

# Intelligence Artificielle

Boi Faltings

Laboratoire d'Intelligence Artificielle

`boi.faltings@epfl.ch`

`http://moodle.epfl.ch/`

# Organisation

- Informations sur Moodle: <http://moodle.epfl.ch/>  
Mot de passe: ai1515
- Cours: dans la salle CM 13. Les cours de 2021 sont enregistrés et disponible sur Moodle avec le même contenu.
- Exercices: dans la salle IN F1.

# Qu'est-ce que l'intelligence

- aptitude au calcul rapide?



# Qu'est-ce que l'intelligence

- aptitude au calcul rapide?
- reconnaissance de formes ou de situations?



# Qu'est-ce que l'intelligence

- aptitude au calcul rapide?
- reconnaissance de formes ou de situations?
- aptitude à l'apprentissage?



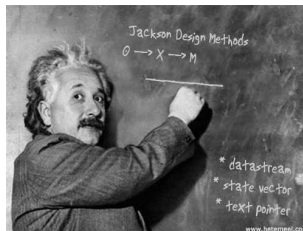
# Qu'est-ce que l'intelligence

- aptitude au calcul rapide?
- reconnaissance de formes ou de situations?
- aptitude à l'apprentissage?
- aptitude à réagir à l'environnement?



# Qu'est-ce que l'intelligence

- aptitude au calcul rapide?
- reconnaissance de formes ou de situations?
- aptitude à l'apprentissage?
- aptitude à réagir à l'environnement?
- capacité d'abstraction, de raisonnement et d'explication?



# Qu'est-ce que l'intelligence

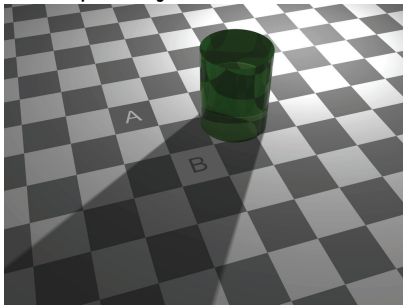
- aptitude au calcul rapide?
- reconnaissance de formes ou de situations?
- aptitude à l'apprentissage?
- aptitude à réagir à l'environnement?
- capacité d'abstraction, de raisonnement et d'explication?

Aucune définition fait l'unanimité!



# L'intelligence: un concept logique?

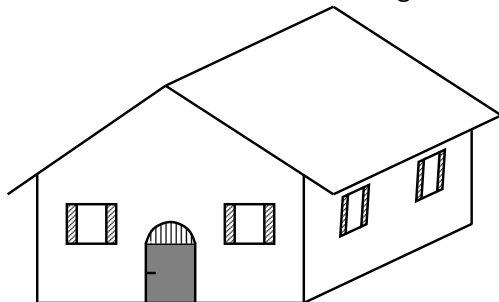
Notre perception n'est pas objective:



- "A" et "B" ont exactement le même niveau de gris...
- ...mais notre cerveau reconnaît l'ombre et "corrige" l'image.

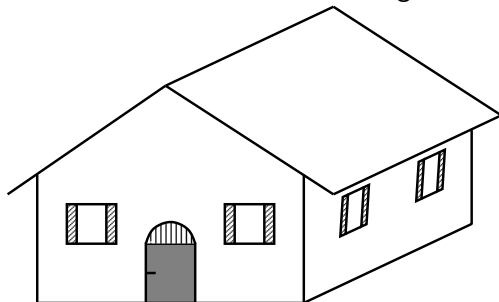
# L'intelligence: un concept logique?

Qu'est-ce qu'on attend d'un ordinateur intelligent?



# L'intelligence: un concept logique?

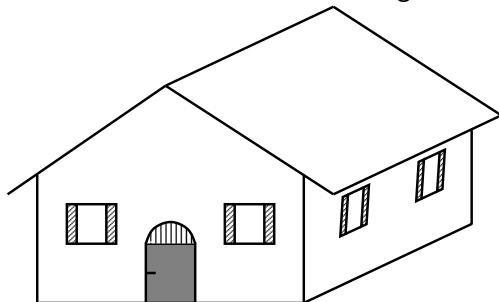
Qu'est-ce qu'on attend d'un ordinateur intelligent?



