ss'}^{a}$ Renforcement lors du passage de s à s’ par a

  - V peut être calculé
  - V* existe et peut être calculé
  - π* existent et peuvent être calculées

Maximiser quoi ?

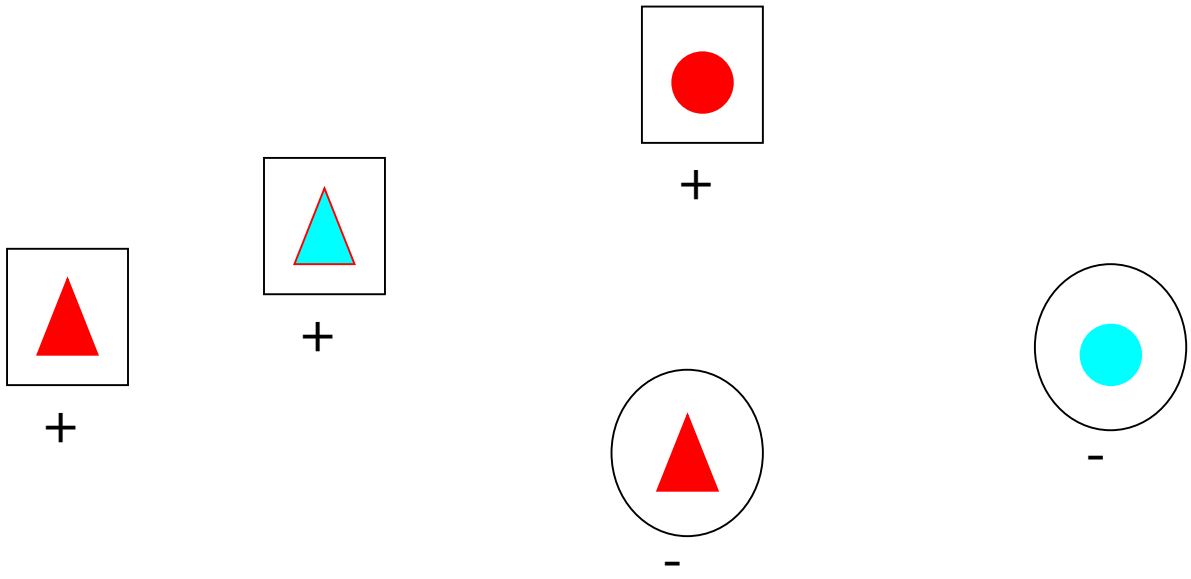
- Maximiser les gains à long terme
  - la meilleure action “maintenant” n’est pas forcément la meilleure à long terme
  - Ne pas maximiser les renforcements “immédiats” !!!
- Apprentissage par épisodes (T)
  - Maximiser $R_t = r_{t+1} + r_{t+2} + \ldots + r_T$
  - exemple : sortie d ’un labyrinthe

Le long terme

- Apprentissage continu : pas de fin
  - la somme des renforcements peut croître indéfiniment quelque soit le choix
  - comment savoir que l'on est sur la bonne voie ?
  - Pondérer le long terme
  - $R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots$
  - $0 \leq \gamma \leq 1$: remise (discount)

Question 5

- Comment fait un programme pour apprendre ?



La représentation des connaissances

- Logique des prédicats
- Réseaux sémantiques (Collins et Quillian 1969) voir annexe A2
- Graphes conceptuels (Sowa 1984)
- Frames (Minsky 1975)
- Logiques non monotones, probabilistes, possibilistes
- Méthodologie KADS
- Ontologies

Notion d'ontologie

- Situation : chef de projet d'une entreprise pharmaceutique : « What is the balance of the project ? «

Thomson, precise to 0.20mg

Balance of 1,5 million €

1 terme pour deux domaines … ambiguité

Notion d'ontologie

- L'ontologie demande l'étude des catégories des choses qui existent ou peuvent exister dans le domaine d'application.

- Étymologie : Ontos(être) Logos(mot)

- Elle sert de « support » à un système logique

Par elle-même une logique ne signifie rien, c'est sa combinaison avec une ontologie qui fournit un langage d'expression à propos d'un domaine

Exemple d'ontologies

- Goi-Taikei's ontology
  - Un lexique de 400 000 mots Japonais pour la traduction
    - Une ontologie
    - Un dictionnaire de la sémantique des mots
    - Un dictionnaire de la structure de la sémantique
  - L'ontologie classe les concepts à utiliser dans l'expression de lien entre les mots. La signification des noms commun est donnée par un arbre 'sémantique hiérarchique' de 2710 nœuds. Chaque nœud est une 'classe sémantique' les arcs représentent une relation 'est-un' ou 'possède-un'. Il y a également 200 classes pour les noms propres et 108 classes de prédicats. Les mots peuvent être affectés à n'importe qu'elle classe sémantique. Cette sémantique permet la classification des noms verbes et adjectifs.

Exemples d'ontologie

- PHYSSYS : modélisation et simulation de systèmes physiques.
- Molecular-Interactions Ontology : ontologie sur les molécules et leurs interactions.
- MeSH : Classification des termes liés à la médecine (anatomie, trouble mentaux, enzymes …)
- EngMath : Ontologie des mathématiciens. Traitent de concepts tels que tenseurs, scalaire, unités, vecteurs, événements discrets …
- WorldNet : Clasification de la langue Anglaise selon une théorie psycholinguistique (antonymes, synonymes, hyperonyme …), 121962 mots, 99642 concepts
- Et beaucoup d'autres …

Re encore des doutes ?

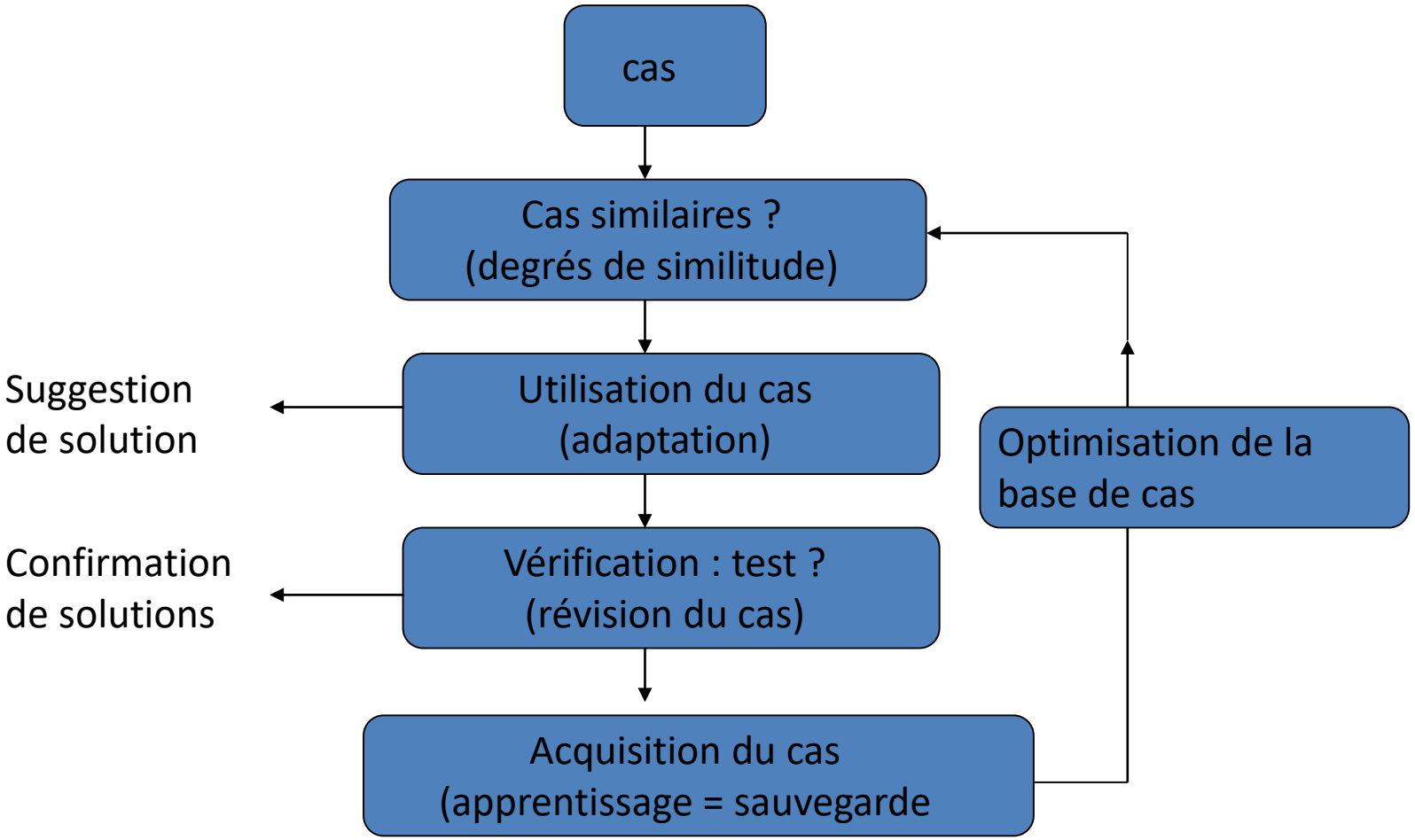
- Généralisation illusoire
- Niveau de détail indéfinissable
- Modélisation du **bon** sens
- Modélisation du savoir faire
- Lien entre les connaissances
- Systèmes ouverts

Techniques Emergentes en IA

Une autre façon de résoudre un problème ?

- En se rappelant la façon dont on a résolu un problème similaire
- C'est le "Case Based Reasoning" (CBR)!
  - Résolution de problème basé mémoire
  - Réutilisation d'expériences passées
  - Raisonnement par analogie
- Les experts sont plus à même de raconter des cas passés que d'expliquer des règles

Raisonnement à base de cas



Machine Learning

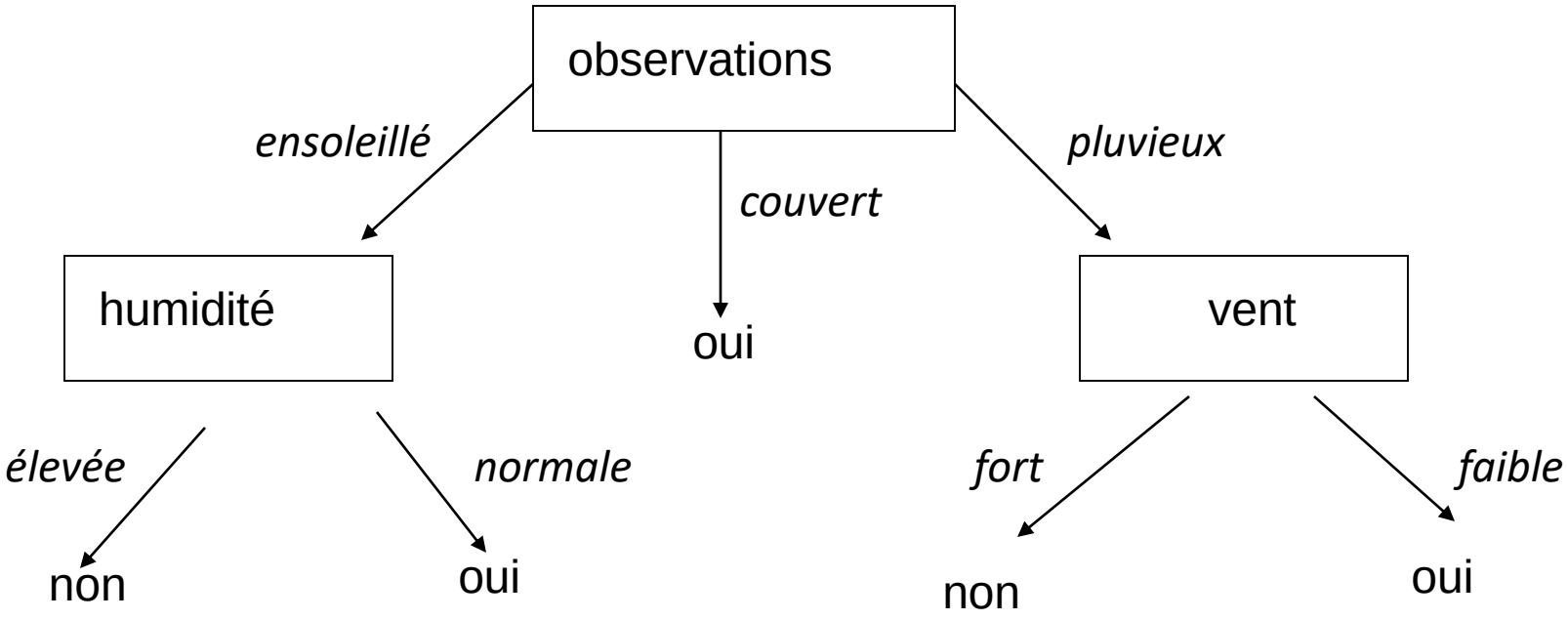
Apprentissage artificiel ou

Jouer au tennis

Jours	Observations	Température	Humidité	Vent	Tennis
1	soleil	élevée	élevée	faible	non
2	soleil	élevée	élevée	fort	non
3	couvert	élevée	élevée	faible	oui
4	pluie	moyenne	élevée	faible	oui
5	pluie	tiède	normale	faible	oui
6	pluie	tiède	normale	forte	non
7	couvert	tiède	normale	forte	oui
8	soleil	moyenne	élevée	faible	non
9	soleil	tiède	normale	faible	oui
10	pluie	moyenne	normale	faible	oui
11	soleil	moyenne	normale	fort	couvert
13	couvert	élevée	normale	faible	ouioui
12					
14	pluie	moyenne	élevée	fort	non

Arbre de décision

- conjonctions et disjonctions
- Exemple : jouer au tennis



Machine Learning

- Apprentissage inductif supervisé
  - C4.5,ID3
  - Réseaux neuronaux feed-forward
  - Réseaux Bayesiens
- Apprentissage par renforcement
  - Q-learning
  - Systèmes de classeurs
- Algorithmes génétiques
  - Microbial
- Un grand défis de l'IA "classique"

Questions philosophiques

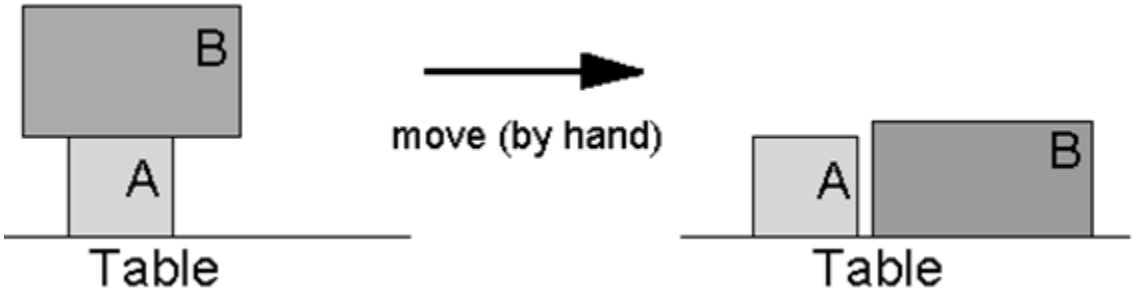
Quand la raison s'emmèle

Les problèmes de l'IA classique (la GOFAI)

