



Règles et introduction aux systèmes experts

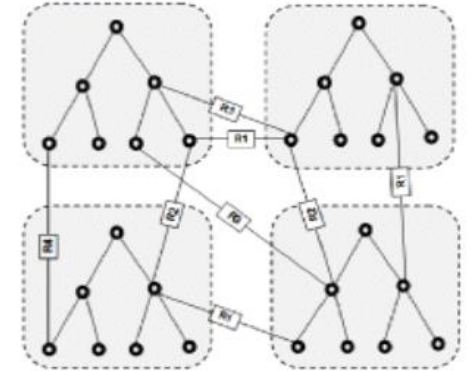
Dr. NSENGE MPIA HERITIER, Ph.D

Précédemment



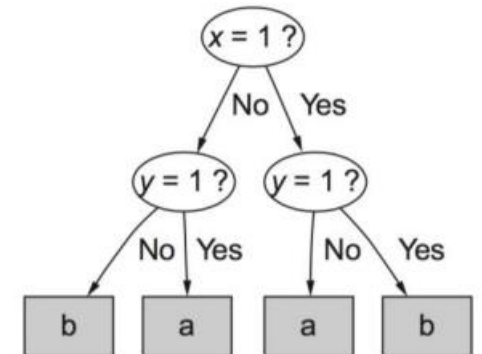
- **Web sémantique**

- Terminologie et normalisation/connexion sémantique
- Liaison entre bases de données, ontologies, données, recherche sur le web
- HTML, XML - RDF, OWL
- Logiques de description



- **Arbres**

- Représentation des connaissances d'experts pour l'induction via l'apprentissage automatique
- Terminologie, structure, liens avec la logique
- Construction d'arbres de décision / induction



- **Autres ML basés sur les arbres**

- Ensembles
- Bagging
 - Random Forrests
- Boosting
 - GBM
 - XGBoost



Plan de la leçon

- Règles
- Systèmes experts :
 - Quoi, pourquoi, quand ?
- Introduction aux systèmes basés sur des règles



Règles

Représentations des connaissances dans l'IA

- **Représentation logique** : (déjà couverte)
 - Moins efficace lorsque nous voulons faire une classe restreinte d'inférences
 - Difficulté à représenter :
 - Le sens commun
 - Le temps, les croyances ou l'incertitude
 - Des logiques spéciales telles que les logiques temporelles et modales permettent la représentation
 - Mais le raisonnement reste souvent inefficace
- **Réseaux sémantiques et cadres** : (c-à-d **objets structurés**) (déjà couvert)
 - Représentations plus simples et plus naturelles
 - Ne nécessite pas de spécification formelle de la sémantique
 - Représente la connaissance factuelle des **classes d'objets** et de leurs **propriétés**
 - **Représentations déclaratives et statiques des faits/rerelations**
- **Règles (c'est-à-dire systèmes basés sur des règles)** :
 - Expressions IF/THEN
 - Certaines similitudes avec les implications logiques
 - L'accent est davantage mis sur ce qui est fait avec les règles que sur ce qu'elles signifient
 - C'est-à-dire que l'accent est mis sur les **aspects procéduraux** plutôt que sur les aspects déclaratifs.

Règles : Principes de base



- Représentation des connaissances sous forme de "règles", c'est-à-dire

- Expressions **IF-THEN**:

If <condition (s)> **then** <conclusion>

- Expressions **Condition-Action**:

If <condition (s)> **then** <action>

- **Implications** effectives de la logique propositionnelle ou du premier ordre
- Exemple simple :
 - SI le feu de circulation est rouge ET que vous vous êtes arrêté, ALORS vous pouvez tourner à droite

- **Exemple biomédical** :

- **SI** le patient présente des niveaux élevés de ferritine dans son sang
 ET que le patient présente la mutation Cys282→Tyr dans le gène HFE,
 ALORS, le patient est atteint d'hémochromatose*.