Réponse logique: "Je vois un ensemble de traits sur un écran..."

# L'intelligence: un concept logique?

Qu'est-ce qu'on attend d'un ordinateur intelligent?



Réponse logique: "Je vois un ensemble de traits sur un écran..."

Réponse intelligente: "Je vois une maison...."

⇒ la réponse intelligente est logiquement fausse!!

# L'intelligence: un concept logique?

Qu'est-ce qu'on attend d'un ordinateur intelligent?

"Pourriez-vous me passer le sel?"

- réponse logique: "oui" ou "non"
- réponse intelligente: action de passer le sel.

# Intelligence et connaissances

L'intelligence ne se définit pas par un processus, mais par les *connaissances* qu'elle implique:

- opérations admissibles
- lois de la nature
- lois, règles et normes sociaux
- ...

⇒ l'ordinateur intelligent a besoin d'une grande quantité de connaissances:

*systèmes* basés sur les connaissances

# WATSON: un succès grâce aux connaissances



- connaissances linguistiques: formuler les questions
- connaissances du monde, obtenus par "lecture" du web
- connaissances pratiques: quelle genre de question est probable?

# Connaissances et données

Les données sont *structurées*:

- nombres, noms, addresses, etc.
- structurées en tableaux et bases de données

Les connaissances sont souvent *non-structurées*:

- textuelles, règles, heuristiques.
- expressions logiques, contraintes, etc.
- parfois probabiliste.

⇒ les techniques sont différentes



# Machine Learning

Abondance de données  $\Rightarrow$

- connaissances peuvent être *appries* au lieu de codées.
- $\Rightarrow$  construction et adaptation de systèmes simplifiée.
- raison pour la grande popularité de l'Intelligence Artificielle.
  - implémentation souvent par circuits spéciaux: réseaux de neurones artificiels.
  - modèles d'une énorme complexité: GPT 4 a plus de 1000 milliards de paramètres.

# Nouveaux paradigmes

Classique: spécification  $\Rightarrow$  programmation

Nouveau:

- raisonnement/optimisation  $\Rightarrow$  comportement émergent.
- incertitude  $\Rightarrow$  calcul probabiliste.
- exemples de bons comportements  $\Rightarrow$  apprentissage.

# Nouveaux paradigmes

Classique: spécification  $\Rightarrow$  programmation

Nouveau:

- raisonnement/optimisation  $\Rightarrow$  comportement émergent.
- incertitude  $\Rightarrow$  calcul probabiliste.
- exemples de bons comportements  $\Rightarrow$  apprentissage.

implique:

- difficile de prévoir le comportement.
- difficile de garantir que les résultats soient corrects.
- difficile de savoir si le bon concept a été appris.

# Les soucis...

- Hawking: L'intelligence artificielle est la fin de l'humanité
- Musk: Plus dangereux que les armes nucléaires.
- Gates: L'humanité devrait se faire du souci.

Comportement moins prévisible  $\Rightarrow$  difficile à maîtriser.

Pendant, nous sommes loin d'une situation dangereuse....

# Structure d'un logiciel IA

## Résolution de problèmes

*comment obtenir pos(moi,p5)?*

$\Rightarrow$  Plan: vérifier(p1,vert)

avancer(moi)

.....

## Apprentissage

$pos(p7,p6,t1), pos(p7,p5,t2)$

$pos(p10,p6,t3), pos(p10,p5,t4)$

$pos(x,p6,y) \rightarrow pos(x,p5,z)$

## Raisonnement

$Pr(traverse(p2))=0.45$

$Pr(change(p8,p5)=0.01$

$Pr(change(p7,p5))=0.2$

traverse(p1,p3)