- Problème du cadre (Frame problem)
- Problème du sens
- Problème de l'ancrage des symboles
- Problème du savoir-faire

Frame problem (1/2)

[McCarthy and Hayes 1969]



On(B,A)
On(A,Table)

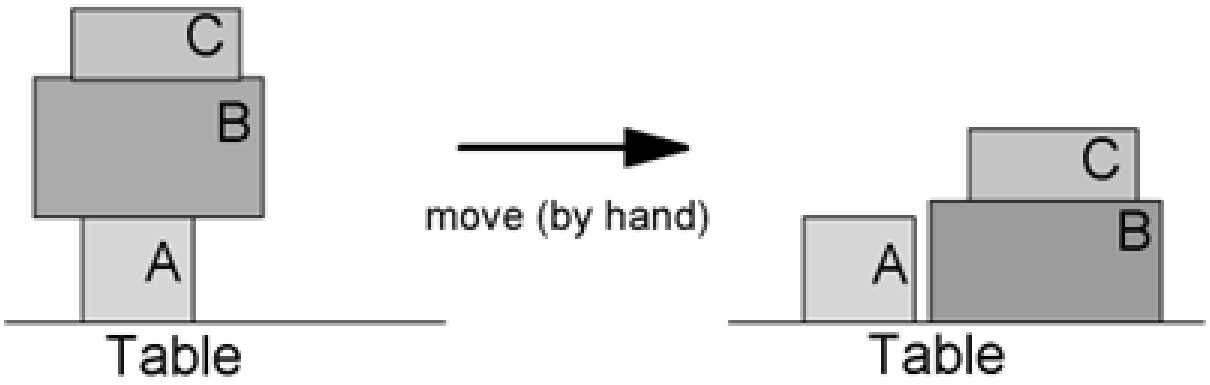
Put(B,Table)

On(B,Table)
On(A,Table)

- En logique, on décrit le passage d'un état à un autre à l'aide d'axiomes :

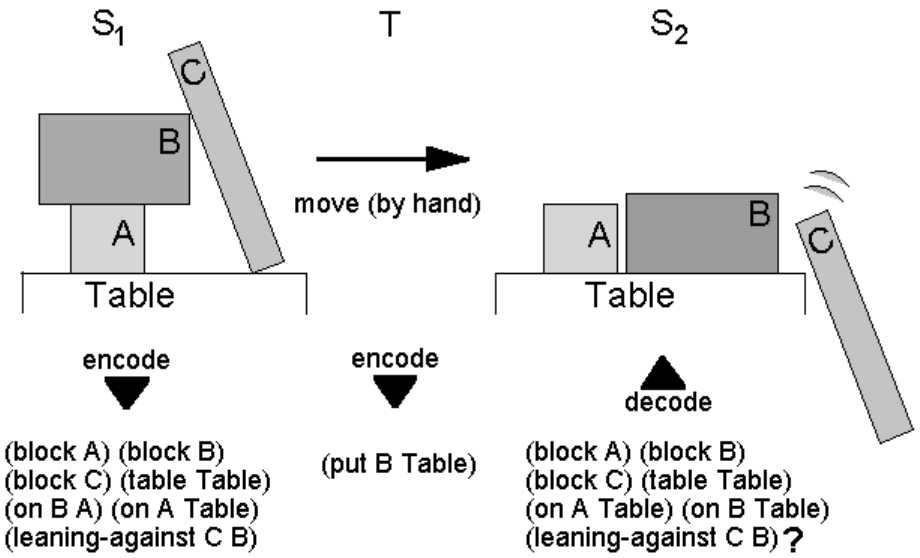
$$on(B,A)(0) \wedge on(A,Table)(0) \wedge put(B,Table)(0) \rightarrow \neg on(B,A)(1) \wedge on(A,Table)(1) \wedge on(B,Table)(1)$$

Cas plus complexe



- C reste sur B même si l'on dépose B sur la table. Or d'un point de vu logique, ce n'est pas implicite, il faut l'écrire.
- Il faut décrire TOUT CE QUI NE CHANGE PAS
- C'est ça le "frame problem". Plusieurs solutions techniques sont proposées.

Cas plus complexe



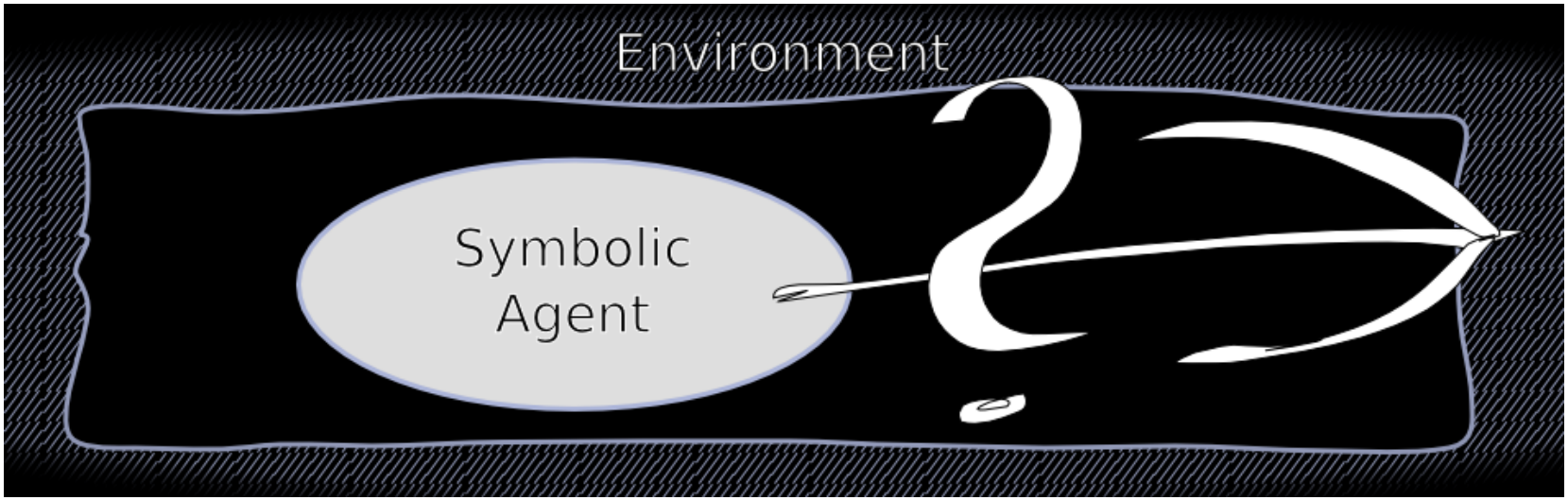
- [Dennet 1987] : Question épistémologique sur la rationnalité : attaque du cognitivisme.

La chambre chinoise : sens versus syntaxe

- [Searles 1980]

Symbol grounding problem [Harnad 1990]

- Comment la signification des symboles est ancrée dans quelque chose qui n'est rien d'autre que des symboles sans signification ?



- Cognition is not the same as computation because computer do not transduce [Harnad 1990]

Simuler la complexité : l'approche bottom-up

La nouvelle IA

Vie Artificielle

« *La vie artificielle est l'étude des systèmes construits de mains d'homme qui exhibent des comportements des systèmes naturels vivants. Elle vient en complément des sciences biologiques traditionnelles, en tentant de synthétiser des comportements semblables au vivant au sein d'ordinateurs et d'autres substrats artificiels. En étendant les fondements empiriques sur lesquels la biologie est basée au-delà de la vie a base de carbone qui a évolué sur Terre, la vie artificielle peut contribuer à la biologie théorique en positionnant la vie telle que nous la connaissons au sein d'un espace plus large: la vie telle qu'elle pourrait être* »

C. Langton

Qu’est que la vie ?

- Ensemble de propriétés [Monod]
  - Téléonomie, morphogenèse, reproduction invariante
- Extensions des propriétés [Mayr, Belin]
  - Structure, macromolécules, ADN, stabilité, évolution
- Approche Thermodynamique [Schrödinger, Prigogine]
  - Structure dissipative
- Autopoïèse [Maturana, Varela]
  - Réseau fermé d’éléments « auto-régénérant »

Jeu de la vie : une simulation minimale

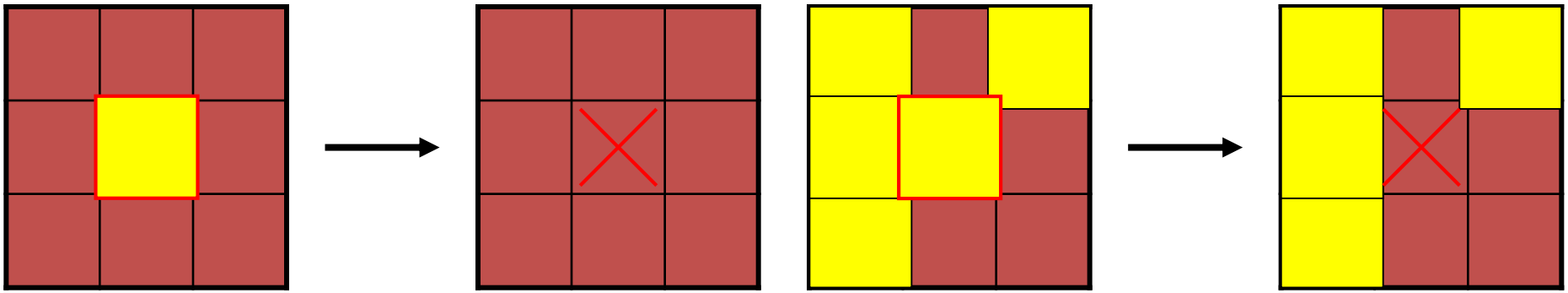
- Une case ‘morte’ avec exactement trois voisins ‘vivants’ devient vivante
- Elle reste vivante tant qu’elle possède 2 ou 3 voisins
- Sinon elle meurt

Jeu de la vie

- John Conway 1970

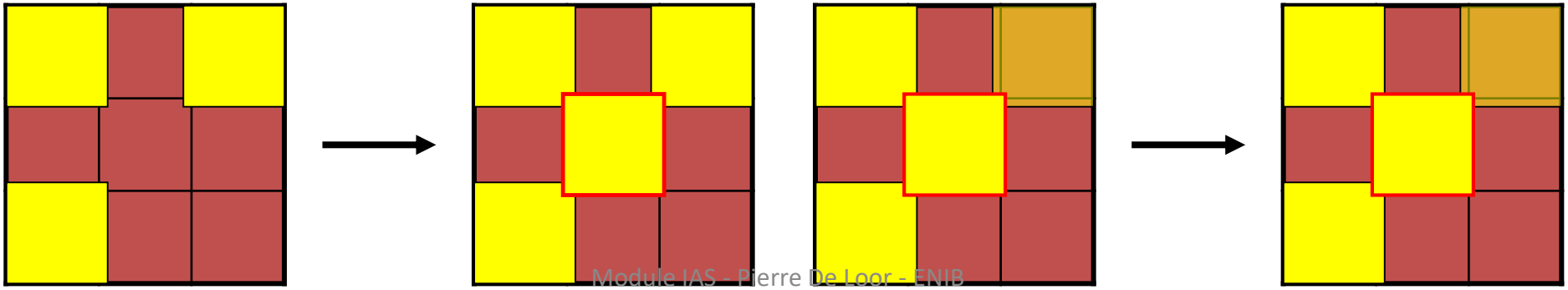
<u>Principe</u>:

1 ou + de 3 voisins → MORT



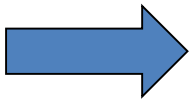
3 voisins → NAISSANCE

2-3 voisins → survie



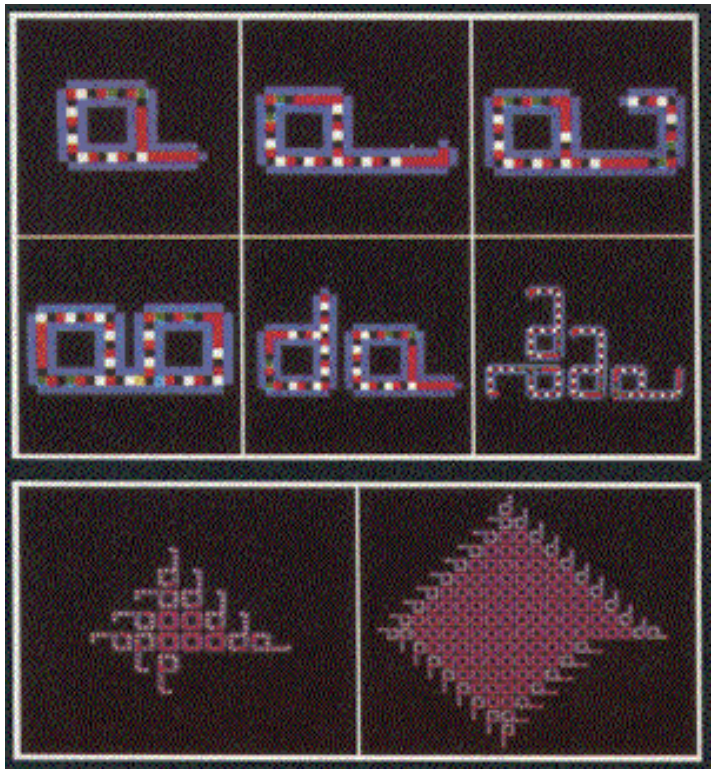
Jeu de la vie

0	0	0	0	0	0	0	0	0
0	1	2	1	0	0	0	0	0
0	**1**	**1**	**2**	1	0	0	0	0
1	**3**	**5**	**3**	2	0	0	0	0
1	**1**	**3**	**2**	2	0	0	0	0
1	2	3	2	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0