Proposition 1: **Vrai**

Proposition 2: **Vrai**

Proposition 3: **Vrai**

- Condition :
 - Aussi connue comme prémisse, **antécédent**, variables indépendantes
- Conclusion :
 - Aussi connue comme action, résultat, **conséquence**, but, variable dépendante

Règles : Antécédents



- "IF"
- **Antécédent** : représente une série de tests
 - Tout comme les tests aux nœuds d'un arbre de décision
 - Il existe des liens naturels entre les règles et les arbres que nous identifierons plus tard
 - Les règles sont une alternative populaire aux **arbres de décision**
- Les tests sont généralement logiquement ET ensemble
 - Comme les "**clauses définies**" utilisées pour l'enchaînement de l'inférence
 - Mais peuvent également être des expressions logiques générales.

Règles : Conséquences

- "THEN"
- Conséquence :
 - Conclusions, ensemble de conclusions
 - Actions, ensemble d'actions
 - Distribution de probabilité attribuée par une règle spécifiant le soutien/l'incertitude dans un conséquent donné
 - Nous examinerons ceci plus en détail lorsque nous aborderons le "raisonnement avec incertitude"
- Des conflits surviennent si des conclusions différentes s'appliquent
 - Stratégies nécessaires pour traiter la résolution des conflits

Plus d'informations sur les conséquences

- **Relation :**
 - Si le "réservoir d'essence" est vide,
 - ALORS la voiture est morte
- **Recommandation :**
 - Si la saison est l'automne
 - ET que le ciel est nuageux
 - ET que les prévisions annoncent de la bruine,
 - ALORS "prendre un parapluie"
- **Directive:**
 - Si la voiture est morte
 - ET que le "réservoir d'essence" est vide
 - ALORS l'action est de "faire le plein".

Plus d'informations sur les conséquences (Cont.)



- **Stratégie :**
 - Si la voiture est morte
 - ALORS l'action consiste à "vérifier le réservoir d'essence«
 - l'étape 1 est terminée.

 - Si l'étape 1 est terminée
 - ET que le "réservoir d'essence" est plein
 - ALORS l'action consiste à "vérifier la batterie "
 - l'étape 2 **est terminée**
- **Heuristique (modèle de prédiction - classification/régression) :**
 - Si le déversement est liquide
 - ET que le "pH du déversement" est < 6
 - ET que l'"odeur du déversement" est celle du vinaigre
 - ALORS la "matière du déversement" est l'"acide acétique".

Résumé : Caractéristiques des règles



	Première partie	Deuxième partie
Noms	Prémisse Antécédent Situation IF	Conclusion Conséquence Action THEN
Nature	Conditions, similaires aux connaissances déclaratives	Résolutions, similaires aux connaissances procédurales
Taille	Peut avoir plusieurs IF	En général, une seule conclusion
Proposition	Propositions ayant ET	Toutes les conditions doivent être réunies pour qu'une conclusion soit vraie
	Propositions ayant OU	Si une condition est vraie, la conclusion est vraie

Règles simples avec relations

- Ajout de relations de condition (par exemple =, >, <, ¬)
- Règles comparant une valeur d'attribut à une constante (par exemple température < 45)
 - Ces règles sont appelées **règles propositionnelles**
 - Elles ont le même pouvoir d'expression que la logique propositionnelle
 - Exemple :
 - **SI** la cote de crédit est élevée **ET** que le salaire est supérieur à 3000 USD, **OU** que les actifs sont supérieurs à 7500 USD, **ET** que les antécédents de paiement ne sont pas "médiocres", **ALORS** approuver un prêt jusqu'à 1000 USD, et classer le prêt dans la catégorie "B".
- Que faire si le problème implique des relations entre les attributs ?
 - Les règles propositionnelles ne permettent pas de l'exprimer
 - Une représentation plus expressive est nécessaire

Relations entre les attributs

- Relations standard : $=$, $<$, $>$
- La comparaison des attributs entre eux permet d'appliquer des règles telles que celles-ci :

```
IF largeur > hauteur THEN couché  
IF hauteur > largeur THEN debout
```

- Cette description se généralise mieux à de nouvelles données
- Mais la recherche de relations entre les attributs peut s'avérer coûteuse
- Solution simple : **ajouter** des attributs supplémentaires (c'est-à-dire construire des caractéristiques)
 - par exemple, un attribut binaire "la largeur **est-elle** $<$ hauteur ?"
- Utilisation de variables et de relations multiples (similaire à la logique du premier ordre) :

```
IF hauteur_et_largeur_de(x,h,l) AND h > l THEN debut(x)
```

Quelques formes de règles moins courantes



- **Conclusion, condition IF :**
 - Votre risque d'être contrôlé est élevé, **SI** votre revenu est élevé
- **Inclusion de ELSE :**
 - **SI** votre revenu est élevé, **OU** si vos déductions sont inhabituelles, **ALORS** votre risque d'être contrôlé est élevé, **OU SINON** votre risque d'être contrôlé est faible
 - **Semblable au IF: THEN** boucle dans le codage
- La partie action peut **déclencher d'autres règles** ou des **actions multiples** :
 - **ALORS** "approuver le prêt" et "référer à un agent".