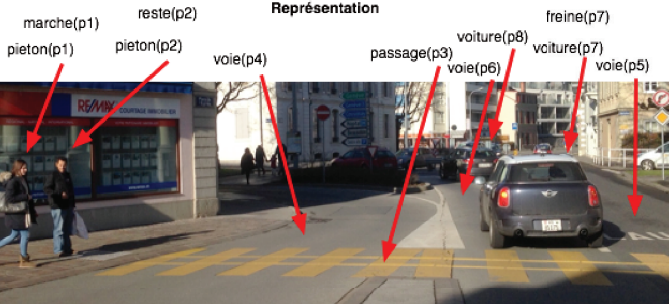
tourne-à-  
gauche(p6)

pos(p8,p6)

voie-opposée(p4)

pos(p7,p6)

## Représentation



# L'IA generative

Jusqu'en 2020, les résultats d'un programme IA étaient des réponses simples:

- choix d'action pour un robot, voiture autonome ou joueur de GO.
- classification: est-ce que X va payer ses factures?
- diagnostic: quel est la maladie du patient?

Depuis sont apparus des techniques pour la génération d'objets complexes, surtout textes et images.

Ils ouvrent des domaines d'application beaucoup plus larges.

# Principes de l'IA generative

- modèle fondamental: modélise des connaissances séquentielles.
  - exemple: modèle de langage prédit le prochain mot sur la base des mots précédents.
  - fournir au modèle un contexte initial (*prompt*) et iterativement produire le text mot par mot.
  - entrainer le modèle sur l'ensemble des textes disponibles, y compris des articles scientifiques, brevets, journaux, dialogues de films, etc.
- ⇒ le modèle contient des connaissances sur le monde.

# Les outils de Open AI

Plusieurs modèles fondamentaux:

- langage: context  $\Rightarrow$  prochain mot.  
Generative Pretrained Transformers (GPT).
- images: annotations  $\Rightarrow$  image ou fragments d'images.  
Dall-E.

Modèles purement statistiques augmentés par un entraînement spécifique par des humains:

- réponses à des questions.
- instructions sur la manière de générer des réponses (style, verbose/court, avec/sans explications, etc.)
- interdiction de révéler certaines informations (comment produire des bombes, insultes, etc)
- correction de réponses erronées
- etc.



# Limitations de la technologie

- consommation d'énergie énorme.
- difficile voire impossible à contrôler les réponses pour un contexte spécifique.
- hallucinations
- incapable de résoudre des problèmes dont la solution ne se trouve pas dans les données d'entraînement (planification, diagnostic, etc.).

Ils ne remplacent pas l'IA "classique".

# Contenu du Cours

**2**

## Résolution de problèmes

*comment obtenir  $\text{pos}(\text{moi}, p5)$ ?*

$\Rightarrow$  Plan: vérifier( $p1, \text{vert}$ )  
avancer( $\text{moi}$ )

.....

**3**

## Apprentissage

$\text{pos}(p7, p6, t1), \text{pos}(p7, p5, t2)$   
 $\text{pos}(p10, p6, t3), \text{pos}(p10, p5, t4)$   
-----  
 $\text{pos}(x, p6, y) \rightarrow \text{pos}(x, p5, z)$

**1**

$\text{Pr}(\text{traverse}(p2))=0.45$

$\text{Pr}(\text{change}(p8, p5))=0.01$

$\text{Pr}(\text{change}(p7, p5))=0.2$

$\text{traverse}(p1, p3)$

tourne-à-

$\text{pos}(p8, p6)$

voie-opposée( $p4$ )

gauche( $p6$ )

$\text{pos}(p7, p6)$

## Représentation

marche( $p1$ )

reste( $p2$ )

freine( $p7$ )

pieton( $p1$ )

pieton( $p2$ )

voie( $p4$ )

passage( $p3$ )

voiture( $p8$ )

voiture( $p7$ )

voie( $p6$ )

voie( $p5$ )



## Contenu du cours (2)

Introduction aux systèmes basés sur la connaissance et à leurs applications:

- ①
  - Représentation de connaissances et d'inférences
  - Moteurs d'inférence
  - Raisonnement incertain
- ②
  - Recherche et satisfaction de contraintes
  - Diagnostic
  - Planification
- ③
  - Induction paramétrique et non-paramétrique
  - Clustering
  - Réseaux de neurones artificiels