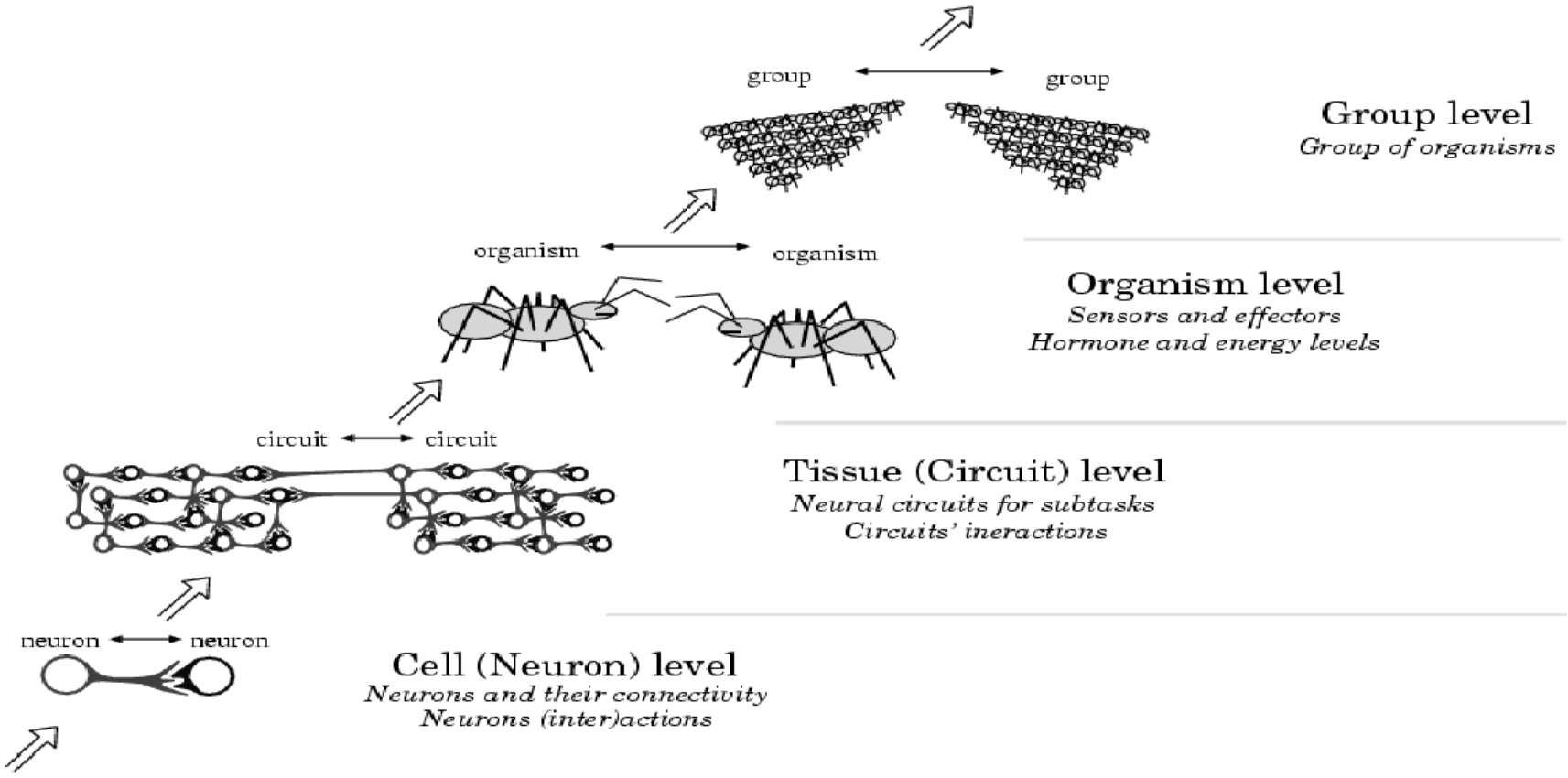
0	0	0	0	0	0	0	0	0
0	1	2	1	0	0	0	0	0
1	**1**	**2**	**1**	1	0	0	0	0
1	**2**	**4**	**2**	2	0	0	0	0
1	**2**	**5**	**3**	2	0	0	0	0
1	2	**2**	3	1	0	0	0	0
0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Structures réplicatives : La boucle de langton



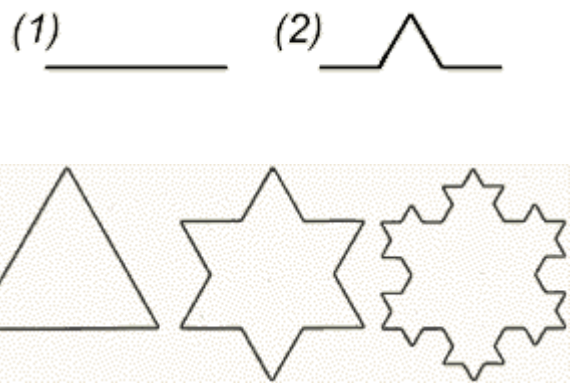
8 états
29 règles

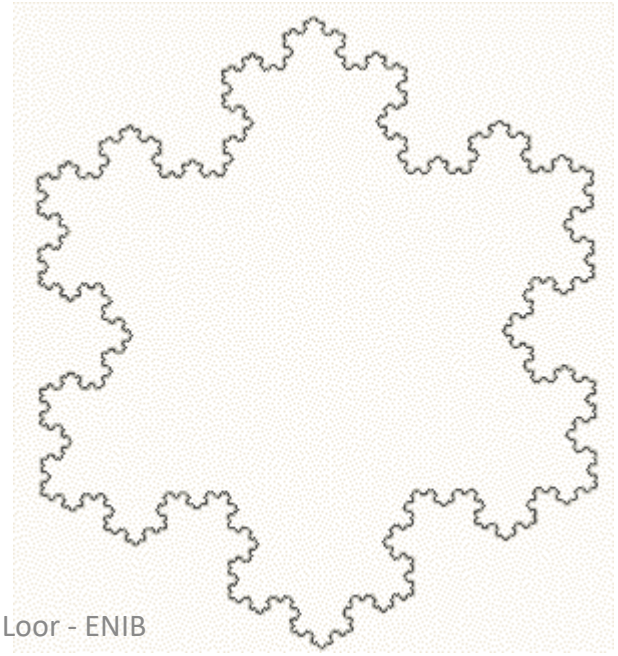
Émergence


group
group
Group level
Group of organisms
organism
organism
Organism level
Sensors and effectors
Hormone and energy levels
circuit
circuit
Tissue (Circuit) level
Neural circuits for subtasks
Circuits' ineractions
neuron
neuron
Cell (Neuron) level
Neurons and their connectivity
Neurons (inter)actions
Molecular level
Molecular concentrations

Récursion et génération

- Fonctions récursives
  - Ex: f(n) = f(n-1) + f(n-2) ; f(0) = 1; f(1) = 4
- Fractales
  - *Géométrie récursive – fonctions itératives – suites géométriques*

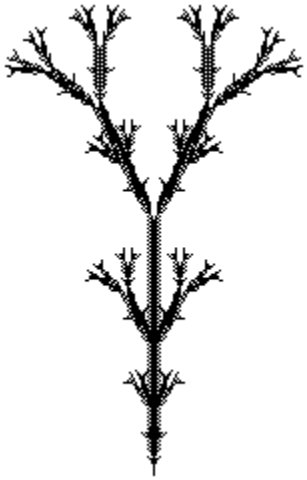


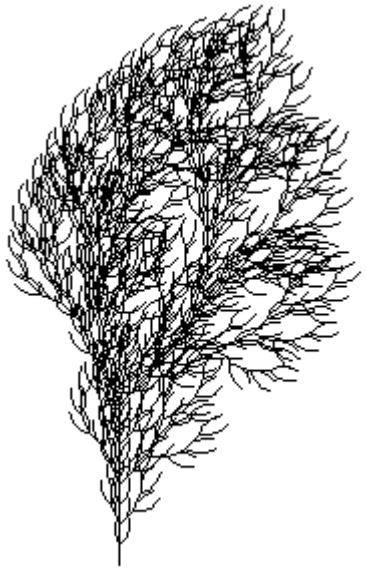


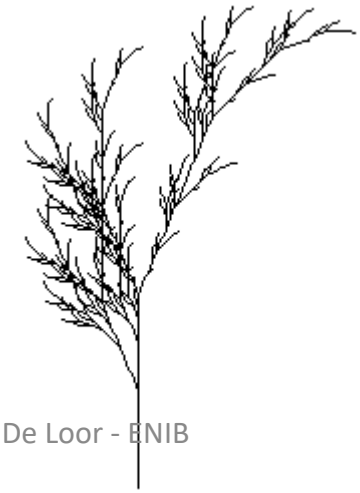

Module IAS - Pierre De Loor - ENIB

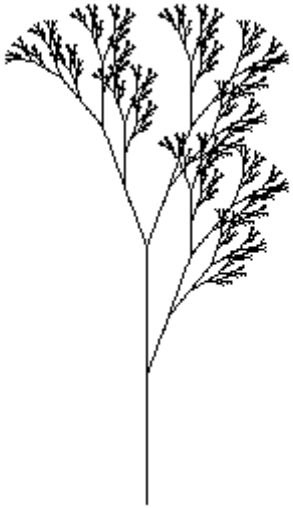
L-Systems

- Lindenmayer
- Principe: idem fractales
  - Initiateur : F
  - Générateur : F[+F]F[-F]F
  - Angle : 22.5



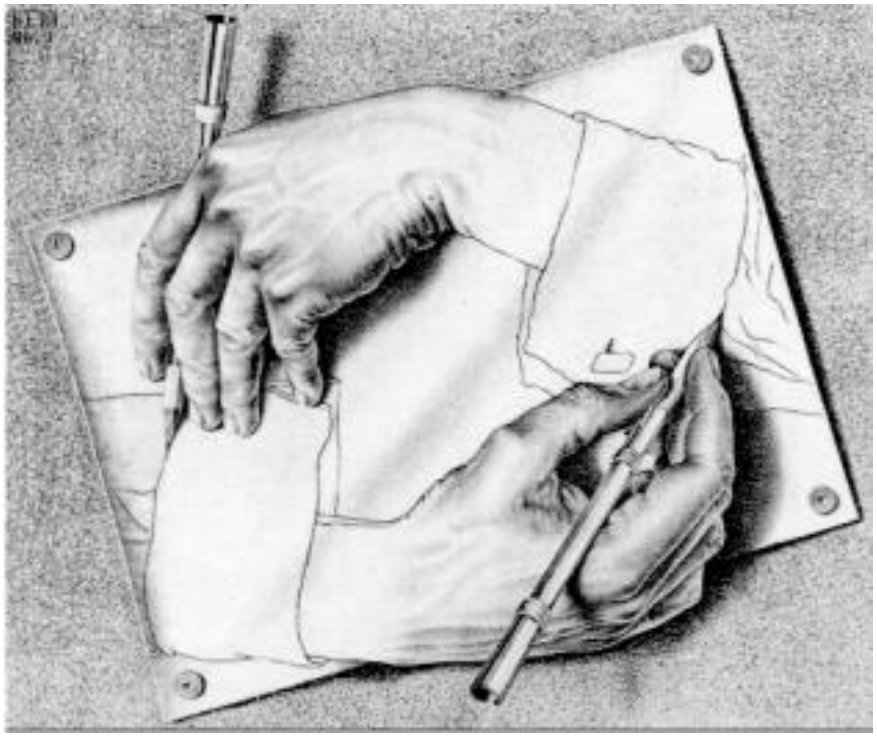






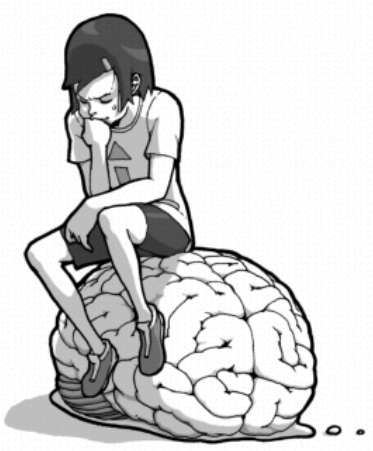
De la vie à la cognition : l'Enaction

- Paradigme des sciences cognitives: De la biologie à la conscience



Conséquences

- Le cerveau n'est pas un ordinateur mais un "interacteur autonome" et il est impossible de séparer le cerveau de son environnement pour comprendre la connaissance



- Il n'y a pas de représentations pré-données. Elles emergent

Aspects perceptifs

- Qu'est ce ça veut dire qu'il n'y a pas de représentations ?

- Ça veut dire que nous les constituons de façon personnelle en expérimentant le lien entre nos actions et nos perceptions.

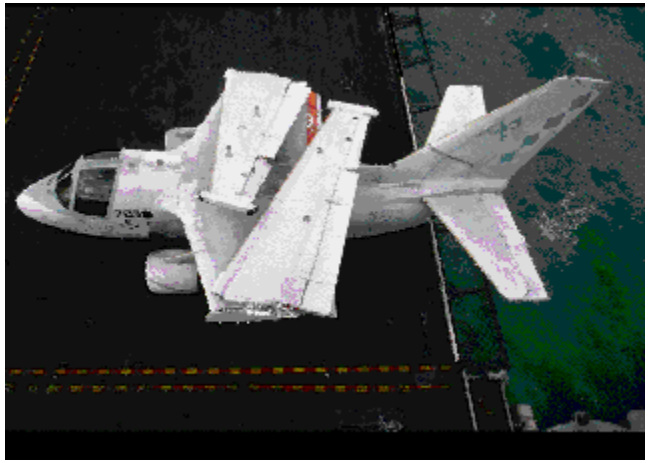
- Exemple de conséquences
  - Il n'y a pas d'image dans le cerveau
  - Il n'y a pas de concept de couleur dans le cerveau
  - …

Pas d’images dans le cerveau ? Exemple de “change blindness”

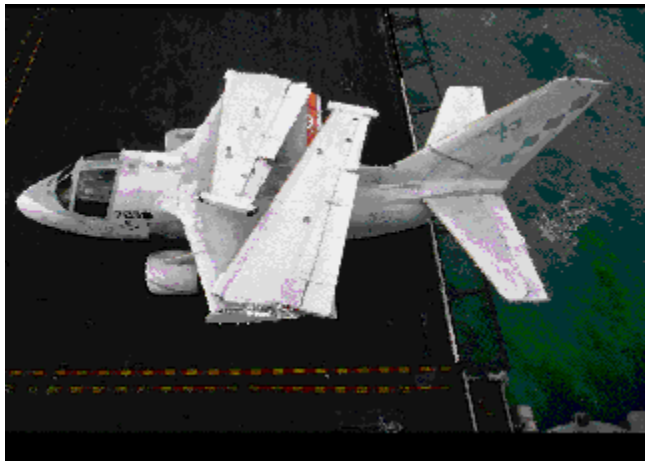
J.K. O'Regan, R.A. Rensink & J.J. Clark, Nature 1998

Change blindness (2)