Les règles en tant que base de connaissances (KB)



- Nous commençons maintenant à considérer les règles comme un groupe collectif qui forme **une base de connaissances** à partir de laquelle nous pouvons travailler.
- Considérons qu'il existe deux façons d'exécuter un ensemble de règles.
 - **Tout d'abord**, vous pouvez disposer d'un ensemble **ordonné de règles**.
 - Cet ensemble est également connu sous le nom de **liste de décisions**.
 - Dans cette situation, l'ordre des règles est important pour la manière dont elles sont interprétées.
 - **À l'inverse**, nous pouvons également avoir un ensemble de **règles non ordonné**.
 - Dans ce cas, les règles peuvent se chevaucher et conduire à des conclusions différentes pour un même cas. En d'autres termes, elles introduisent des **conflits**.
 - Cependant, il y a **des avantages à avoir un ensemble de règles non ordonné** par rapport à un ensemble de règles ordonné.
- Une autre chose que vous pouvez faire avec une base de connaissances de règles est de vous appuyer sur des **règles par défaut**.
 - Il est courant de s'appuyer sur des règles par défaut lorsqu'on utilise un ensemble de règles ordonnées.
 - Fondamentalement, l'idée est de créer des règles pour d'autres résultats et, si elles ne sont pas satisfaites, de déclencher une règle par défaut avec une action ou une conclusion par défaut.
 - Voici un exemple d'utilisation d'une **règle par défaut**. Dans ce cas, nous avons deux règles, la seconde étant une règle par défaut qui se déclenche lorsque **x** est vide ou qu'aucune autre règle ne se déclenche.

```
IF cest_au_dessus_de(x,z) and  
   hauteur_et_largeur_de(z,h,w) and h > w  
   and est_reste_de(x,y) and debout(y)  
   then debout(x)  
IF empty(x) then debout(x)
```

Règles par défaut : Classe booléenne



- Examinons un peu plus en détail cette idée de règles par défaut dans le contexte de problèmes où nous avons une classe booléenne ou un résultat booléen.
- L'hypothèse ici est que si notre instance n'appartient pas à la classe "oui", alors elle appartient par défaut à la classe "non".
- Ce qui est bien dans cette situation, c'est qu'il suffit d'implémenter des règles qui indiquent quand nous avons une situation où la classe est "oui" et de s'appuyer sur la règle par défaut pour toutes les situations où la classe est "non".
- Dans l'exemple ci-dessous, les deux classes qui nous intéressent sont a et b.
 - Nous n'avons donc prévu que des règles indiquant que nous faisons partie de la classe "a",
 - sinon la règle par défaut de la classe "b" s'applique. Dans cette situation, l'ordre des règles n'est pas important et il n'y a pas de conflit.

```
If x = 1 and y = 1 then class = a  
If z = 1 and w = 1 then class = a  
Otherwise class = b
```

- De plus, cet ensemble de règles est écrit sous une forme normale disjonctive. Plus précisément, nous avons deux règles, toutes deux avec des antécédents utilisant AND, mais l'une ou l'autre de ces règles peut être vraie pour décider que nous sommes de la classe "a"

Des règles avec des exceptions

- **Idée** : permettre aux règles d'avoir des exceptions
- **Exemple** : règle pour les données de la fleur d'iris

```
If petal-length  $\geq$  2.45 and petal-length  $<$  4.45 then Iris-versicolor
```

- **Nouvelle instance:**

Sepal Length	Sepal Width	Petal Length	Petal Width	Type
5.1	3.5	2.6	0.2	?