# Domaines spécifiques

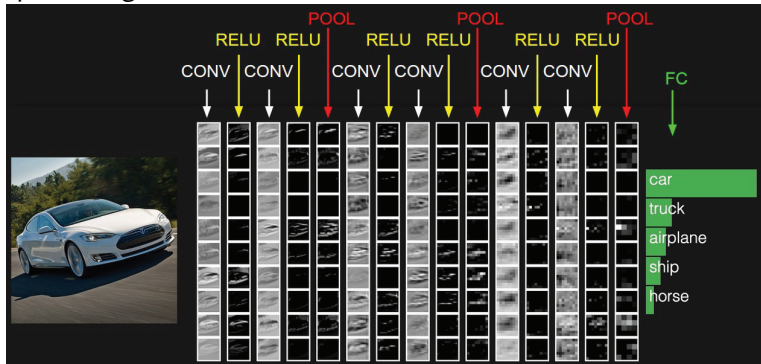
Certains domaines ont mis au point des techniques propres à leur problématique:

- Robotique: importance de la mécanique, des capteurs et du réglage automatique.
- Vision: traitement de signaux, grandes quantités de données.
- Interpretation du langage: analyse syntaxique, méthodes statistiques.

⇒ font l'objet de cours spécifiques.

# Reconnaissance de formes

La reconnaissance de formes a fait des grands progrès grâce au *deep learning*:



Meilleures performances que les humains en reconnaissance de visages, parole, etc.

# Types d'applications

- Simulation d'un raisonnement
- Résolution de problèmes complexes
- Apprentissage de nouvelles connaissances

# Simulation d'un raisonnement

- Gestion de connaissances (knowledge management): formalisation de procédures et du savoir-faire
- Semantic web: formalisation du contenu de pages web pour permettre la recherche et synthèse automatique
- Systèmes experts: codification des connaissances d'un expert

# Prédiction de toxicité: DEREK

- Tâche: est-ce qu'une structure chimique sera toxique pour des humains?
- DEREK: Système à base de connaissance
- Mise à jour continuellement par des experts
- Utilisé par la plupart des entreprises chimiques et pharmaceutiques

<http://www.lhasalimited.org/products/derek-nexus.htm>



# Résolution de problèmes

Aide à la décision:

- ordonnancement d'usines
- configuration automatique
- diagnostic de radiographies

Systèmes autonomes:

- robots mobiles
- opérations militaires
- diagnostic d'automobiles, vaisseaux spatiaux, etc.

# Exemple: Aurora

- Problème: gestion de processus complexes, en particulier réaction à des situations imprévues.
- Exemple: production d'avions (Boeing 787)
- Aurora utilise des techniques de recherche et planification IA pour ce problème.

<https://www.stottlerhenke.com/products/aurora/>

# Exemple: Diagnostic autonome

- Diagnostic basé sur modèles:
- Structure du système + raisonnement  $\Rightarrow$  diagnostic
- Nombreuses applications dans l'espace, l'automobile, etc.

# Apprentissage de nouvelles connaissances

## Découverte des connaissances (Data Mining):

- détection d'associations: délit d'initié, blanchissage d'argent
- classification: détection de mauvais payeurs potentiels
- apprentissage automatique: recommandation, marketing

# Detection d'associations

Système d'identification de fraudes (IBM):

- analyse de textes  $\Rightarrow$  reconnaissance d'événements.
- apprentissage automatique de caractéristiques d'événements et de leur relations.
- analyse de relations entre événements  $\Rightarrow$  fraudes.

<https://ojs.aaai.org/index.php/AAAI/article/view/19045>

# Marketing Intelligent

- But: montrer la publicité à ceux qui veulent la voir.
- ⇒ apprendre les caractéristiques de personnes (ou leurs ordinateurs) et de la publicité, et la relation entre les deux.
- montrer la publicité qui correspond le mieux.
- toute une industrie: Google, Yahoo, Criteo, etc.

<https://www.criteo.com/technology/ai-engine/>

# Connaissances

Connaissances = données non-structurées

Souvent sous forme textuelle

Défis pour un traitement informatique:

- modélisation
- traduction en langue formelle
- inférence

# Modélisation de connaissances

Faiblesse des connaissances non-structurées: pas clair comment on peut les appliquer!

