

Agents logiques

1

Plan

- Agents à base de connaissances
- Le monde du wumpus
- La logique
- Ingénierie ontologique
- « Situation calculus »

2

Agents qui raisonnent logiquement

- **Agents logiques** : agents basés sur les **connaissances** disponibles concernant le monde et un **raisonnement** (logique) portant sur les actions possibles dans ce monde.
- Les agents logiques doivent connaître :
 - L'état actuel du monde.
 - Comment le monde change dans le temps ?
 - Qu'est ce qu'il faut accomplir ?
 - Quelles sont les conséquences des actions dans différentes circonstances ?

3

Agents à base de connaissances

- **Base de connaissances** : un ensemble de représentations de faits concernant le monde
 - chaque représentation est appelée un *énoncé*
 - une base de connaissances est un ensemble d'énoncés exprimés dans un *langage formel*
 - les énoncés sont exprimés dans un langage de représentation des connaissances.

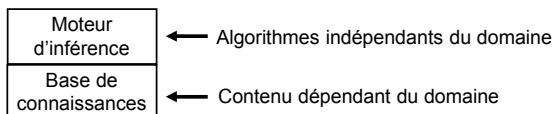
4

Gestion des connaissances

- L'ajout de connaissances est symbolisé par l'action *Tell* et l'interrogation (requête) est symbolisée par l'action *Ask*.
- La réponse à une requête (*Ask*) doit découler de ce qui a été ajouté (*Tell*) dans la base de connaissances.
- La base de connaissances ne peut pas inventer, elle doit déduire (inférer) à partir de ses mécanismes de déduction (moteur d'inférence).
- La base de connaissances peut contenir des informations initiales, i.e. des connaissances de base (background knowledge).

5

Structure



6

Capacités d'un agent

- Un agent à base de connaissances doit avoir les capacités suivantes :
 - représenter des états, des actions, etc.
 - incorporer de nouvelles perceptions
 - faire la mise à jour de sa représentation du monde
 - déduire des propriétés « cachées » du monde
 - déduire des actions appropriées

7

Description d'un agent

- Un agent à base de connaissances peut être décrit selon deux niveaux:
 - Niveau connaissances: description de l'agent par une description de ce qu'il sait.
 - Niveau implémentation: Description des structures de données de la base de connaissances et des algorithmes qui les manipulent.

8

Connaissances initiales

- Acquisition des connaissances initiales:
 - Approche déclarative: les connaissances initiales de l'agent sont ajoutées avec TELL, avant toutes perceptions.
 - Approche procédurales: les comportements désirés sont programmés directement.
- On peut aussi donner à l'agent la possibilité d'apprendre de nouvelles connaissances par lui-même pour devenir un agent autonome.

9

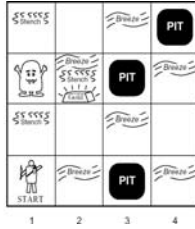
Le monde du *Wumpus* (PEAS)

- Mesure de performance

- +1000 pour l'or, -1000 pour tomber dans un trou, -1 pour chaque action, -10 pour tirer une flèche.

- Environnement

- Une grille de 4 x 4
- L'agent commence en [1,1] en regardant à droite
- Les locations de l'or, du wumpus et des trous sont choisis aléatoirement.

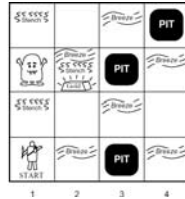


10

Le monde du *Wumpus* (PEAS)

- Effecteurs

- L'agent peut avancer, tourner à gauche ou tourner à droite.
- L'agent meurt s'il tombe dans un trou ou arrive sur la même case que le wumpus
- L'agent peut être sur la même case qu'un wumpus mort.
- Avancer n'a aucun effet s'il y a un mur
- L'action *Prendre* permet de ramasser un objet
- L'action *Tirer* permet de tirer une flèche en avant si l'agent en a une.