- **Règle modifiée:**

```
If petal-length  $\geq$  2.45 and petal-length  $<$  4.45 then Iris-versicolor  
  EXCEPT if petal-width  $<$  1.0 then Iris-setosa
```

Un exemple plus complexe



- Exceptions aux exceptions aux exceptions ...

```
default: Iris-setosa
except if petal-length  $\geq$  2.45 and petal-length  $<$  5.355
    and petal-width  $<$  1.75
    then Iris-versicolor
        except if petal-length  $\geq$  4.95 and petal-width  $<$  1.55
            then Iris-virginica
            else if sepal-length  $<$  4.95 and sepal-width  $\geq$  2.45
                then Iris-virginica
        else if petal-length  $\geq$  3.35
            then Iris-virginica
            except if petal-length  $<$  4.85 and sepal-length  $<$  5.95
                then Iris-versicolor
```

Avantages de l'utilisation des exceptions



- Les règles peuvent être mises à jour de manière incrémentale
 - Il est facile d'intégrer de nouvelles données
 - Il est facile d'intégrer les connaissances du domaine
- Les humains pensent souvent en termes d'exceptions
- Chaque conclusion peut être considérée uniquement dans le contexte des règles et des exceptions qui y conduisent
 - La propriété de localité est importante pour comprendre les grands ensembles de règles
 - Les ensembles de règles "normaux" n'offrent pas cet avantage.

Règles : Vue d'ensemble



- Chaque règle d'une base de connaissances représente **un morceau autonome d'expertise**
 - Sur le plan conceptuel, les règles individuelles sont souvent logiquement reliées par un **OU**
- Lorsqu'elles sont combinées et transmises au **moteur d'inférence**, **l'ensemble des règles se comporte de manière synergique**
- Les règles peuvent être considérées comme une **simulation** du **comportement cognitif** des **experts humains**
 - Les règles représentent un modèle du comportement humain réel
- Sont des techniques prédominantes pour formuler les **connaissances** dans les **systèmes experts**, souvent en conjonction avec des cadres
 - c-à-d les **systèmes basés sur des règles** et les **systèmes de production** (**prochaine leçon**)

Inférence de règles

- Maintenant que nous connaissons ce que sont les règles et comment elles peuvent être représentées, examinons à nouveau la façon dont elles peuvent être utilisées pour faire des inférences.
- En ce qui concerne les problèmes que nous pourrions essayer de résoudre, il existe deux principaux types de tâches qui impliquent l'utilisation de l'un ou l'autre des deux régimes de contrôle :
 - **Chaînage avant** (axé sur les données/data driven) :
 - Commencer par les faits, déterminer les règles applicables et en appliquer une
 - **Chaînage arrière** (axé sur les objectifs/goal driven) :
 - Rechercher des règles qui décomposent l'objectif ; résoudre des objectifs plus petits.
- L'un est-il meilleur que l'autre ?
 - Pas de réponse générale possible
 - Cela dépend de l'application.

Correspondance/matching des règles



- Une partie importante de l'utilisation des règles pour faire des inférences est le processus de correspondance des règles.
- La mise en correspondance consiste à déterminer les connaissances applicables à une situation donnée.
- En d'autres termes, il s'agit de déterminer, sur la base des connaissances disponibles, quelles sont les règles qu'il convient d'examiner plus avant.
- Les règles sont "activées" ou déclenchées en fonction du type d'inférence ou de chaînage utilisé.
 - Dans le chaînage avant, nous essayons de faire correspondre l'antécédent de la règle à la base de faits dont nous disposons.
 - Dans le chaînage arrière, nous essayons de faire correspondre la conséquence de la règle aux faits et à la base de règles pour voir si nous pouvons prouver que notre conséquence est vraie ou trouver d'autres règles qui correspondent à cette conséquence.
 - Nous pouvons alors utiliser ces autres règles pour tenter de prouver leur véracité.
- Cependant, c'est ce processus de correspondance qui peut être l'étape la plus coûteuse en termes de calcul.

Correspondance/matching des règles (Cont.)



- En outre, la mise en **correspondance** produit souvent ***plus d'une règle applicable*** qui doit être prise en considération et, dans ce cas, elle doit être inscrite sur un agenda de règles à examiner dans l'ordre.
- Lorsque plusieurs règles sont applicables, on dit que ***ces instances de règles sont en conflit*** et nous devons utiliser la ***résolution des conflits*** pour en choisir une à **activer**.
- **L'activation** est un terme utilisé dans le **chaînage avant** pour décrire ***l'exécution de la conséquence d'une instance de règle applicable***
- en d'autres termes, si les antécédents d'une règle sont satisfaits, la conséquence est ajoutée en tant que nouveau fait par le biais de l'activation.
- C'est également au cours du processus de mise en **correspondance** que les variables et les règles acquièrent des valeurs, ***comme l'instanciation de la logique***.