But de la modélisation:

- définir quels sont les conclusions qui peuvent être tirées des connaissances.
- définir des méthodes pour arriver à ces conclusions



## Exemple: règlement $\Rightarrow$ programme

Extrait du règlement de Lointainie:

*Tout voyageur adulte a le droit d'importer hors-taxe une petite quantité de marchandise.*

*Sont considérée comme petite quantité:*

- *pour le cognac, moins de 1 litre.*
- *pour le vin, moins de 2 litres.*
- *pour toute marchandise, une valeur de moins de 100 Francs.*

A construire: programme qui décide s'il faut taxer un voyageur.

# Demandons à ChatGPT

Tout voyageur adulte a le droit d'importer hors-taxe une petite quantité de marchandise.

Sont considérée comme petite quantité:

- pour le cognac, moins de 1 litre.
- pour le vin, moins de 2 litres.
- pour toute marchandise, une valeur de moins de 100 Francs.

Le voyageur Charles est adulte et porte avec lui 1.5 litres de vin. Est-ce qu'il peut l'importer hors taxe?

ChatGPT:

Non, Charles ne peut pas importer les 1.5 litres de vin hors taxe selon les critères spécifiés. Selon les règles données, un voyageur adulte peut importer hors taxe jusqu'à 2 litres de vin. Dans le cas de Charles, **il dépasse cette limite en portant 1.5 litres de vin**. Par conséquent, il ne peut pas bénéficier de l'importation hors taxe pour le vin.

# Modélisation

Identifier les objets:

*voyageur (v), marchandise (m)*

Identifier les propriétés:

*adulte(v), hors-taxe(m,v), petite-quantite(m), cognac(m),  
<-1-litre(m), <-2-litres(m), <-100-Frs(m)*

Identifier les règles:

*Si adulte(v) et petite-quantite(m) alors hors-taxe(m,v)*

*Si cognac(m) et <-1-litre(m) alors petite-quantite(m)*

*Si vin(m) et <-2-litres(m) alors petite-quantite(m)*

*Si <-100-Frs.(m) alors petite-quantite(m)*

# Modèle $\Rightarrow$ Programme

Traduction en algorithme:

```
Read(adult,type,quantite,valeur)
if (type = cognac) {
  if ((quantite < 1) and adult) { print("hors-taxe") }
  else { print("tax") }}
else {
  if (type=vin) {
    if ((quantite < 2) and adult) { print ("hors-taxe") }
    else { print("tax") }}
  else {
    if ((valeur < 100Frs) and adult) { print("hors-taxe") }
    else { print("tax") }}}}
```

# Faiblesses

- Important de documenter:
  - le programme attend comme entrées ...
  - rend comme résultat ...
- Difficile à mettre à jour!
- Aucune explication du résultat!
- Pas de capacité de trouver comment éviter les taxes

# Désiderata

- Préserver la sémantique du problème dans le programme.
- Maintenir une correspondance entre programme et spécifications pour faciliter la maintenance et pouvoir donner des explications.
- Préserver la logique des règles au lieu de les traduire dans une prescription de calcul.
- Solution: représentation en *calcul des prédicats*

# La logique des prédicats

La logique des prédicats permet la représentation complète de modèles:

- Symboles = objets et paramètres
- Prédicats = propriétés et relations

Exemples:

- Symboles: Jacques, vitesse, 3.523
- Prédicats: `grand(Jacques)`, `valeur(vitesse,3.523)`

# La formulation de propositions

Expressions logiques = *propositions*

Propositions simples = prédicats sans variables libres:

il-fait-beau

jeune(Jacques)

valeur(vitesse, 3.543)

Propositions composées = expressions formées par des prédicats et des connecteurs:

jeune(Jacques)  $\wedge$  vieux(Charles)

vient(Marie)  $\vee$  vient(Chantal)

( $\wedge$  = "et",  $\vee$  = "ou")

Une proposition peut être vraie ou fausse. On peut bien représenter des propositions fausses: par exemple, `consiste-en(lune,fromage)`.