Un autre exemple



Le changement …



Par contre, l'attention fait tout …

Corps, représentation … et conscience

L'Art invisible, comprendre la bande dessinée, Scott McCloud

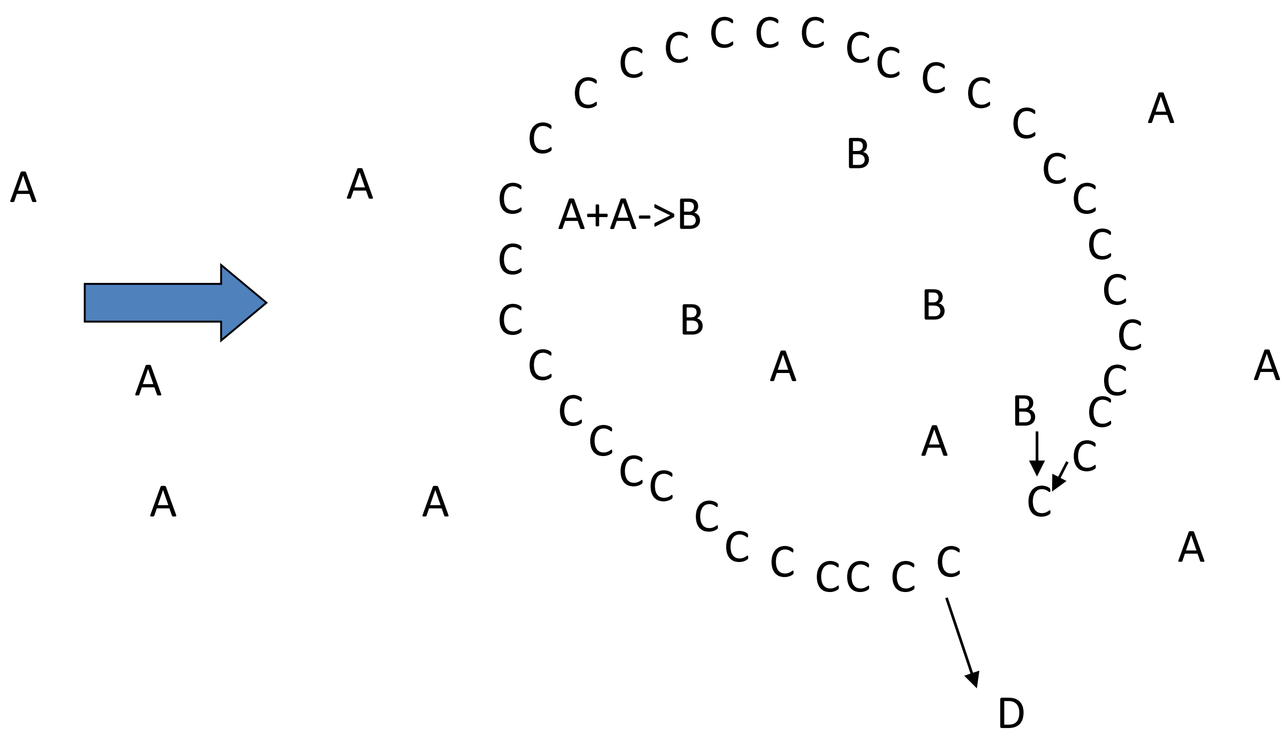
Conséquences

- Il faut partir de la notion d'organismes biologiques "auto-organisés" co-évoluant
- Il faut considérer le lien sensorimoteur des machines
- Pistes explorées :
  - Vie artificielle
  - Robotique

L’automate de Tesselation [Stewart & Bourgine 2002]

- Un système autopoiétique minimal
  - La membrane est constituée d’éléments C qui se désagrègent en D
  - Un élément B en contact avec un élément C peut s’y attacher s’il a la place et se transformer en C (bouche les trous)
  - Un élément B est créé à partir de deux éléments A à proximité d’un C (catalyseur)
  - La membrane est perméable aux A pas aux B
  - L’environnement est composé de A en libre circulation

L’automate de Tesselation



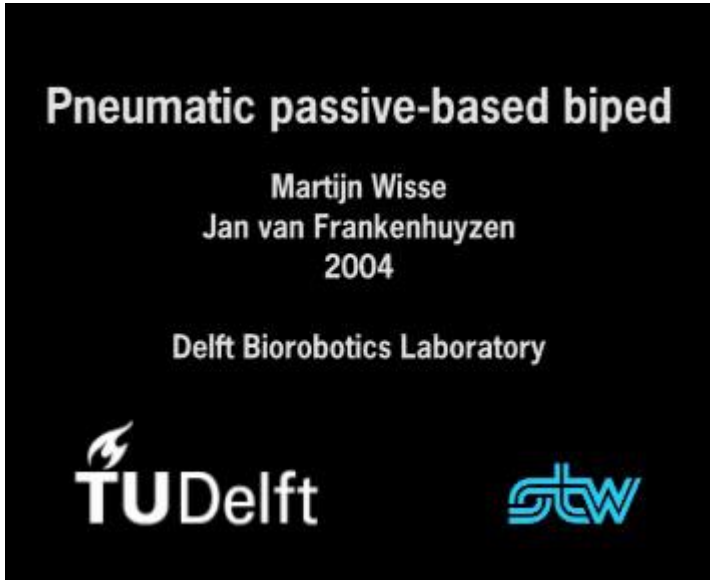
Cognition incarnée

Morpho-computation

- Robots wi-fi
  - Sans contrôleurs, sans capteurs

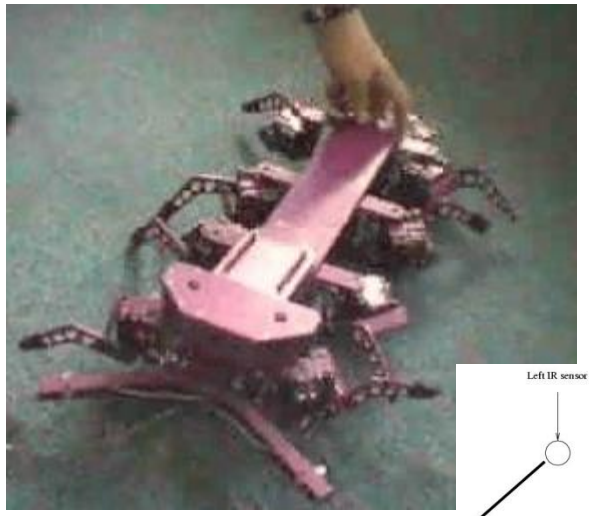
- Arguments :
  - La connaissance n'est pas que dans le cerveau
  - La morphologie influence le cerveau
  - Une morphologie adaptée à un environnement "soulage" le cerveau

Exemple

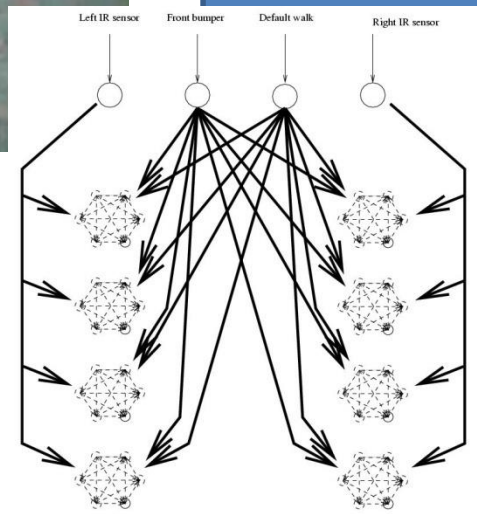

Pneumatic passive-based biped
Martijn Wisse
Jan van Frankenhuyzen
2004
Delft Biorobotics Laboratory
TU Delft
stw

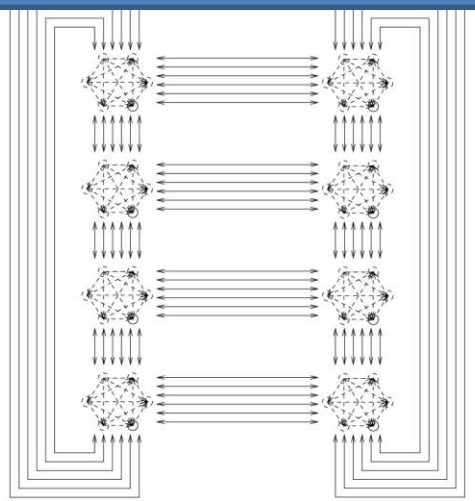
Cognition dynamique artificielle

– Dynamicité et récursivité



$$\tau_i \frac{dy_i}{dt} = -y_i + \sum_{j=1}^{N} w_{ji}\sigma_j(y_j) + I_i(t)$$





- Environment = perturbation

Bibliographie

- **Modélisation cognitive et résolution de problèmes,** *G. Caplat,* Press Polytechniques et universitaires romandes.
- **Ontologie et réutilisabilité : expérience et discussion,** *J.Charlet, B. Bachimont, J. Bouaud, P. Zweigenbaum,* Acquisition et ingénierie des connaissances, pp. 232-272, Springer Verlag, 1993.
- **Les activités mentales**, *Jean François Richard*, Armand Colin, 1990.
- **Intelligence artificielle et psychologie cognitive**, *Hervet Chaudet et Liliane Pellegrin,* Dunod 1998.
- **Machines à penser, une histoire de l'intelligence artificielle,** *Vernon Pratt,* PUF, 1987.
- **The essence of artificial intelligence,** *Alison Cawsey,* **1998.**
- **Apprentissage artificiel, concepts et algorithmes .** *Antoine Cornuéjols et Laurent Miclet*, Eyrolles, 2002.
- **Approche dynamique de la cognition artificielle,** *Guillot, A. & Daucé, E.Guillot, A. & Daucé, E., ed. (2002), Hermes-Lavoisier.*

- Maturana, M. & Varela, F. (1984), *Autopoiesis and Cognition, Reidel, Dordrecht.*
- McGeer, T. (1990), 'Passive dynamic walking', *International Journal of Robotics Research* ***9(2), 62--82.***
- McMullin, B. (2004), 'Thirty Years of Computational Autopoiesis: A Review', *Artificial Life* ***10, 277--295.***
- Nolfi, S. & Floreano, D. (2000), *Evolutionnary Robotics: The biology, intelligence, and technology of self-organizing machines, MIT Press/Bradford Books.*
- Pfeifer, R. & Scheier, C. (1999), *Understantding Intelligence, MIT Press.*
- Sigaud, O. & Flacher, F. (2002), *Approche dynamique de la cognition artificielle, Hermes, Traité des sciences cognitives, chapter Vers une approche dynamique de la sŭlection de l'action, pp. 163--176.*
- Whitaker, R. (2001),'Tutorials 1 and 2 on Autopoièsis and Enaction', http://www.enolagaia.com.

Annexes

- A1. Demonstration automatique
- A2. Réseaux sémantiques
- A3. Raisonnement à Base de Cas : exemple
- A4. A*

A.1.Démonstration automatique

- On veut démontrer : auDessus(b, table), on affirme le contraire
- déduction / résolution

¬sur(U,V) ∨ auDessus(U,V) (1)

¬auDessus(X,Y) ∨ ¬auDessus(Y,Z) ∨ auDessus(X,Z) (2)

sur(b,a) (3)

sur(a,table) (4)

¬ auDessus(b, table) (5)

Inférence :

{A ∨ C, ¬C ∨ B} |=A ∨ B

UNIFICATION : instanciation de X à b
de Z à table

résolution

¬auDessus(b,Y) ∨ ¬auDessus(Y,table) remplace (2) et (5) -> (6)

Démonstration automatique

$\neg sur(U,V) \vee auDessus(U,V)$ (1)

$sur(b,a)$ (3)

$sur(a,table)$ (4)

$\neg auDessus(b,Y) \vee \neg auDessus(Y,table)$ (6)

V à table
U à Y

$\neg sur(Y,table) \vee \neg auDessus(b,Y)$ (7)

(1) + (7) instanciation de U avec b + unif : $\neg sur(b,V) \vee \neg sur(Y,table)$ (8)

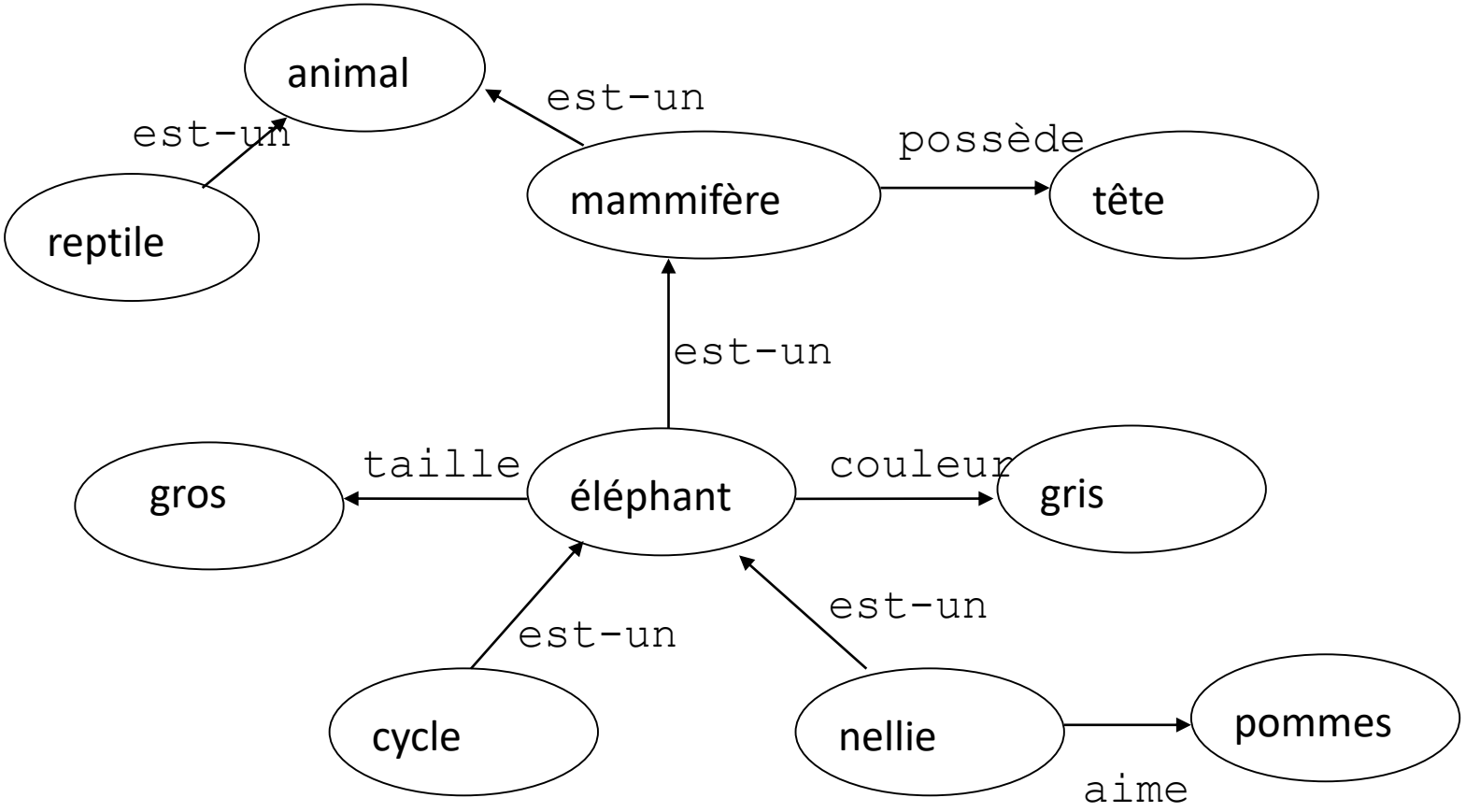
(3)+(8) instanciation de A avec y + unif : $\neg sur(a,table)$ (9)