Avantages des règles



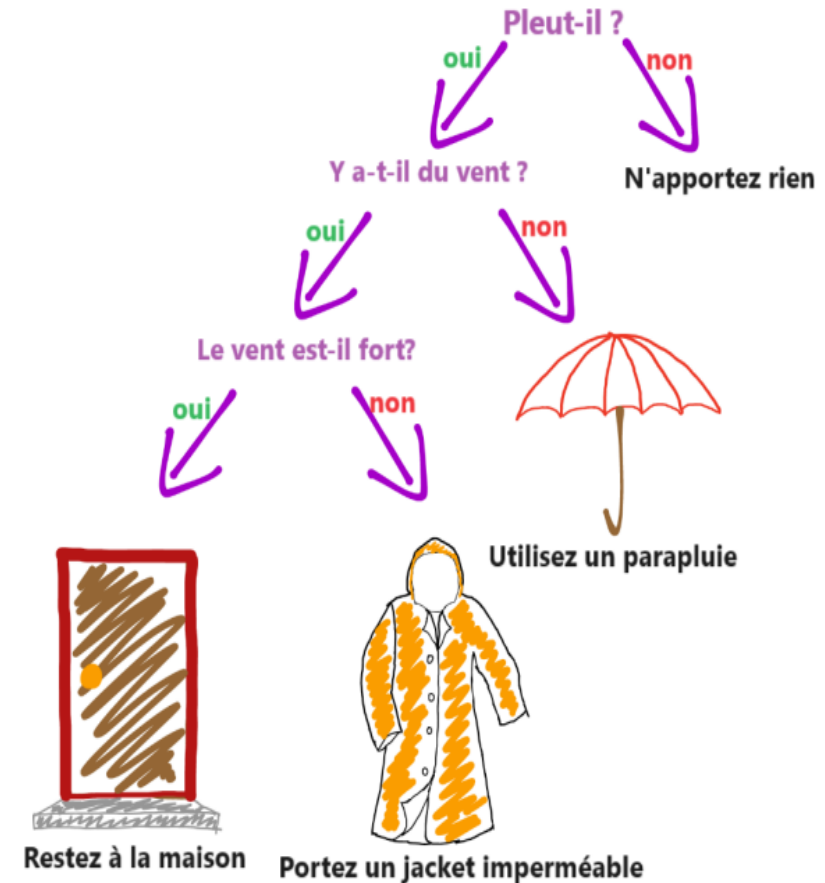
- Simple et facile à comprendre
 - Correspond à la façon dont les gens pensent souvent à la connaissance
 - Très expressif
- Possibilité d'implémentation directe dans les ordinateurs
 - Connaissance modulaire
 - Facile à écrire et à déboguer par rapport à d'autres représentations
- Fondements formels pour certaines variantes

Inconvénients des règles

- Les mises en œuvre simples sont très inefficaces
- Certains types de connaissances ne sont pas facilement exprimés dans ces règles
- Les grands ensembles de règles deviennent difficiles à comprendre et à maintenir
- Ils peuvent être gourmands en mémoire
- Ils peuvent être gourmands en **calcul**
- Ils sont parfois difficiles à déboguer

Règles de décision : Interprétation des arbres

- L'arbre de décision peut être "**linéarisé**" en **règles de décision**
 - Une règle par chemin de la racine à la feuille.
 - **Résultat** de la règle = nœud de la feuille
- Règle :
 - **If** [condition1] and [condition2] and [condition3]
Then: **résultat**
- Exemple :
 - **If** [ne pleut pas] **Then:** **n'apportez rien**
 - **If** [pleut] and [pas de vent] **Then:** **utilisez un parapluie**
 - **If** [pleut] and [il y a le vent] and [pas un vent fort] **Then:** **portez un jacket imperméable**