# Les connecteurs logiques

a **et** b:  $a \wedge b$

a **ou** b:  $a \vee b$

**negation**(non) a:  $\neg a$

a **implique** b:  $a \Rightarrow b$  (équivalent à  $\neg a \vee b$ )

Précédence habituelle:  $\wedge$  est plus fort que  $\vee$ , mais les parenthèses peuvent grouper des expressions

Exemples:

$(\text{étudie}(\text{Pierre}) \wedge \text{réussit}(\text{Pierre})) \vee \neg \text{réussit}(\text{Pierre})$

$\text{étudie}(\text{Pierre}) \Rightarrow \text{réussit}(\text{Pierre})$

$\neg \text{étudie}(\text{Pierre}) \vee \text{réussit}(\text{Pierre})$

# Règles d'équivalence logique

- $\neg(\neg X_1) \Leftrightarrow X_1$
- Les lois de Morgan:  
 $\neg(X_1 \wedge X_2) \Leftrightarrow \neg X_1 \vee \neg X_2$   
 $\neg(X_1 \vee X_2) \Leftrightarrow \neg X_1 \wedge \neg X_2$
- Les lois distributives:  
 $X_1 \wedge (X_2 \vee X_3) \Leftrightarrow (X_1 \wedge X_2) \vee (X_1 \wedge X_3)$   
 $X_1 \vee (X_2 \wedge X_3) \Leftrightarrow (X_1 \vee X_2) \wedge (X_1 \vee X_3)$
- Les lois commutatives:  
 $X_1 \wedge X_2 \Leftrightarrow X_2 \wedge X_1$   
 $X_1 \vee X_2 \Leftrightarrow X_2 \vee X_1$
- les lois associatives:  
 $(X_1 \wedge X_2) \wedge X_3 \Leftrightarrow X_1 \wedge (X_2 \wedge X_3)$   
 $(X_1 \vee X_2) \vee X_3 \Leftrightarrow X_1 \vee (X_2 \vee X_3)$
- la loi de la contraposée:  
 $X_1 \Rightarrow X_2 \Leftrightarrow \neg X_2 \Rightarrow \neg X_1$

# Inférence

- Le calcul de prédicats définit une représentation qui sert de base pour l'*inférence* logique.
- Une inférence suit le schéma suivant: à partir d'un ensemble de propositions  $\{\mathcal{P}\}$ , on trouve une nouvelle proposition  $x$  tel que si toutes les propositions de  $\{\mathcal{P}\}$  sont vraies,  $x$  le sera aussi:  
$$\{\mathcal{P}\} \vdash x.$$
- L'inférence logique est équivalente à une machine de Turing: tout calcul peut être effectué en n'utilisant que l'inférence logique.

⇒ idée de programmation déclarative et logique: on n'exprime que les connaissances nécessaires pour la solution, et c'est un algorithme général qui effectue les inférences pour résoudre les problèmes. (exemple: PROLOG)

## Exemple douanier...

Connaissances:

1.  $(\text{vin}(m) \wedge <-2\text{-litres}(m)) \Rightarrow \text{petite-quantite}(m)$
2.  $(\text{cognac}(m) \wedge <-1\text{-litre}(m)) \Rightarrow \text{petite-quantite}(m)$
3.  $<-100\text{-Frs}(m) \Rightarrow \text{petite-quantite}(m)$
4.  $(\text{petite-quantite}(m) \wedge \text{adulte}(v)) \Rightarrow \text{hors-tax}(m,v)$

donc, par exemple:

$\text{vin}(b), <-2\text{-litres}(b), \text{adulte}(\text{Charles}) \vdash$   
 $\text{hors-tax}(b, \text{Charles})$   
 $\text{cognac}(b), <-100\text{-Frs}(b) \vdash ???$

# Règles d'inférence

L'inférence logique se fait en utilisant des *règles*. Un premier jeu de règles d'inférence est donné par les règles d'équivalence des expressions. D'autres règles unidirectionnelles:

- **modus ponens:**  $(p \Rightarrow q), p \Rightarrow$  inférence  $q$
- **introduction et:**  $p, q \Rightarrow$  inférence  $p \wedge q$
- **élimination et:**  $p \wedge q \Rightarrow$  inférence  $p$   
inférence  $q$
- d'autres règles plus ou moins générales.