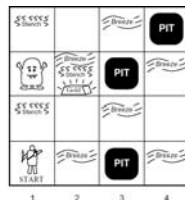


11

Le monde du *wumpus* (PEAS)

- Capteurs

- L'agent perçoit une pointe sur la case du wumpus et sur les cases adjacentes.
- L'agent perçoit une brise sur les cases adjacentes à un trou.
- L'agent perçoit scintillement s'il est sur la case de l'or.
- Si l'agent avance dans un mur, il va percevoir une collision.
- Lorsque le wumpus meurt, l'agent va percevoir un cri



12

Propriété de l'environnement

- **Observable ?**
 - Non, uniquement une perception locale
- **Déterministe ?**
 - Oui, les effets des actions sont spécifiés exactement
- **Épisodique ?**
 - Non, c'est séquentiel au niveau des actions
- **Statique ?**
 - Oui, le wumpus et les trous ne bougent pas
- **Discret ?**
 - Oui
- **Multiagent ?**
 - Non, le wumpus n'est qu'une composante de l'environnement

13

Action et raisonnement

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK			
1,1	2,1	3,1	4,1
A OK	OK		

- A** = Agent
- B = Breeze
- G = Glitter, Gold
- OK = Safe square
- P = Pit
- S = Stench
- V = Visited
- W = Wumpus

14

Action et raisonnement

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK	P?		
1,1	2,1	3,1	4,1
V OK	A B OK	P?	

- A** = Agent
- B = Breeze
- G = Glitter, Gold
- OK = Safe square
- P = Pit
- S = Stench
- V = Visited
- W = Wumpus

15

Action et raisonnement

1,4	2,4	3,4	4,4
1,3 W?	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P?	4,1

A = Agent
 B = Breeze
 G = Glitter, Gold
 OK = Safe square
 P = Pit
 S = Stench
 V = Visited
 W = Wumpus

16

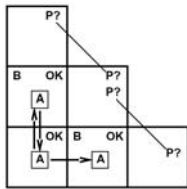
Action et raisonnement

1,4	2,4 P?	3,4	4,4
1,3 W?	2,3 A S G B	3,3 P?	4,3
1,2 S V OK	2,2 V OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P?	4,1

A = Agent
 B = Breeze
 G = Glitter, Gold
 OK = Safe square
 P = Pit
 S = Stench
 V = Visited
 W = Wumpus

17

Décisions difficiles



- Brise en (1,2) et (2,1)
 - Aucun déplacement sécuritaire
 - L'agent peut tenter sa chance
- Puanteur en (1,1)
 - Aucun déplacement sécuritaire
 - Stratégie: lancer une flèche
 - Si le wumpus était là, il est mort, donc c'est sécuritaire.
 - Si le wumpus n'était pas là, c'est sécuritaire.

18

La logique

- Les **logiques** sont des langages formels pour représenter de l'information de manière à permettre d'en déduire des conclusions.
- **Syntaxe**: Définit les configurations possibles pouvant constituer des phrases.
- **Sémantique**: Définit le sens d'une phrase, c'est-à-dire, définit la véracité d'une phrase.

19

La logique

- Exemple: le langage de l'arithmétique
 - « $x + 2 > y$ » est un énoncé, mais « $x2 + y >$ » n'est pas un énoncé.
 - « $x + 2 > y$ » est vrai si le nombre $x + 2$ est plus grand que le nombre y .
 - « $x + 2 > y$ » est vrai dans un monde où $x = 7$ et $y = 1$.
 - « $x + 2 > y$ » est faux dans un monde où $x = 0$ et $y = 6$.

20

Inférence

- La base de connaissance (KB) infère α si α est vrai dans tous les mondes où KB est vrai. On note cela

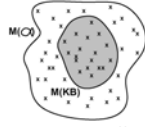
$$KB \models \alpha$$

- Exemple, $x + y = 4$ permet d'inférer que $4 = x + y$

21

Modèles

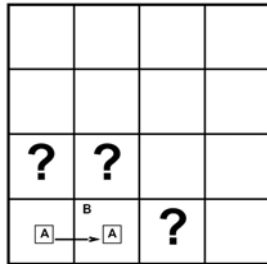
- m est un modèle d'un énoncé α si α est vrai dans m .
- $M(\alpha)$ est l'ensemble de tous les modèles de α .
 $KB \models \alpha$ ssi $M(KB) \subseteq M(\alpha)$
- Exemple,
 - KB = Canadiens ont gagnés et Sénateurs ont gagnés
 - α = Canadiens ont gagnés