Des arbres aux règles



- Facile : conversion d'un arbre en un ensemble de règles
- Une règle pour chaque feuille de l'arbre
- L'arbre définit un ensemble de règles uniques représentant chacune un chemin dans l'arbre, de la racine à une feuille différente
 - L'antécédent est la conjonction des valeurs à chaque nœud
 - Le conséquent est la classe/l'action assignée par la feuille
- L'ordre d'exécution n'a pas d'importance
- Mais : les règles résultantes peuvent être inutilement complexes
- **Élagage** pour supprimer les tests/règles redondants

Des règles aux arbres

- Plus difficile : transformer un ensemble de règles en arbre
- L'arbre ne peut pas facilement exprimer la disjonction entre les règles
 - Bien que l'inverse soit vrai
 - Une base de règles peut ne pas avoir le même sous-ensemble/supérieur d'attributs requis pour assembler une structure arborescente
- Considérez :
 - Lorsque l'on passe d'un arbre à des règles :
 - Toutes les règles auront le nœud racine
 - Et si ce n'était pas le cas d'un ensemble de règles initial ?

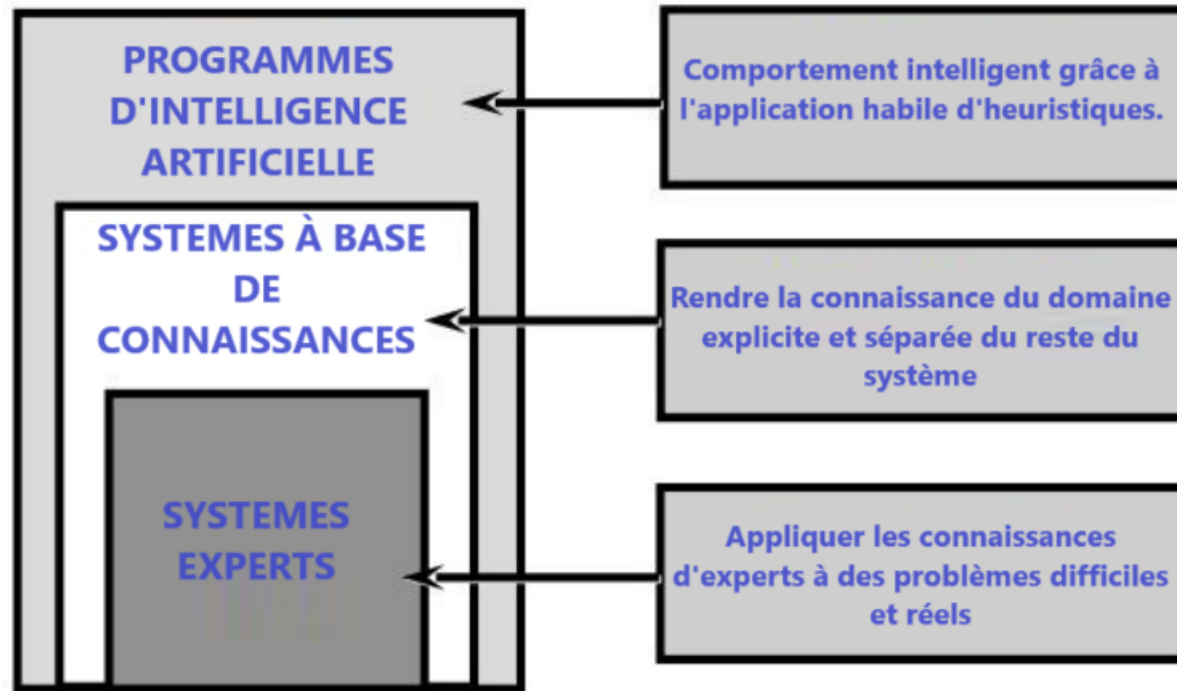


Systemes experts : quoi, pourquoi, quand ?