Démonstration automatique

sur(a,table)	(4)
¬ sur(a,table)	(9)
FALSE	(10)

- Preuve par déduction/réfutation

A.2 Exemple de réseau sémantique (Collins et Quillians 1969)

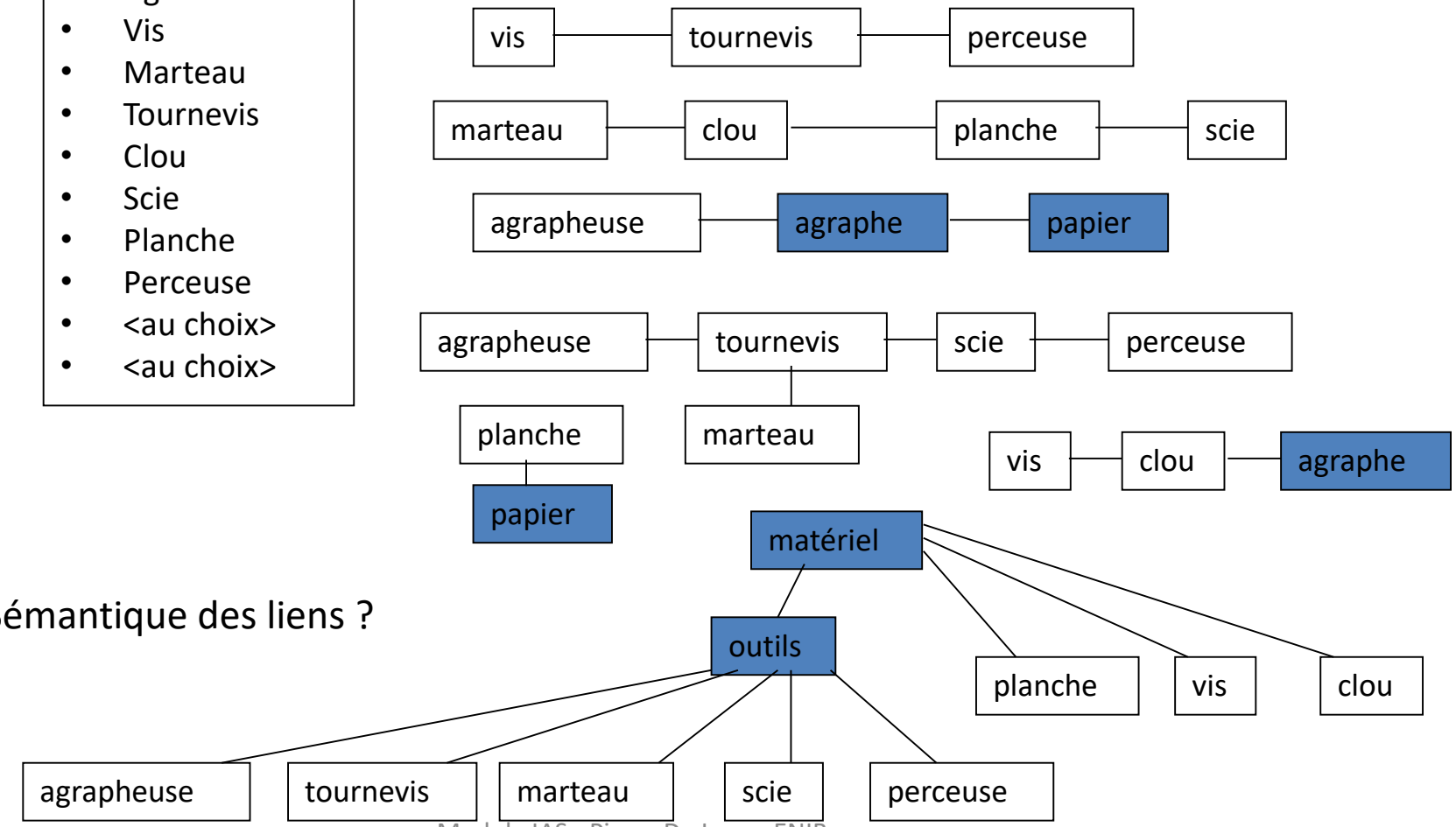


Faites votre réseau

- Agrafeuse
- Vis
- Marteau
- Tournevis
- Clou
- Scie
- Planche
- Perceuse
- <au choix>
- <au choix>

Faites votre réseaux

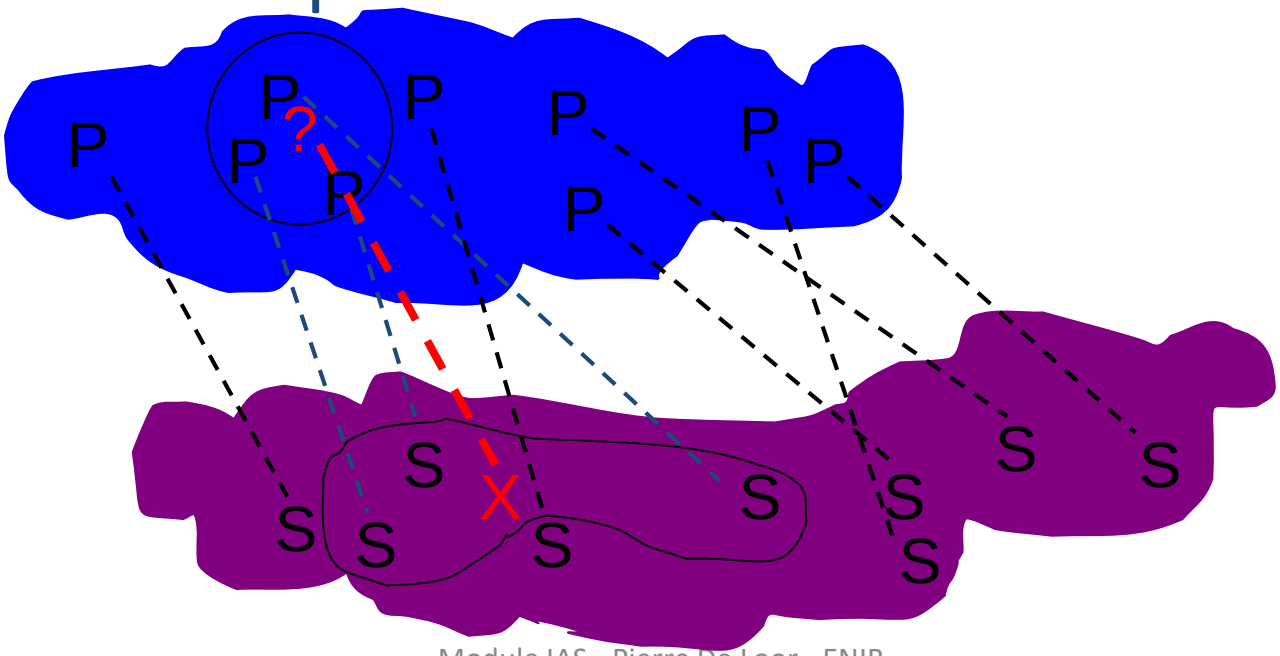
- Agrafeuse
- Vis
- Marteau
- Tournevis
- Clou
- Scie
- Planche
- Perceuse
- <au choix>
- <au choix>



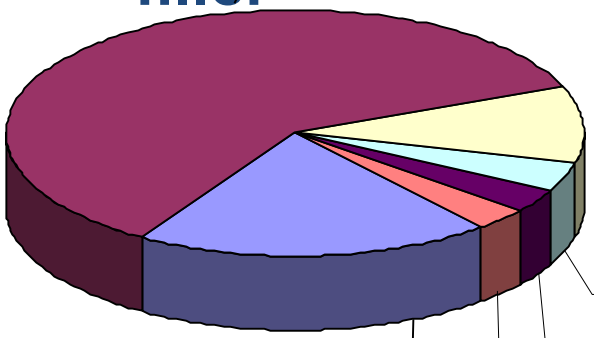
Sémantique des liens ?

A3 : CBR Assumption

- New problem can be solved by
  - retrieving similar problems
  - adapting retrieved solutions
- Similar problems have similar solutions



Exemple : What's in a Tablet?



filler — provides bulk to be large enough to handle and compress (~65%)

binder — makes it cohesive to hold together

lubricant — enables it to come out of the die

disintegrant — allows rapid break down after swallowing

drug — active ingredient (~25%)

surfactant — aids wetting and dissolution of drug

Tablet Formulation Problem

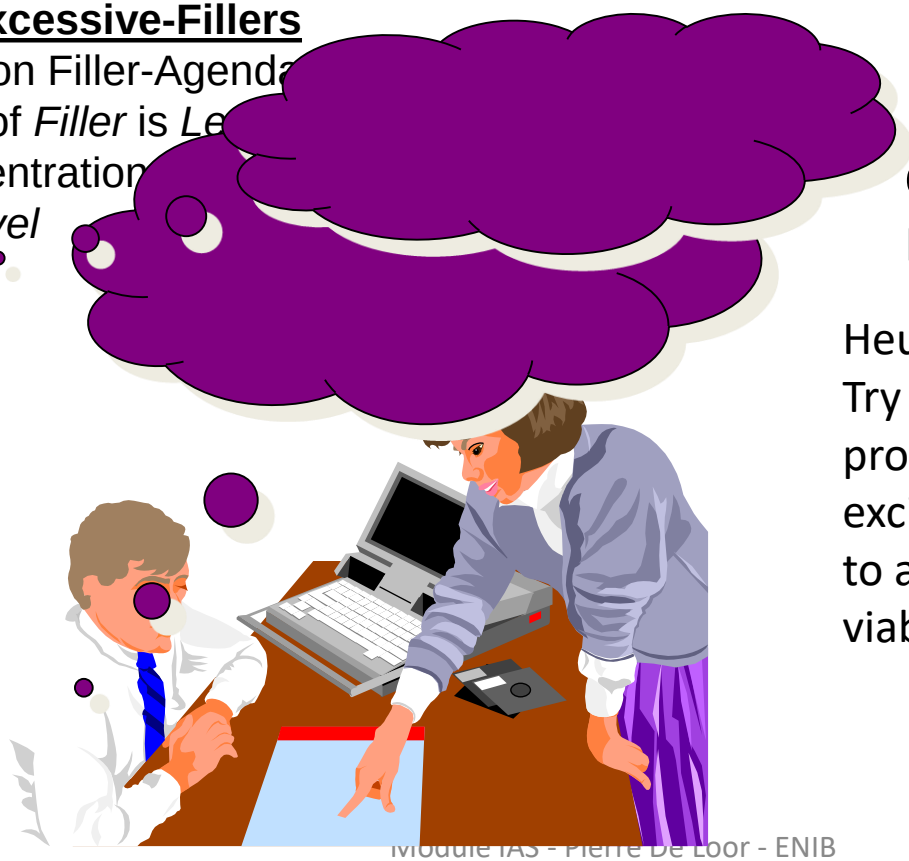
- Given:
  - physical and chemical properties of a drug
  - desired dose
- Knowing:
  - properties of available excipients
- Goal:
  - choose 5 excipients and their quantities
  - which achieve the desired mechanical and chemical properties of the tablet

Solution		
filler	DCP	92.3%
binder	GEL	2.1%
lubricant	MGS	1.0%
disintegrant	CRO	2.1%
surfactant	SLS	0.3%

Tablet Formulation Knowledge

Get-Insoluble-Filler
IF: Reqd-Filler-Solubility has value Insoluble
Filler is-on Filler-Agenda
Solubility has value *Sol* in *Filler*
Slightly-Soluble has value *Slightly-Soluble*
Sol < Min-Val (*Slightly-Soluble*)
THEN refine Filler to be *Filler* in Formulation

Remove-Excessive-Fillers
IF: *Filler* is-on Filler-Agenda
Max-Level of *Filler* is *Le*
Filler-Concentration
Conc > Level
THEN ...

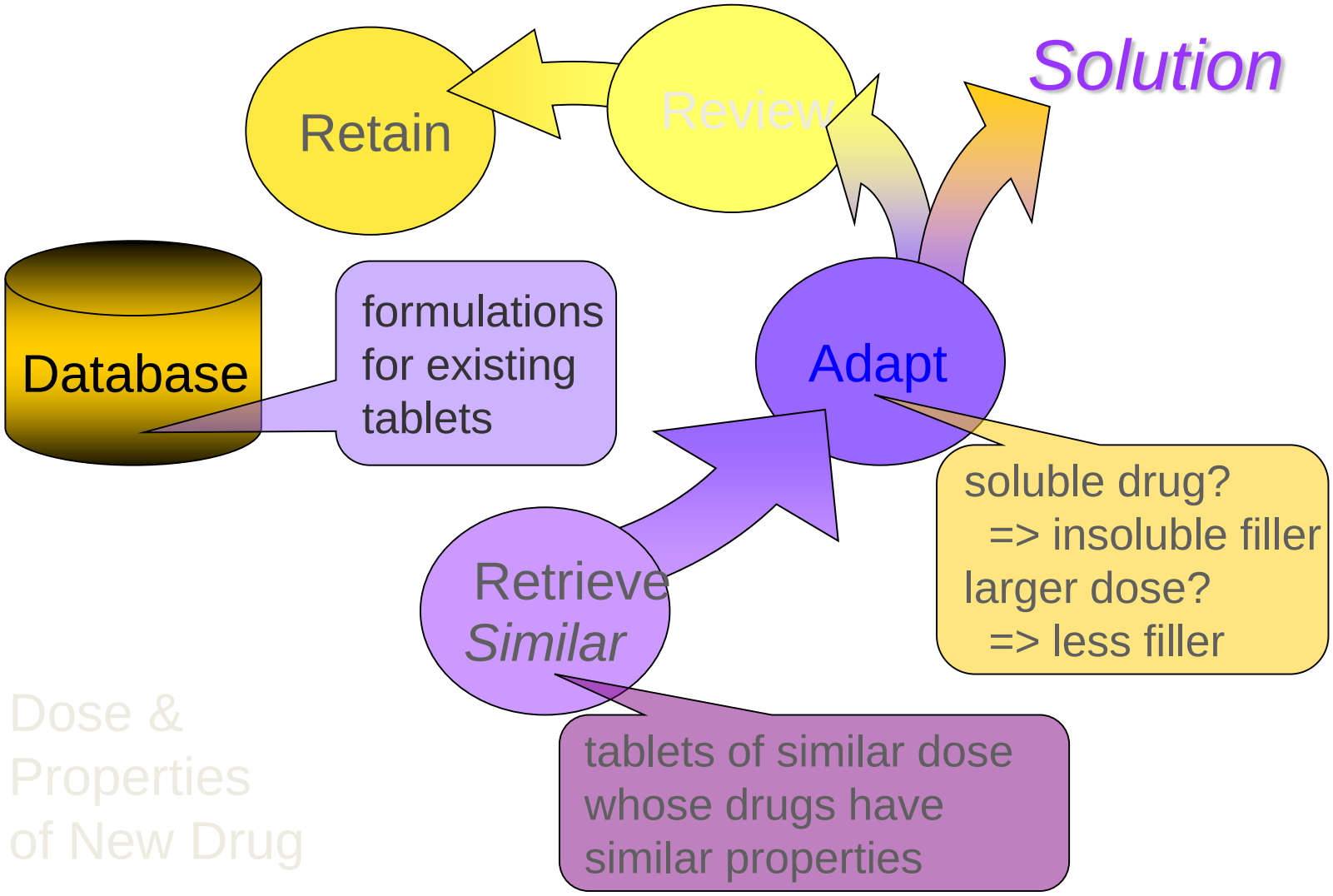


Drug Properties
Excipient Properties
Drug/Excipient Stabilities

Chemical Relationships
Physical Relationships

Heuristics
Try to balance physical properties with stable excipients
to achieve a tablet with viable properties

CBR for Tablet Formulation



Why was filler X chosen?