22

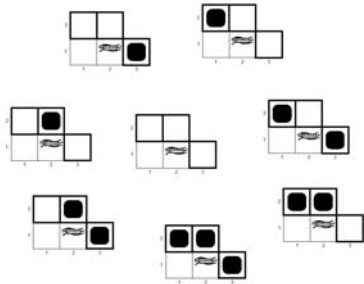
Exemple

- Considérons les modèles possibles pour les trois « ? ».
- On ne s'attarde qu'au trous, donc il y a 2^3 modèles possibles.



23

Modèles possibles



24

Modèles possibles

KB = Les règles du monde + les observations

25

Modèles possibles

KB = Les règles du monde + les observations

$\alpha_1 = \text{"(1,2) est OK"}$, $KB \models \alpha_1$
prouvé par vérification de modèle

26

Modèles possibles

KB = Les règles du monde + les observations

$\alpha_2 = \text{"(2,2) est OK"}$, $KB \not\models \alpha_2$

27

Inférence

- Si un algorithme d'inférence i permet d'inférer α à partir de KB , on écrit:

$$KB \vdash_i \alpha$$

- Un algorithme d'inférence **préserve la véracité** (sound) si:

$$KB \vdash_i \alpha \Rightarrow KB \models \alpha$$

- Un algorithme d'inférence est **complet** si:

$$KB \models \alpha \Rightarrow KB \vdash_i \alpha$$

28

Remarques

- Il faut distinguer entre un fait et un énoncé:
 - Un fait est une partie intégrante du monde.
 - Un énoncé est une représentation encodée d'un fait qui peut être emmagasiné et manipulé par l'agent.
 - Les mécanismes de raisonnement opèrent sur les représentations des faits (c-à-d les énoncés) et non pas sur les faits eux-mêmes.

29

Remarques

- La procédure d'inférence:
 - génère de nouveaux énoncés à partir de la base de connaissances
 - ou elle vérifie si un énoncé peut-être dérivé à partir de la base de connaissances.
- Ce type de système préserve la vérité (sound) : à partir d'énoncés vrais dans la base de connaissances, d'autres énoncés vraies sont générés par preuve. $KB \vdash_i \alpha \Rightarrow KB \models \alpha$

30

Remarques

- Le système ne peut pas inférer des énoncés qui contredisent la base de connaissances.

31

Interagir avec une BC en LPO

- Supposons un agent dans le monde du wumpus utilisant une BC en LPO.
 - L'agent perçoit une puanteur, une brise, mais pas de scintillement au temps 5.

$Tell(BC, Percept([Puanteur, Brise, Rien], 5))$

$Ask(BC, \exists a Action(a, 5))$

– Réponse: Oui, $\{a/Tire\}$ (Substitution)

- Exemple de substitution : $S = Soeur(x, y)$
 $\alpha = \{x/Marie, y/Jean\}$
 $S\alpha = Soeur(Marie, Jean)$

32

BC pour le monde du wumpus

- **Perceptions**
 $\forall b, s, t Percept([Puanteur, b, s], t) \Rightarrow Puanteur(t)$
 $\forall p, b, t Percept([p, b, Scintillement], t) \Rightarrow Scintillement(t)$
- **Réflexe**
 $\forall t scintillement(t) \Rightarrow MeilleureAction(Prendre, t)$
- **Réflexe avec état interne**
 $\forall t Scintillement(t) \wedge \neg Possède(Or, t) \Rightarrow Action(Prendre, t)$
- On ne peut pas observer $Possède(Or, t)$, donc c'est important de tenir à jour les changements de l'environnement.