Définitions: Systèmes Experts

- **Aussi connu comme:**
 - Système basé sur la connaissance
 - Système d'aide à la décision basé sur la connaissance
 - Système intelligent d'aide à la décision
 - Système basé sur des règles
 - Système de production
- Un système informatique qui **émule**, ou agit à tous égards, avec les capacités **de prise de décision d'un expert humain** [dans un domaine limité]. **(Feigenbaum)**
- "Un système informatique qui fonctionne en **appliquant un mécanisme d'inférence** à un ensemble de **compétences spécialisées** représentées sous forme de "connaissances". **Goodall (1985)**
- "Un programme destiné à **porter des jugements raisonnés** ou à **fournir une assistance** dans un domaine complexe dans lequel les **compétences humaines sont faillibles ou rares**." **Lauritzen et Spiegelhalter (1988)**
- "Un programme conçu **pour résoudre des problèmes** à un niveau **comparable** à celui d'un **expert humain** dans un domaine donné". Cooper (1989)
- "Système **incarnant l'expertise d'un spécialiste** (par exemple, les connaissances médicales)". **Cawsey (1998)**

Hiérarchie des termes : De SE à l'IA



- Les systèmes basés sur la connaissance sont une bonne alternative :
 - Certains systèmes nécessitent des connaissances, mais pas des connaissances d'expert, par exemple
 - Nous pourrions vouloir mettre en œuvre un système de planification des activités quotidiennes qui ne nécessite aucune expertise
- Nous utiliserons le terme "**système expert**", mais considérons qu'il est à peu près synonyme de système à base de connaissances dans le cadre de ce cours.

Caractéristiques générales des systèmes experts

- À un niveau très élevé, quelles sont les **caractéristiques générales** des systèmes experts ?
- Tout d'abord, ils utilisent le ***raisonnement symbolique***, en d'autres termes, les connaissances doivent être représentées symboliquement par des éléments tels que des **arbres, des règles, des cadres ou des logiques de description**.
- Les systèmes experts appliquent également des **heuristiques** pour guider le raisonnement et tenter de réduire la zone de **recherche** d'une solution.
- La **recherche** est un aspect essentiel qui sous-tend le fonctionnement de ces systèmes, l'objectif étant de trouver les bonnes ou les meilleures solutions.
 - Nous aborderons la recherche de manière plus explicite dans les leçons ultérieures.

Caractéristiques des systèmes experts vs autres IA

- Traiter des sujets d'une complexité considérable
 - Généralement dans des domaines étroits et spécialisés
 - Solutions de niveau expert à des problèmes complexes
- **Connaissance est séparée du traitement** (c-à-d inférence)
 - Noyau réutilisable
- Performance de haute qualité
 - La vitesse ne signifie pas grand-chose si la réponse est fausse
- Doit être rapide
 - Une réponse correcte ne signifie pas grand-chose si elle arrive trop tard
 - Par exemple, un patient est décédé ou une centrale nucléaire a explosé.
- Capable d'expliquer et de justifier les solutions
 - c'est-à-dire interprétable/compréhensible
- Flexible :
 - Intégration de nouvelles informations, correction d'anciennes informations

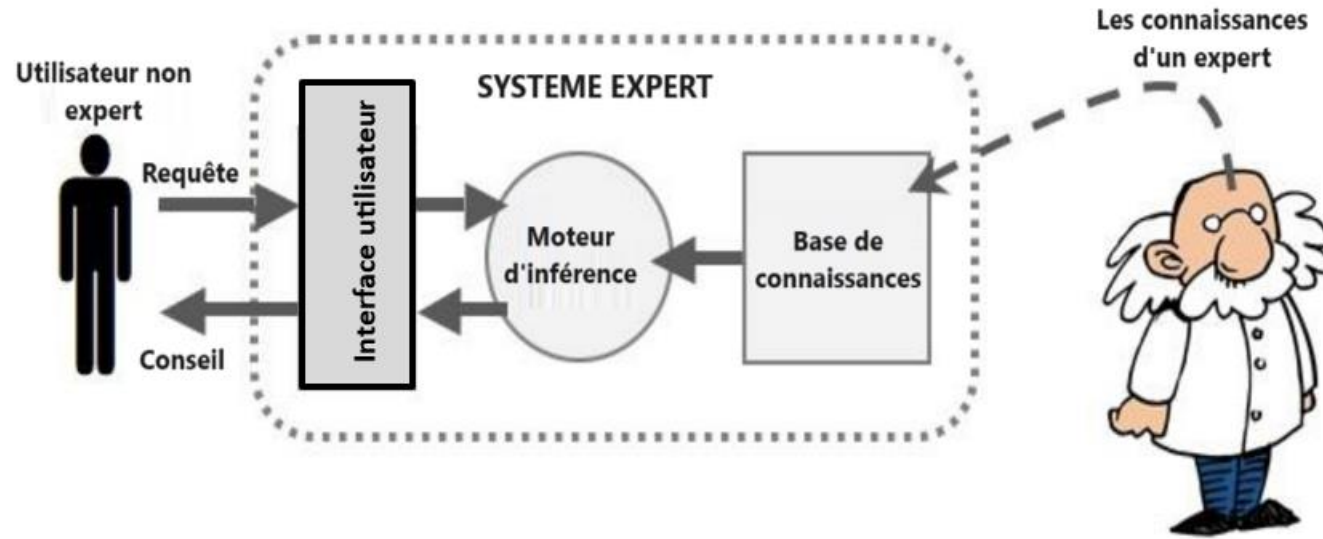
Systèmes experts vs programmes informatiques classiques