# Une notation pour les preuves

- Une preuve est une séquence d'inférences successives dont le résultat final est la conclusion cherchée.
- Chaque inférence d'une preuve se justifie par une règle d'inférence et une ou plusieurs propositions

Notation pour la trace d'une preuve:

	<i>proposition</i>	<i>justification</i>
1.	vin	prémisse
2.	$\text{vin} \Rightarrow \text{alcool}$	prémisse
3.	alcool	MP 1 2
4.	...	....

## Exemple de preuve

<i>proposition</i>	<i>justification</i>
1. cognac	prémisse
2. <-1-litre	prémisse
3. (cognac $\wedge$ <-1-litre) $\Rightarrow$ petite-quantité	prémisse
4. cognac $\wedge$ <-1-litre	IE 1 2
5. petite-quantité	MP 3 4

# Demandons à ChatGPT

1.  $(\text{vin}(m) \wedge <-2\text{-litres}(m)) \Rightarrow \text{petite-quantite}(m)$
2.  $(\text{cognac}(m) \wedge <-1\text{-litre}(m)) \Rightarrow \text{petite-quantite}(m)$
3.  $<-100\text{-Frs}(m) \Rightarrow \text{petite-quantite}(m)$
4.  $(\text{petite-quantite}(m) \wedge \text{adulte}(v)) \Rightarrow \text{hors-taxe}(m, v)$

Est-ce que l'inférence suivante est juste:

$\text{vin}(b), <-2\text{-litres}(b), \text{adulte}(\text{Charles}) \vdash \text{hors-taxe}(b, \text{Charles})$

ChatGPT:

*...Les prémisses données sont :*

$\text{vin}(m) \wedge <-2 - \text{litres}(m)$

$\text{adulte}(\text{Charles})$

*Appliquons les règles d'inférence :*

*À partir de la première prémisses, nous pouvons dire que le vin est en petite quantité :  $\text{vin}(b) \wedge <-2 - \text{litres}(b) \Rightarrow \text{petite} - \text{quantite}(b)$ .*

*À partir de la quatrième prémisses, nous pouvons dire que si quelque chose est en petite quantité et qu'une personne est adulte, alors cet objet est hors-taxe :  $(\text{petite} - \text{quantite}(m) \wedge \text{adulte}(v)) \Rightarrow \text{hors} - \text{taxe}(m, v)$ .*

*Maintenant, nous pouvons appliquer la modus ponens sur la prémisses 1 pour obtenir  $\text{petite} - \text{quantite}(b)$ . Ensuite, en combinant cela avec la prémisses 2, nous pouvons conclure  $\text{hors} - \text{taxe}(b, \text{Charles})$ .*

*Ainsi, l'inférence est correcte.*



# D'une preuve à son explication

"Pourquoi hors-taxé?"

⇒ parce-que:

① petite-quantité

② adulte

③ Règlement:

$\text{adulte} \wedge \text{petite-quantité} \Rightarrow \text{hors-taxé}$

Poursuivre récursivement:

"Pourquoi petite-quantité?"

# Mise à jour

Introduction d'une nouvelle substance:

$$(\text{parfum}(m) \wedge \text{<-50-ml}(m)) \Rightarrow \text{petite-quantité}$$

Introduction d'une nouvelle condition:

$$\begin{array}{l} (\text{petite-quantité}(m) \wedge \text{non-resident}(v)) \Rightarrow \\ \text{hors-taxe}(m, v) \end{array}$$

$\Rightarrow$  adjonctions de règles

# Systèmes à base de connaissances

- Calculer les taxes + expliquer pourquoi = intelligence!
- Si le nombre de règles devient élevé  $\Rightarrow$  difficile à maîtriser dans un logiciel classique.
- Règles  $\simeq$  données  $\Rightarrow$  apprentissage  $\simeq$  addition de données.