- The tablet in the case-base whose
  - drug properties are most similar
  - dose is most similar

  is Drug-Y-50 and its filler is Z

- However adaptation is needed
  - because of a significant difference
  - the stability of Z with the new drug is much lov

- Adaptation proposes filler X instead:
  - greater stability with new drug
  - similar properties to Z

A4: Exemple : A*

Paramètres :

START : nœud de départ
GOAL : nœud d'arrivée
OPEN : liste des nœuds à traiter (en général : représenté comme une file de priorité)
CLOSED : liste des nœuds traités
N : nombre de nœuds
h(i) : distance estimée d'un nœud *i* au nœud d'arrivée ($i \in \{1, 2, ..., N\}$)
g(i) : distance réelle d'un nœud *i* au nœud de départ ($i \in \{1, 2, ..., N\}$)
f(i) : somme des distances h(i) et g(i)
parent() : parent d'un nœud *i* (*parent(x)* donne la position dans *parent()* du nœud précédant *x*))

Initialiser la liste *OPEN* à vide
Initialiser la liste *CLOSED* à vide

Ajouter *START* à la liste *OPEN*
Tant que la liste *OPEN* n'est pas vide
Retirer le nœud *n* de la liste *OPEN* tel que *f(n)* soit le plus petit
Si *n* est le *GOAL* retourner la solution ;
Sinon ajouter *n* à *CLOSED*
Pour chaque successeur *n´* du nœud *n*
Heuristique H = estimation du coût de *n´* au *GOAL*
Stocker dans G_tmp *g(n´)*, le coût *g(n)* + le coût pour aller de *n* à *n´*
Stocker dans F_tmp *f(n´)*, *g(n´)* + H ; c'est l'heuristique
Si *n´* se trouve déjà dans *OPEN* avec un *f(n´)* meilleur on passe au point *n´* suivant
Si *n´* se trouve déjà dans *CLOSED* avec un *f(n´)* meilleur on passe au point *n´* suivant
Mettre *n* dans *parent(n')*
Mettre G_tmp dans *g(n')*
Mettre F_tmp dans *f(n´)*
Retirer *n´* des deux listes *OPEN* et *CLOSED*
Ajouter *n´* à la liste *OPEN*

Learning

- Case-base
  - inserting new cases into case-base
  - updating contents of case-base to avoid mistakes
- Retrieval Knowledge
  - indexing knowledge
    - features used
    - new indexing knowledge
  - similarity knowledge
    - weighting
    - new similarity knowledge
- Adaptation knowledge

CBR Resources

- CBR Tools
  - ReCall (www.isoft.fr), Orenge (www.tecinno.com) Kaidara (www.kaidarausa.com)
- CBR Websites
  - www.ai-cbr.org
  - www.aic.nrl.navy.mil/~aha/
  - www.scms.rgu.ac.uk/research/kbs/kacbd/
- CBR Conferences
  - ICCBR'01: www.iccbr.org/iccbr01/
  - UK-CBR'01: www.ai-cbr.org/ukcbr5/
  - ECCBR 2002: www.scms.rgu.ac.uk/eccbr2002/

Schémas de connaissances/frame/cadres sémantiques (Minsky 75)

- Modélisation
  - Des prototypes (cadres/schémas)
  - Des situations « prototypiques » (scénarios/scripts)
  - Intègre la notion d'information manquante, partielle ou l'exception

- Blocs de connaissances
  - Insécables
  - Autonomes (≠concepts)
  - Un schéma « porte » tout son sens
  - Cadre qui regroupe des concepts relativement à un contexte
    - schéma « restaurant » ≠ concept « restaurant »

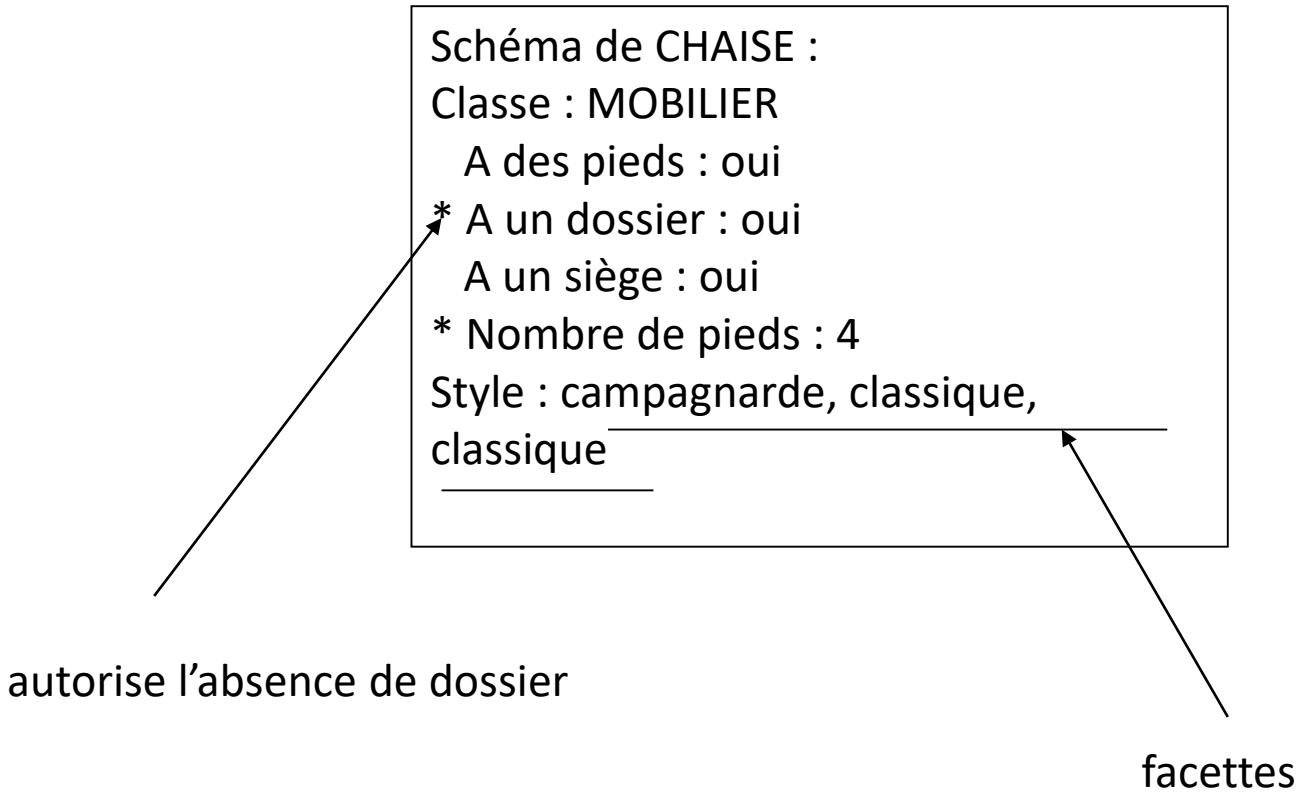
Situation prototypique

- Frame/Script « Manger au restaurant »

Eléments : (restaurant, argent, nourriture, menu, tables, chaises, attente)
Rôles : (gens affamés, serveurs, maîtres d'hôtel)
Horaires : (heures d'ouverture du restaurant)

Séquence d'événements :
Initial : script Rentrer au restaurant;
Puis : si (signal-Appeler le serveur), alors script Appeler le serveur;
Puis : script commander;
Puis : script Manger à moins que (longue attente), alors script Se plaindre et script Sortir mécontent;
Puis : si(bonne nourriture), alors script compliment au chef;
Puis : script Payer facture;
Puis : script Quitter le restaurant

Formalisation des frames



Si l'on parle d'une chaise en général, elle aura 4 pieds et un dossier

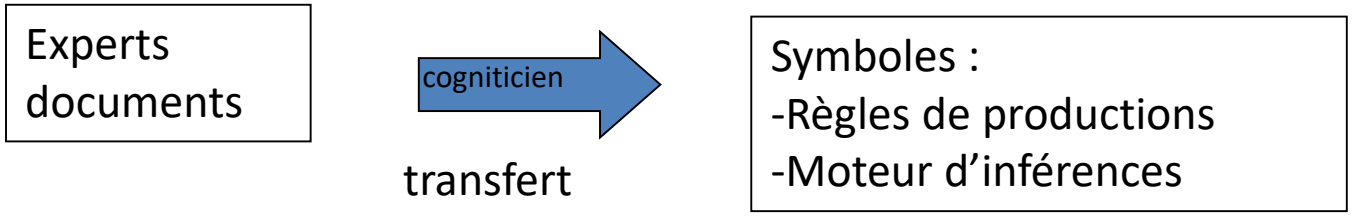
Comparaison

- Avantage de la logique/manque des modèles graphiques
  - notion de condition
  - notion de preuve
  - inférence automatique
  - frame et réseaux sémantiques peuvent être modélisés en logique
- Avantage des modèles graphiques/manque de la logique
  - « intuitifs »
  - pratique pour discuter
- Manques communs
  - inférence systématique pas si naturelle
  - Raisonnement approximatif ? Incertain ?
  - Quels sont les « bons » concepts ?

Exemples de domaines

- Medecine
  - Le médecin se rappelle de patients ayant eut les mêmes symptomes (surtout pour les symptômes rares)
- Loi
  - La loi est établie sur des précédents …
  - Les cas historiques sont consultés
- Management
- Financial
  - Les performances sont prédites des résultats passés

Ingénierie des connaissances



- Le transfert n'est pas du tout évident :

  - La vision d'ensemble est perdue dans les règles
  - Un expert utilise des heuristiques implicites

Quelques exemples d'expression de la connaissance

- Modèles déterministes
  - Exemple : logique des prédicats

$\forall x \in \text{Chien}, \forall y \in \text{Homme} : \text{possède}(y,x) \rightarrow \text{aimeSonMaitre}(x,y) \wedge \text{aimeSonAnimal}(y,x)$

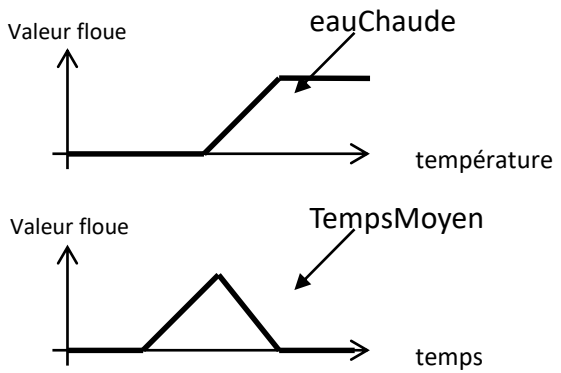
- Modèles probabilistes
  - Exemple : théorème de Bayse

p(froid|neige)=p(neige-froid)/(p(neige-froid)+p(neige-pas-froid))

- Modèles flous
  - Exemple : Logique de Zadeh

EauChaude $\wedge$ TempsMoyen = OeufBon

OeufBon = min (température eau, temps)



Logique du premier ordre

- Choix d'une représentation en logique du premier ordre -> raisonnement à un niveau différent

  nourriture(pâtes)
  homme(tom)
  couvert(fourchette)
  mange(tom,pâte,fourchette)
  « tom mange des pâtes avec une fourchette »

  sorte-de(nourriture, pâtes)
  sorte-de(homme, tom)
  sorte-de(couvert, fourchette)
  « la nourriture est mangée par les hommes avec des couverts »

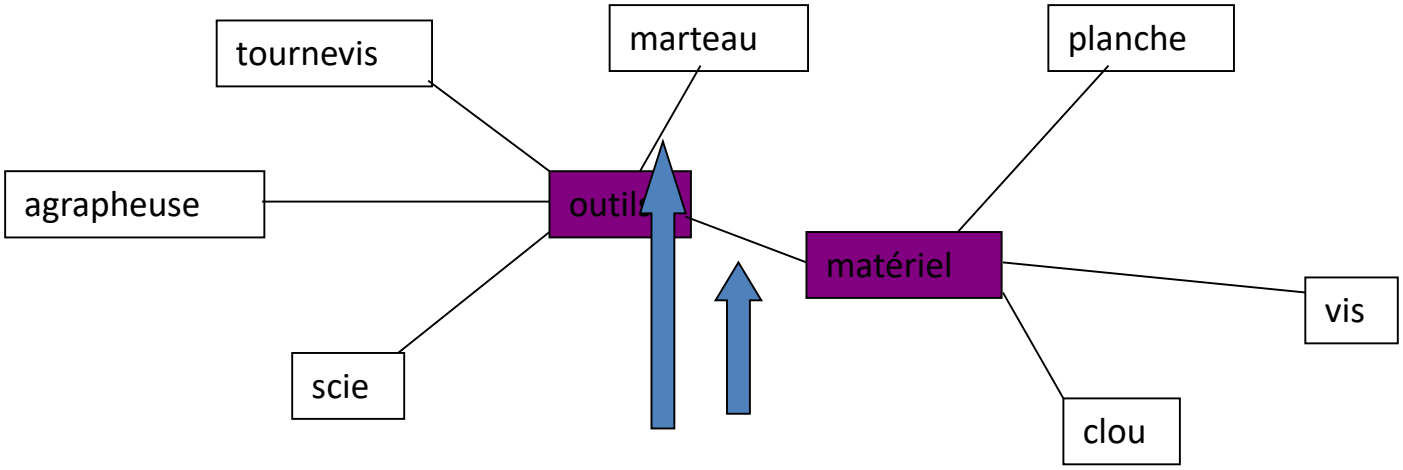
  egal(nature(pâte, nourriture))
  egal(nature(tom, homme))
  egal(nature(fourchette, couvert))
  « les pâtes sont-elles de la même nature que les hommes »

Réseaux sémantiques : La catégorisation

- Notion de concepts (catégorisation)
  - Un concept est une entité cognitive qui possède un sens
  - Réseau d'association de concepts // proximité sémantique et spatiale

- Quillian(1968)
  - Travail sur la mémoire sémantique
    - Comment est organisée notre connaissance ?
    - Comment fonctionne notre mémoire ?