33

Définir l'environnement

- Propriétés des emplacements
 $\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Puanteur}(t) \Rightarrow \text{Puc}(x)$
 $\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Venteux}(t) \Rightarrow \text{Brise}(x)$
- Il y a une brise sur les cases adjacentes à un trou
 - Règle de diagnostic : inférer la cause à partir de l'effet
 $\forall y \text{ Brise}(y) \Rightarrow \exists x \text{ Trou}(x) \wedge \text{Adjacent}(x, y)$
 - Règle causale: inférer l'effet par la cause
 $\forall x, y \text{ Trou}(x) \wedge \text{Adjacent}(x, y) \Rightarrow \text{Brise}(y)$
- Ces définitions ne sont pas complètes, la bonne définition est :
 $\forall y \text{ Brise}(y) \Leftrightarrow [\exists x \text{ Trou}(x) \wedge \text{Adjacent}(x, y)]$

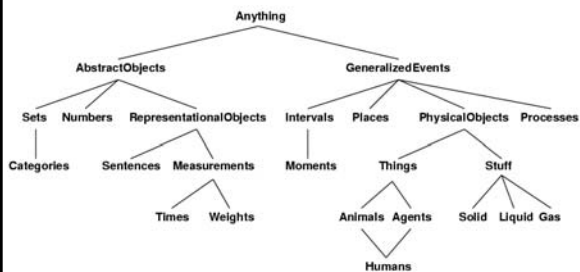
34

Ingénierie ontologique

- Pour des problèmes jouets, la représentation n'est pas vraiment importante.
- Les domaines complexes demandent des représentations plus complexes, plus générales et plus flexibles.
- Comment représenter des concepts comme :
 - Actions, Temps, Objets physiques, Croyance.
- La représentation de ces concepts abstraits est appelé **l'ingénierie ontologique**.

35

« Upper ontology »



36

Catégories

- L'interaction avec le monde est au niveau des **objets**, mais le raisonnement a souvent lieu au niveau des **catégories**.
- En LPO, on peut utiliser les **prédicats** ou les **objets** pour représenter les catégories.
 - Ex: $\text{BallonDeBasketball}(b)$ ou $\text{BallonDeBasketballs}$
 - $\text{Membre}(b, \text{BallonDeBasketballs})$ ou $b \in \text{BallonDeBasketballs}$
 - $\text{SousEnsemble}(\text{BallonDeBasketballs}, \text{Ballons})$ ou $\text{BallonDeBasketballs} \subset \text{Ballons}$

37

Prédicats

- Phrases catégoriques Considérons tout d'abord des phrases qui ont clairement une structure sujet/prédicat (celles que la tradition philosophique appelle catégoriques):
 - a. Platon est un homme
 - b. Socrate est mortel
 - c. Le train siffle
 - d. Cette bouilloire fuit
- Dans chaque phrase on peut identifier : une partie "propriété" (prédicat), et une partie "entité" (individu). Vont leur correspondre en logique de prédicats : des noms de prédicats (constantes de prédicat), et des constantes individuelles, une phrase étant considérée comme l'application (fonctionnelle) d'un prédicat à un individu. Par exemple, avec une "légende" : $H = \text{être un homme}$, $p = \text{Platon}$, etc :
 - a. $H(p)$
 - b. $M(s)$
 - c. $S(t)$
 - d. $F(b)$

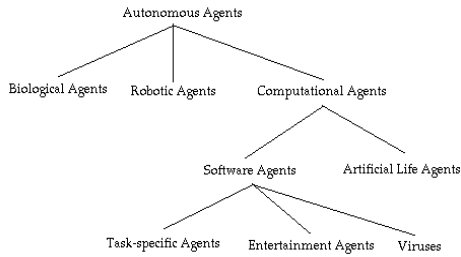
38

Catégories

- **Héritage**: permet de simplifier la base de connaissances.
 - Catégorie *Nourriture* : Toutes les instances de *Nourriture* sont comestibles.
 - *Fruit* est une sous-classe de *Nourriture*
 - *Pomme* est une sous-classe de *Fruit*
 - Alors, on sait que toutes les pommes sont comestibles.
- Les relations de sous-classe organise les catégories en **taxonomie**.