# Manipulation de connaissances

- Application de règles d'inférence = *déduction*.
- Utile pour simuler un raisonnement.

Autres possibilités:

- *chercher* selon des critères logiques.
- *apprendre* de nouvelles connaissances.

# Résolution de problèmes

- Fiduciaire: comment payer moins de taxes?
- ⇒ quels objets nous feront payer moins?
- ⇒ *recherche* entre différentes possibilités.
- Explosion combinatoire ⇒ approximations.

# Apprentissage

- Supposons que les textes de la loi ne soient pas accessibles.  
⇒ on peut les deviner en observant les douaniers.  
⇒ apprentissage automatique.
- Le plus fiable s'il y a beaucoup de d'exemples: "Big Data".

# Trois modes d'inférence

## déduction:

$$\begin{array}{l} \text{oiseau(Titi)} \\ (\forall x) \text{oiseau}(x) \Rightarrow \text{chante}(x) \\ \hline \text{chante(Titi)} \end{array}$$

## abduction:

$$\begin{array}{l} \text{chante(Titi)} \\ (\forall x) \text{oiseau}(x) \Rightarrow \text{chante}(x) \\ \hline \text{oiseau(Titi)} \end{array}$$

## induction:

$$\begin{array}{l} \text{oiseau(Titi)}, \text{oiseau(Fred)} \\ \text{chante(Titi)}, \text{chante(Fred)} \\ \hline (\forall x) \text{oiseau}(x) \Rightarrow \text{chante}(x) \end{array}$$

# Utilité des inférences

- Dédution: raisonnement automatique, programmation classique.
- Abduction: recherche d'une solution qui correspond à des critères logiques.
- Induction: apprentissage de nouvelles connaissances à partir d'exemples.



# Contenu de ce cours

3 parties:

- ① déduction: moteurs d'inférence basés sur des règles et applications dans des systèmes experts.
- ② abduction: algorithmes de recherche, satisfaction de contraintes et applications en diagnostic et planification.
- ③ induction: algorithmes d'apprentissage à partir d'exemples et applications pour la classification.

voir <http://moodle.epfl.ch/>

# Spécificité du cours

- Le cours est centrée sur la pratique: les méthodes sont accompagnées d'implémentations simplifiées en PYTHON.
- Les sujets sont sélectionnées pour bien illustrer les différents types de techniques et induire une capacité de choisir.
- Chaque méthode est est important pour rendre possible certains applications pratiques; un exemple est fourni à la fin de chaque leçon.
- Par contre, des sujets qui n'ont que peu d'applications sont omis.

# Exercices

Le cours comporte des exercices de programmation en PYTHON.

Importance:

- comprendre les méthodes.
- développer une intuition pour la complexité
- référence pour plus tard

Solutions sont données, pas de notation.

Les exercices ne sont pas optionels: ils aident fortement à comprendre.

# Python

L'Intelligence Artificielle a besoin de

- données symboliques
- structures dynamiques

Python est un langage moderne qui offre ces possibilités.  
De nombreuses sociétés basent leur logiciel sur Python, comme par exemple Google.

<http://www.python.org/>

# Dates importantes

- 31 mars, 17:15-18:15: Test intermédiaire
- 26 mai, 17:15-20:00: Examen final  
(15:15-18:00 si salles disponibles)

# Test intermédiaire

- Les exercices ne sont pas corrigés ni notés, mais des solutions sont disponibles.
- Le progrès dans le cours et les exercices est contrôlé par un test intermédiaire qui contient 1-2 questions sur les exercices et 1-2 questions sur les cours des semaines précédentes.
- Durée: 1h.
- But: contrôler la progression pour éviter de mauvaises surprises à l'examen final.

# Notation

2 examens:

- test intermediaire (31 mars): sur la programmation et théorie (30%).
- théorique final (26 mai): sur la théorie/cours (70%).

# Résumé

- Intelligence  $\simeq$  systèmes à base de connaissances.
- Formalisation de la connaissance utilisant la logique des prédicats.
- Calcul  $\Rightarrow$  inférence logique.
- 3 types d'inférence: déduction, abduction, induction.