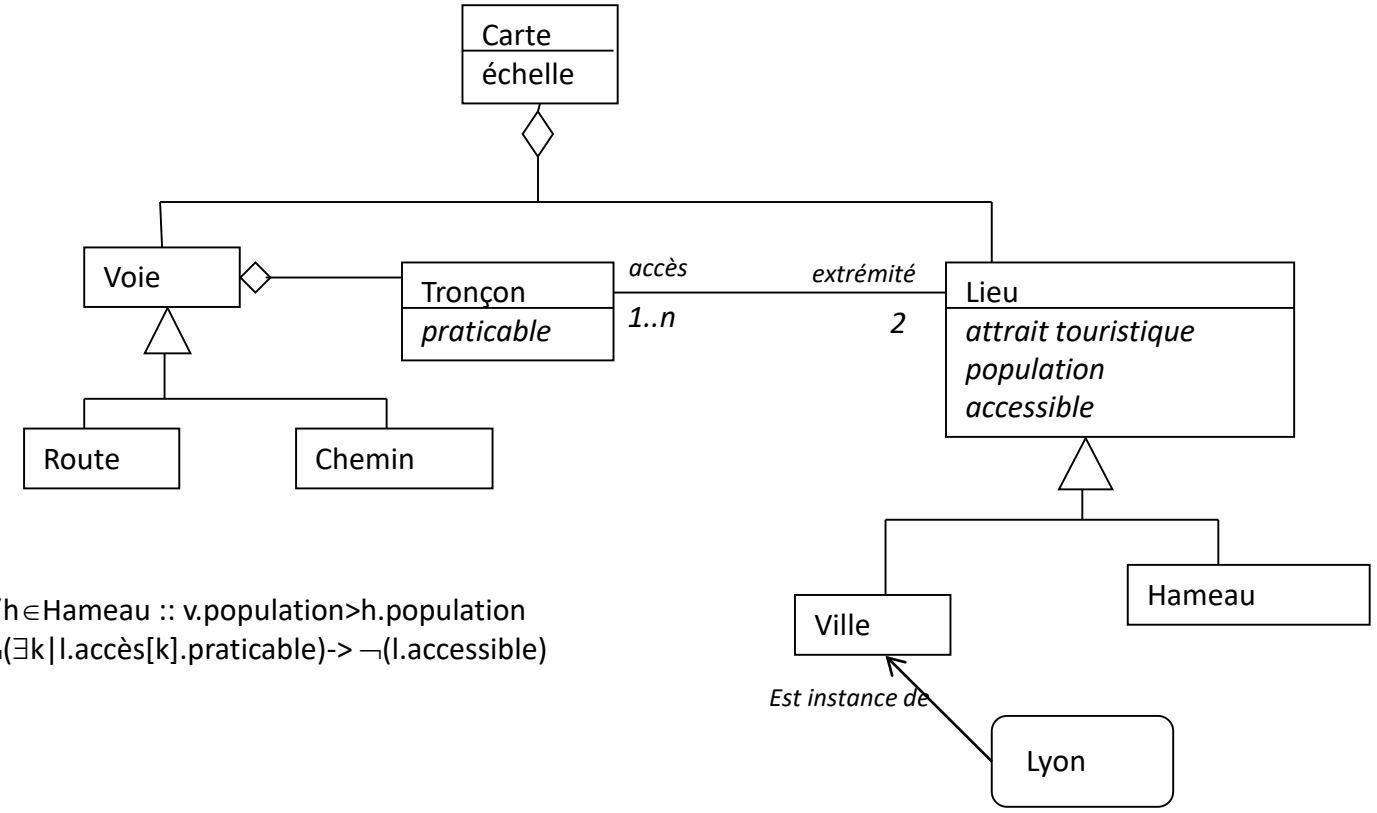
Sémantique implicite = danger



Pas de signification claire des arcs : pas de sémantique claire

Diagrammes de classe

- Modèle de connaissances + spécification formelle

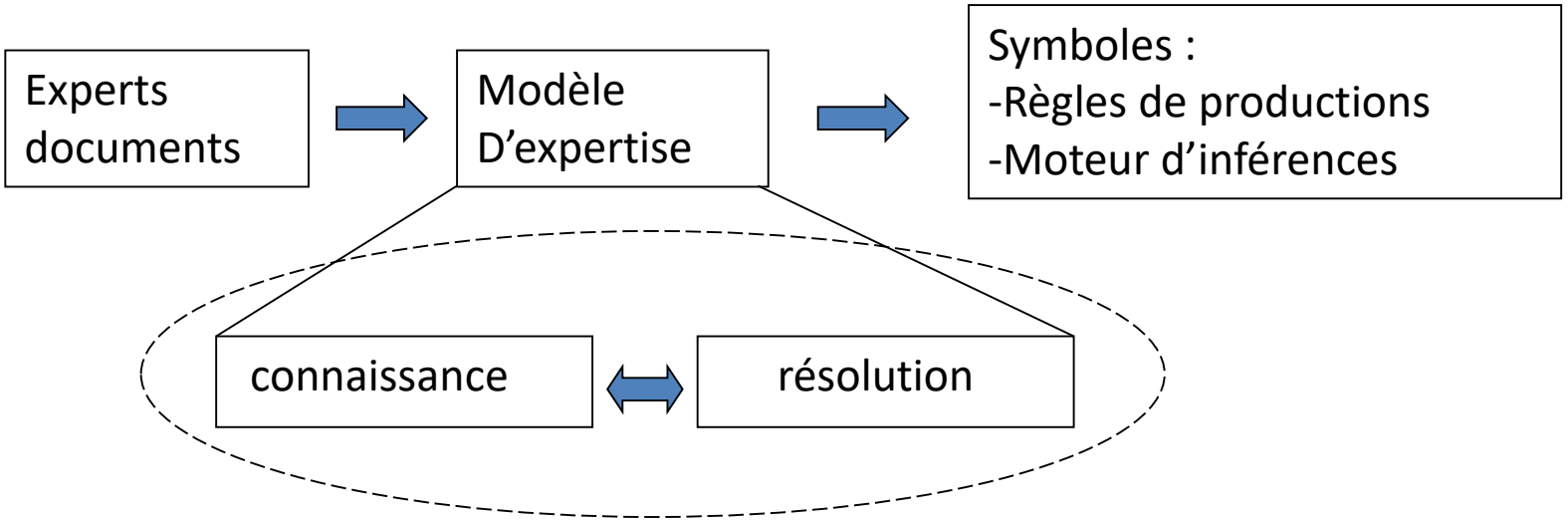


E1 : $\forall v \in Ville, \forall h \in Hameau :: v.population > h.population$

E2 : $\forall l \in Lieu :: \neg(\exists k | l.accès[k].praticable) \rightarrow \neg(l.accessible)$

Utilisation d'un modèle d'expertise

Approche descendante :



Problème :

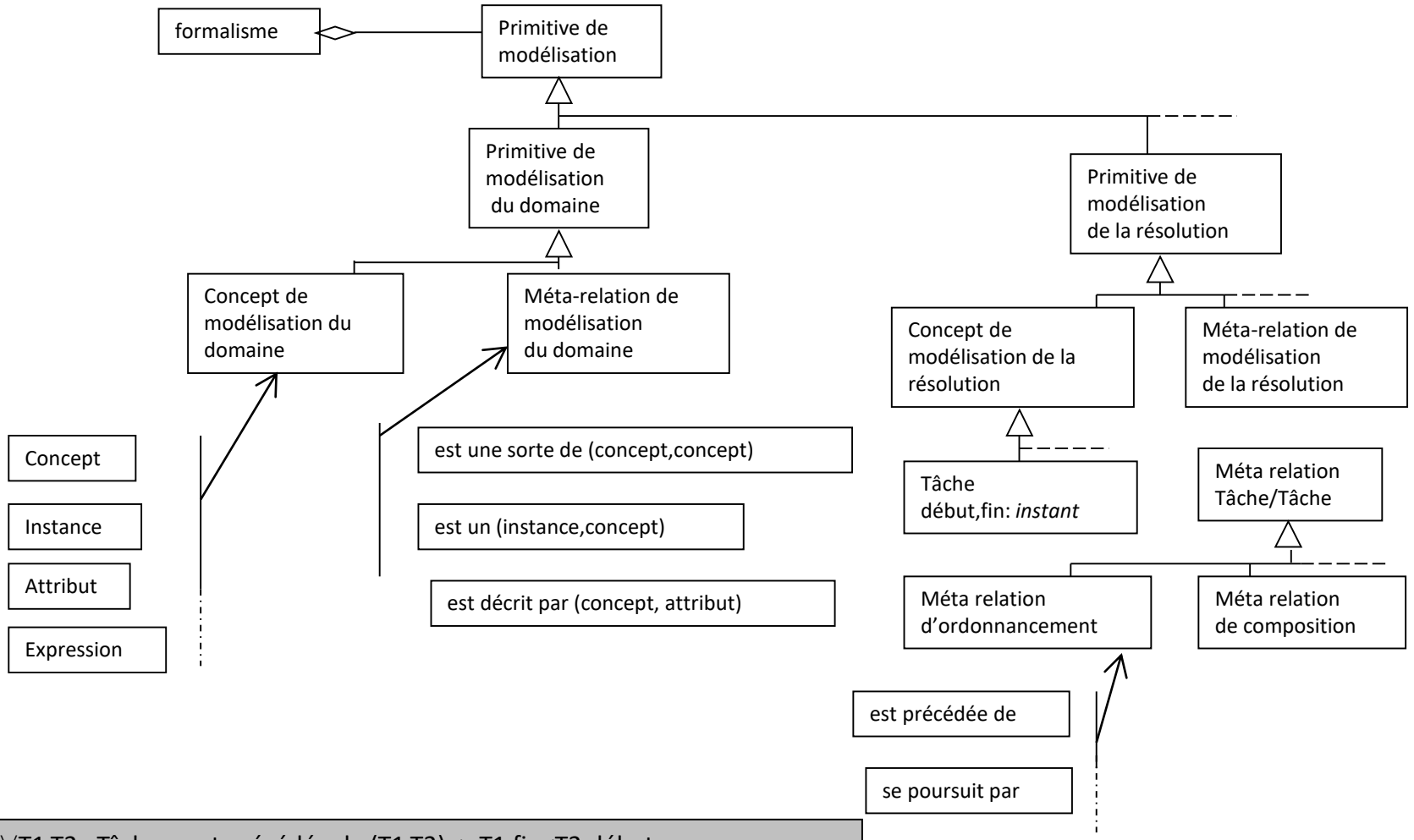
- choix du modèle d'expertise :
  - le m. de résolution définie celui des connaissances
  - le m. des connaissances est nécessaire au choix du modèle de résolution

KADS : methodology for knowledge-based systems.

- Kads
  - Méta-modèles de connaissances
  - Méta-modèles de résolution
  - Méthodologie de choix des modèles (critères, taxinomie)
  - Conception descendante

- CommonKads
  - Croisement de modèles

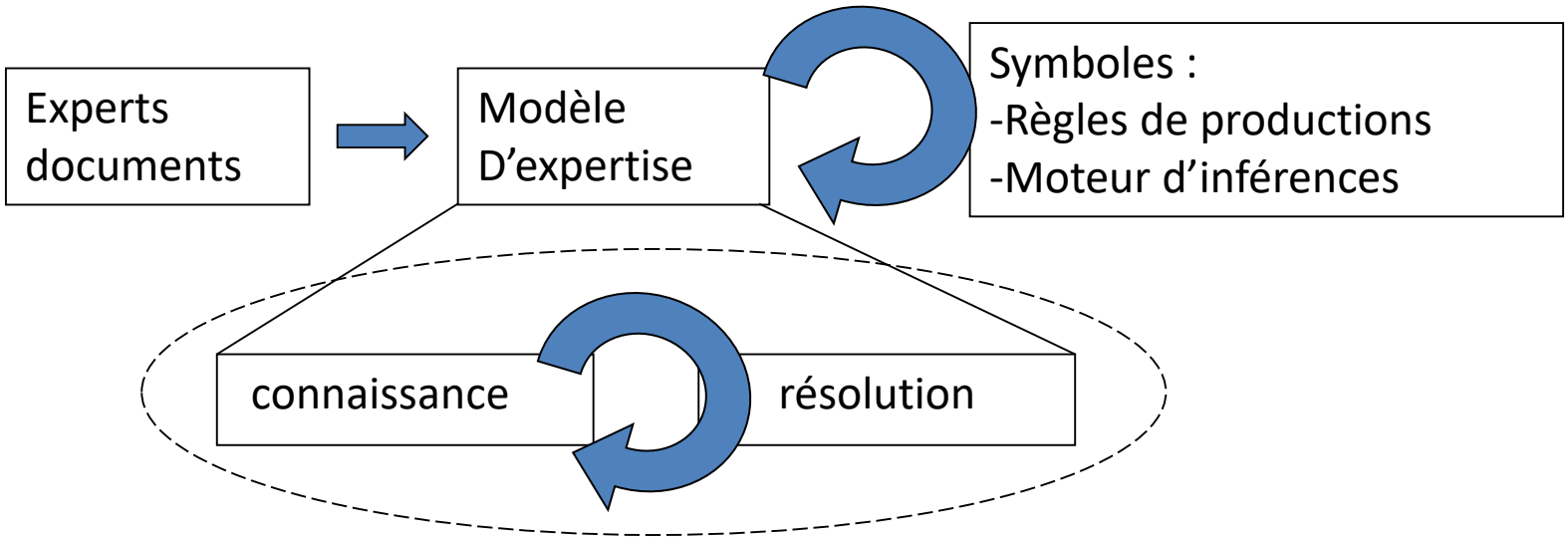
Exemple de méta-modèle de connaissance



∀T1,T2∈Tâche :: est précédée de (T1,T2) -> T1.fin<T2.début
…

Approches constructivistes

Construction incrémentale :



Notion d'ontologie

- Propre à un domaine
- l'intégrité de l'ontologie étant invariante d'une application à l'autre, il suffit de distinguer parmi les connaissances, celles relevant spécifiquement du domaine, I.E. l'ontologie, pour que l'acquisition puisse être effectuée une fois pour toute, la permanence ontologique du monde assurant ainsi la réutilisabilité (Charlet et al 96)

Attribut discriminant

- Théorie de l ’information
  - attribut apportant le plus de gain à l ’information
  - défini à partir de la notion d ’entropie
  - entropie d ’un ensemble d ’exemples
    - S exemples positifs ou négatifs
    - p_+: proportion d ’exemples positifs
    - p_- : proportion d ’exemples négatifs
    - Entropie(S) = - (p_+ x $\log_2$ (p_+)) - (p_- x $\log_2$ (p_-))