39

Exemple de taxonomie



40

LPO et catégories

- Un objet est membre d'une catégorie:
 $BB_9 \in Basketballs$
- Une catégorie est une sous-classe d'une autre:
 $Basketballs \subset Balls$
- Tous les membres d'une catégorie ont certaines propriétés:
 $x \in Basketballs \Rightarrow Round(x)$
- Les membres d'une catégorie peuvent être reconnus en identifiant certaines propriétés:
 $Orange(x) \wedge Round(x) \wedge Diameter(x) = 9.5'' \wedge x \in Balls \Rightarrow x \in Basketballs$

41

Caractéristiques des catégories

- **Disjointes** : Deux catégories sont disjointes si elles n'ont pas de membres en commun.
– Ex: animaux et végétaux
- **Décomposition exhaustive** : Décomposition d'une catégorie en toutes les sous-classes possibles.
– Ex: Américains, Canadiens et Mexicains est une décomposition exhaustive de la catégorie Nord-américains.
- **Partition** : décomposition exhaustive (approfondie) et disjointe.
– Mâle et Femelle est une partition de la catégorie Animales.

42

Composition physique

- Un objet fait parti d'un autre.
 - On utilise la relation *PartOf*
 - Ex: *PartOf*(Québec, Canada)
 - *PartOf* est réflexive et transitive
- Une composition d'objets avec des parties définies, mais sans structure
 - On utilise la relation *BunchOf* (*ensemble serait abstrait*)
 - *BunchOf*(*{Pomme₁, Pomme₂, Pomme₃}*)
 - Important de distinguer *BunchOf*(*Pommes*) et *Pommes*

43

Mesures

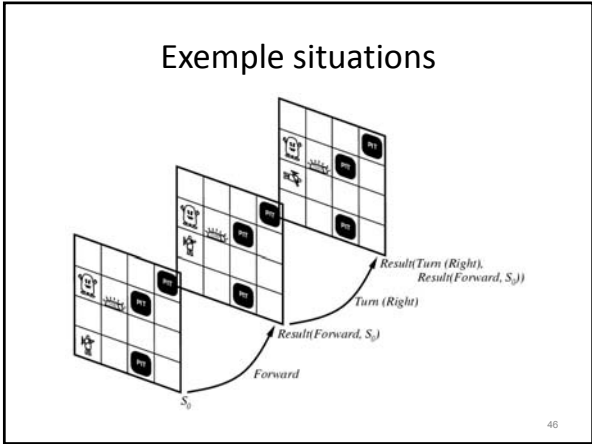
- Comment représenter des mesures ?
- Une mesure peut avoir différentes unités, donc il faut représenter l'unité.
 - Ex: *Longueur*(L) = *Pouces*(1.5) = *Centimètres*(3.81)
- On peut exprimer les conversions:
 - *Centimètres*(2.54 x d) = *Pouces*(d)
- Pour les grandeurs qualitatives, on va plutôt ordonner les symboles.
 - Ex: *Difficulté*(*exercice₁*) > *Difficulté*(*Exercice₂*)

44

« Situation calculus »

- **Actions:** terme logique
 - Ex: *Avancer*, *Turner*(*Droite*)
- **Situations:** la situation initiale (S_0) ou toutes les situations résultant de l'application d'une action.
 - La fonction *Result*(*a,s*) est la situation résultant en appliquant l'action *a* dans la situation *s*.
 - Toutes les situations (sauf S_0) sont le résultat d'une action.

45



« Situation calculus »

- **Fluents**: fonctions ou prédicats qui varient d'une situation à l'autre.
– Ex: $\neg Tient(G_1, S_0)$
- **Éternel ou intemporel**: Qui ne change pas d'une situation à l'autre.
– Ex: $Or(G_1)$

47

Séquence d'actions

- On peut représenter des séquences d'actions
- Exécuter une séquence vide ne change pas la situation.
– $Result([], s) = s$
- Exécuter une séquence non vide est la même chose que d'exécuter la première action et d'ensuite exécuter le reste de la séquence dans la situation résultante.
– $Result([a|seq], s) = Result(seq, Result(a, s))$