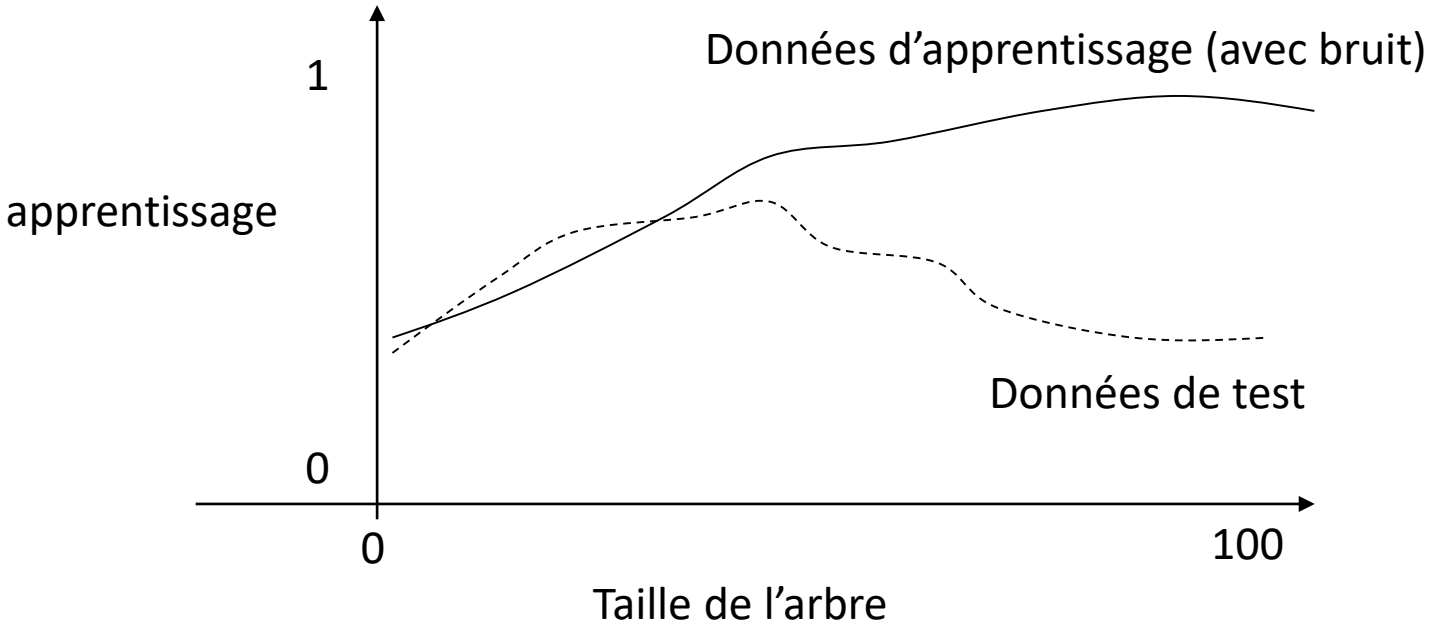
Sur-Apprentissage

- Si il y a peu d'exemples, un exemple erroné peut introduire un biais : arbre de décision sous-optimal.

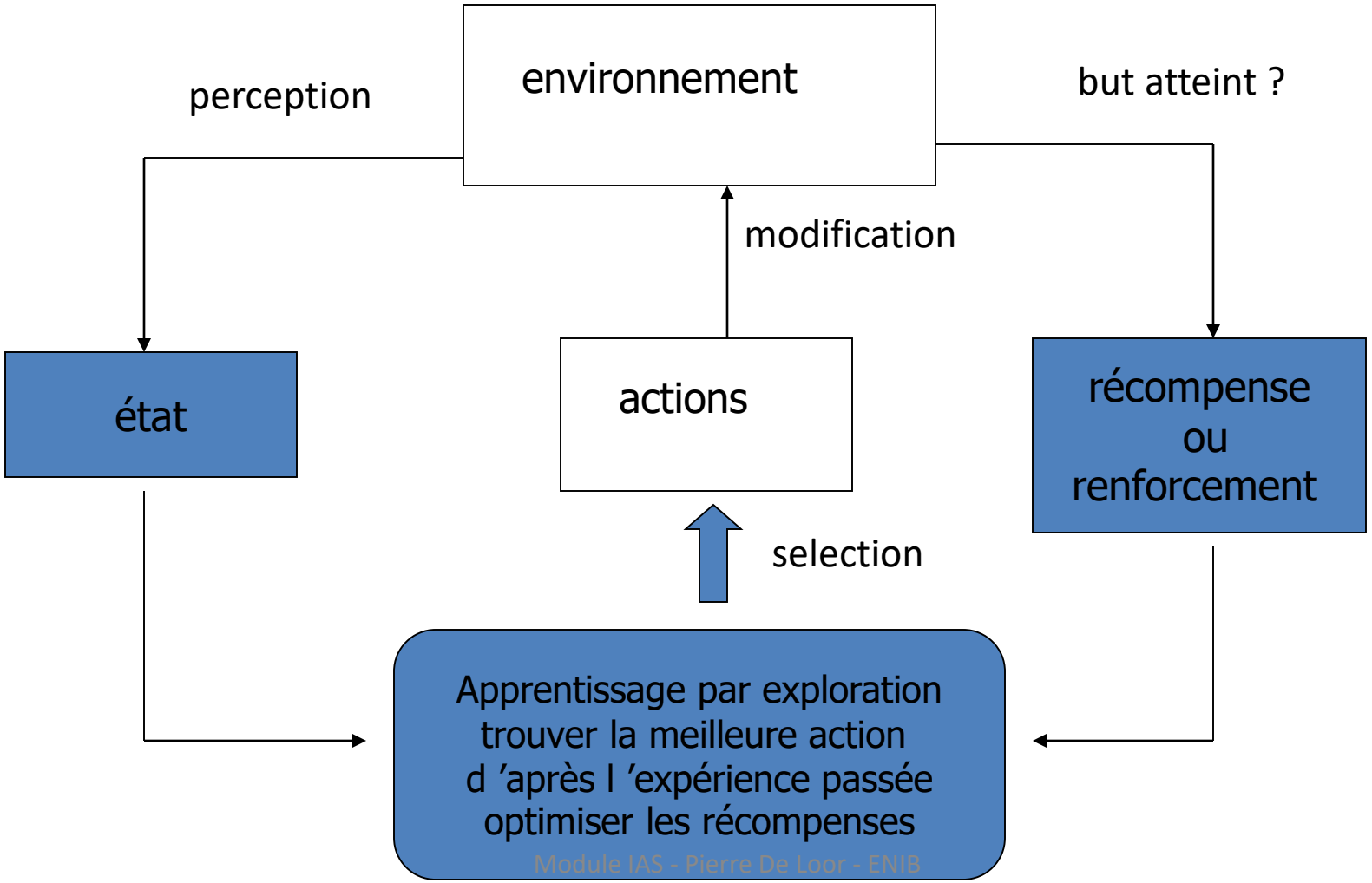
- L'algorithme apprend le bruit : arbre de décision grandissant.

Sur-apprentissage

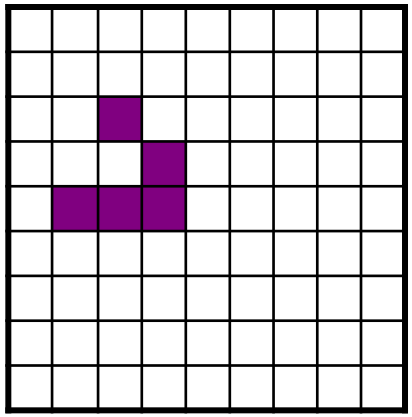
- Exemple [machine Learning]

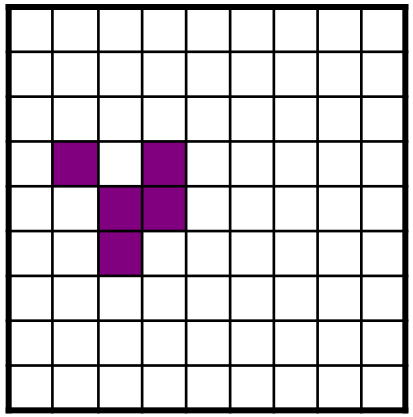


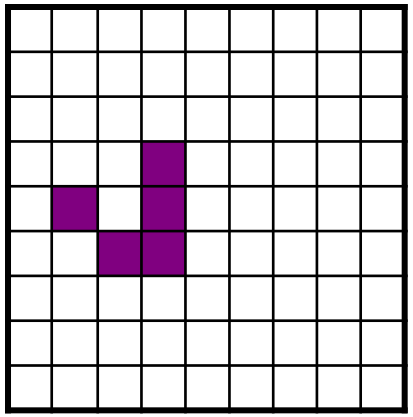
Apprentissage par renforcement

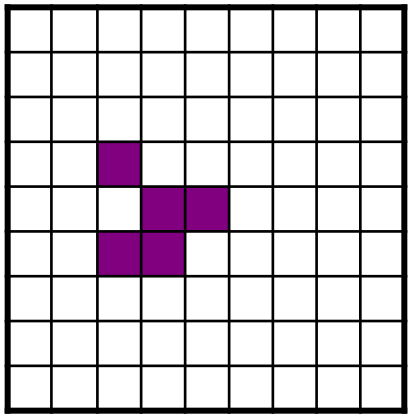


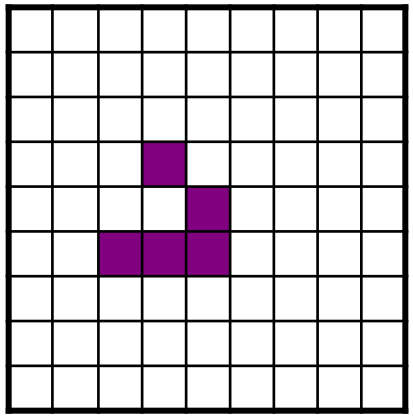
Jeu de la vie et autopoièse





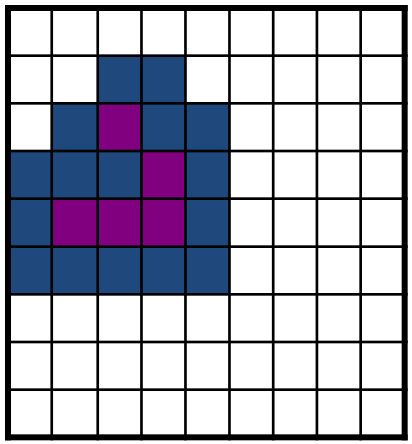
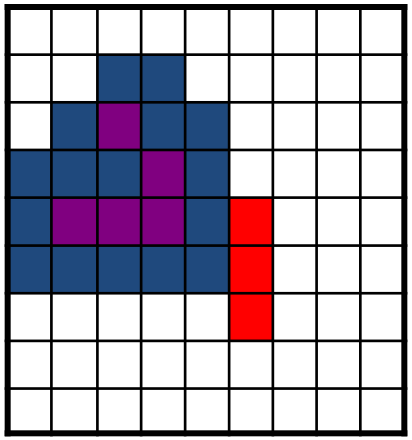
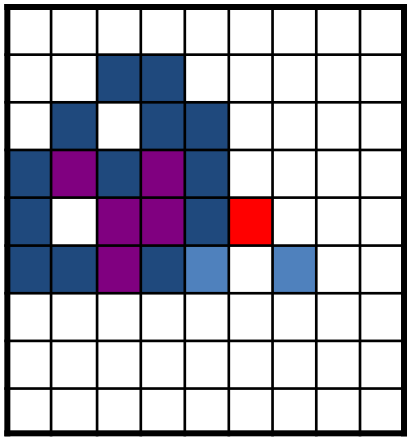
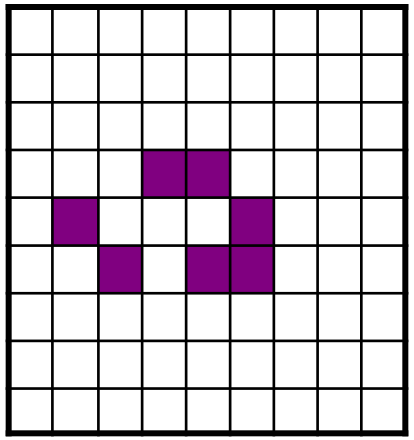
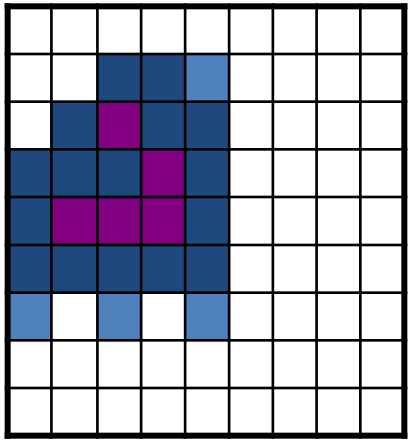
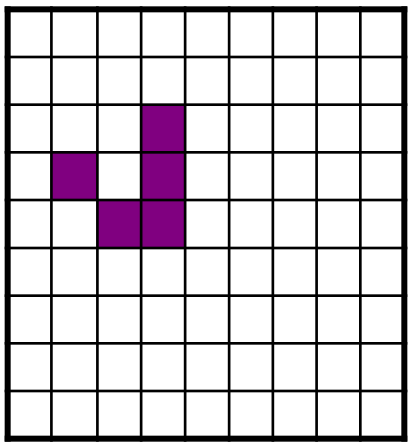
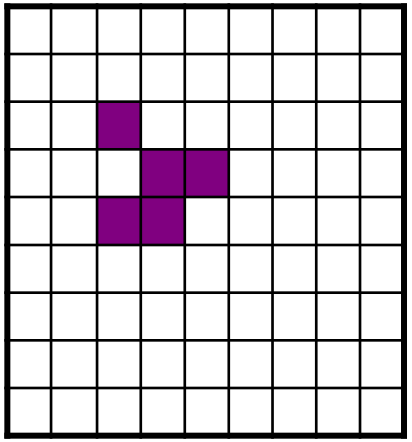
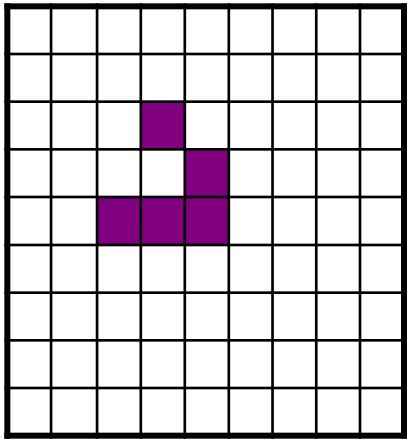






glisseur

Glisseur et perturbations

Comment l'expert résoud un problème ?

- La causalité ne suffit pas :
  - Une maladie cause un symptome
  - Mais les symptomes ne causent pas les maladies ..
- Comment fait l'expert ?
  - Il utilise un 'mode d'emploi' comme les débutants
  - Mais sait sélectionner très rapidement les bonnes infos
  - *"I don't know why this works but it does and so I'll use it again!"*
  - Difficile à élucider