48

Exemple wumpus

- Monde simple, agent en (1,1), or en (1,2)
- Situation initiale: $At(Agent, [1, 1], S_0) \wedge At(G_1, [1, 2], S_0)$
- Ce n'est pas suffisant, il faut aussi dire ce qui n'était pas vrai. $At(o, x, S_0) \Leftrightarrow [(o = Agent \wedge x = [1, 1]) \vee (o = G_1 \wedge x = [1, 2])]$.
 $\neg Holding(o, S_0)$.
- On doit aussi dire que G1 est de l'or et que les deux cases sont adjacentes.
 $Gold(G_1) \wedge Adjacent([1, 1], [1, 2]) \wedge Adjacent([1, 2], [1, 1])$.
- Pour trouver un plan, on pose la question:
 $\exists seq \ At(G_1, [1, 1], Result(seq, S_0))$

49

Décrire des actions

- On utilise les axiomes:
 - **Possibilité:** dit quand c'est possible d'exécuter l'action.
 - **Effets:** dit ce qui se passe si l'action est exécutée.

AXIOME POSSIBILITÉ: $Préconditions \Rightarrow Poss(a, s)$.
 AXIOME EFFET: $Poss(a, s) \Rightarrow Changements$
 qui résultent de l'exécution de l'action.

50

Exemple

- Axiomes possibilités:
 - $At(Agent, x, s) \wedge Adjacent(x, y) \Rightarrow Poss(Go(x, y), s)$.
 - $Gold(g) \wedge At(Agent, x, s) \wedge At(g, x, s) \Rightarrow Poss(Grab(g), s)$.
 - $Holding(g, s) \Rightarrow Poss(Release(g), s)$.
- Axiomes effets:
 - $Poss(Go(x, y), s) \Rightarrow At(Agent, y, Result(Go(x, y), s))$.
 - $Poss(Grab(g), s) \Rightarrow Holding(g, Result(Grab(g), s))$.
 - $Poss(Release(g), s) \Rightarrow \neg Holding(g, Result(Release(g), s))$.
- Supposons que l'agent va en (1,2)
 $At(Agent, [1, 2], Result(Go([1, 1], [1, 2]), S_0))$
- Par la suite, il faudrait prendre l'or, mais on n'a pas
 $At(G_1, [1, 2], Result(Go([1, 1], [1, 2]), S_0))$

51

Difficulté

- Le problème, c'est que l'axiome effet représente ce qui change, mais ne dit pas ce qui ne change pas.
 - C'est le **Frame Problem** ou **problème de persistance de ce qui reste identique et ne change pas**
- Il faut trouver un moyen de représenter ce qui ne change pas.

52

« Frame problem »

- Comment gérer ce qui ne change pas ?
 - **Représentation**: On ne peut pas lister pour chacune des actions ce qui ne change pas.
 - **Inférence**: On veut éviter de devoir copier à répétition pour connaître l'état.

53

Axiome Successor-State

- Règle le problème de représentation du « frame problem »

AXIOME SUCCESSOR-STATE:

Action est possible \Rightarrow

(Fluent est vrai dans l'état résultant \Leftrightarrow L'effet de l'action l'a rendu vrai
 \vee Il était vrai et l'action n'a rien changé).

- Exemple: L'axiome pour le fluent **Holding**

$Poss(a, s) \Rightarrow$

$(Holding(g, Result(a, s)) \Leftrightarrow a = Grab(g) \vee (Holding(g, s) \wedge a \neq Release(g)))$.

54

Exemple

- Écrire les axiomes « Successor-State » nécessaire pour définir les effets de l'action *Tirer* dans le monde du wumpus. Décrire ces effets sur le wumpus, en se rappelant que *Tirer* utilise la seule flèche de l'agent.

$$\begin{aligned} \forall a, s \text{ Has}(\text{Agent}, \text{Arrow}, \text{Result}(a, s)) &\Leftrightarrow \\ &[\text{Has}(\text{Agent}, \text{Arrow}, s) \wedge (a \neq \text{Shoot})] \\ \forall a, s \text{ Alive}(\text{Wumpus}, \text{Result}(a, s)) &\Leftrightarrow \\ &[\text{Alive}(\text{Wumpus}, s) \\ &\wedge \neg(a = \text{Shoot} \wedge \text{Has}(\text{Agent}, \text{Arrow}, s) \\ &\wedge \text{Facing}(\text{Agent}, \text{Wumpus}, s))] \end{aligned}$$

55