- Voici un tableau qui tente de mettre en évidence certaines des principales différences entre la réflexion sur un **système expert** et la réflexion sur des **programmes informatiques traditionnels**.

- Par exemple, en ce qui concerne le contrôle et les données, la **programmation conventionnelle intègre implicitement les deux**, alors que les **systèmes experts les séparent explicitement**, comme vous venez de l'entendre.
- Dans la programmation conventionnelle, une **solution est généralement obtenue à l'aide d'un algorithme**, alors que dans un système expert, elle est **obtenue par inférence**.

Caractéristiques	Programmation classique	Systèmes experts
Contrôle par...	Ordre des déclarations	Moteur d'inférence
Contrôle et données	Intégration implicite	Séparation explicite
Force de contrôle	Forte	Faible
Solution par ...	Algorithmes	Règles et inférence
Recherche de solution	Petite ou aucune	Grande
Résolution de problèmes	L'algorithme est correct	Règles
Input (Entrée)	Supposé correct,	Incomplet, incorrect
Input inattendu	Difficile de traiter avec	Très réactif
Output	Toujours correct	Varie en fonction du problème
Explication	Aucune	Habituellement
Applications	Numérique, fichier et texte	Raisonnement symbolique
Exécution	Généralement séquentielle	Règles opportunistes
Conception du programme	Conception structurée	Peu ou pas de structure
Modifiabilité	Difficile	Raisnable
Expansion	Fait par grands sauts	Incrémental

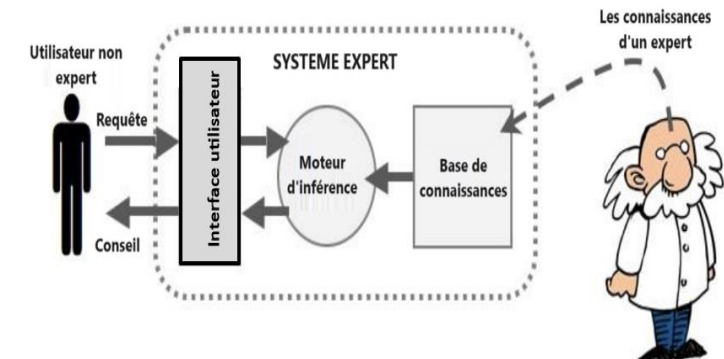
Schéma de haut niveau du système expert



- Objectif des systèmes experts :
 - Remplacer (c'est-à-dire automatiser) ou aider les experts humains à résoudre ou à conseiller des problèmes réels.

Composants clés des systèmes experts

- Un système expert se compose de deux éléments principaux.
 - Le premier est la **base de connaissances** elle-même.
 - C'est là que se trouvent les connaissances représentées de manière symbolique et formelle, qui peuvent prendre de nombreuses formes, comme nous l'avons mentionné : **Règles, cadres, arbres, réseau sémantique, réseau bayésien, etc.**
 - Le deuxième composant clé est le **moteur d'inférence**.
 - Il est responsable du raisonnement.
 - En d'autres termes, il tire des conclusions à partir de la base de connaissances.
 - Il est essentiel que l'approche de raisonnement ou d'inférence soit adaptée à la représentation utilisée.
 - Ainsi, si vous disposez d'un système expert conçu pour fonctionner avec des règles, vous ne pouvez pas simplement les remplacer par des arbres ou des cadres sans vous attendre à devoir résoudre des problèmes majeurs ou sans envisager de revoir complètement la conception du système expert à partir de zéro.



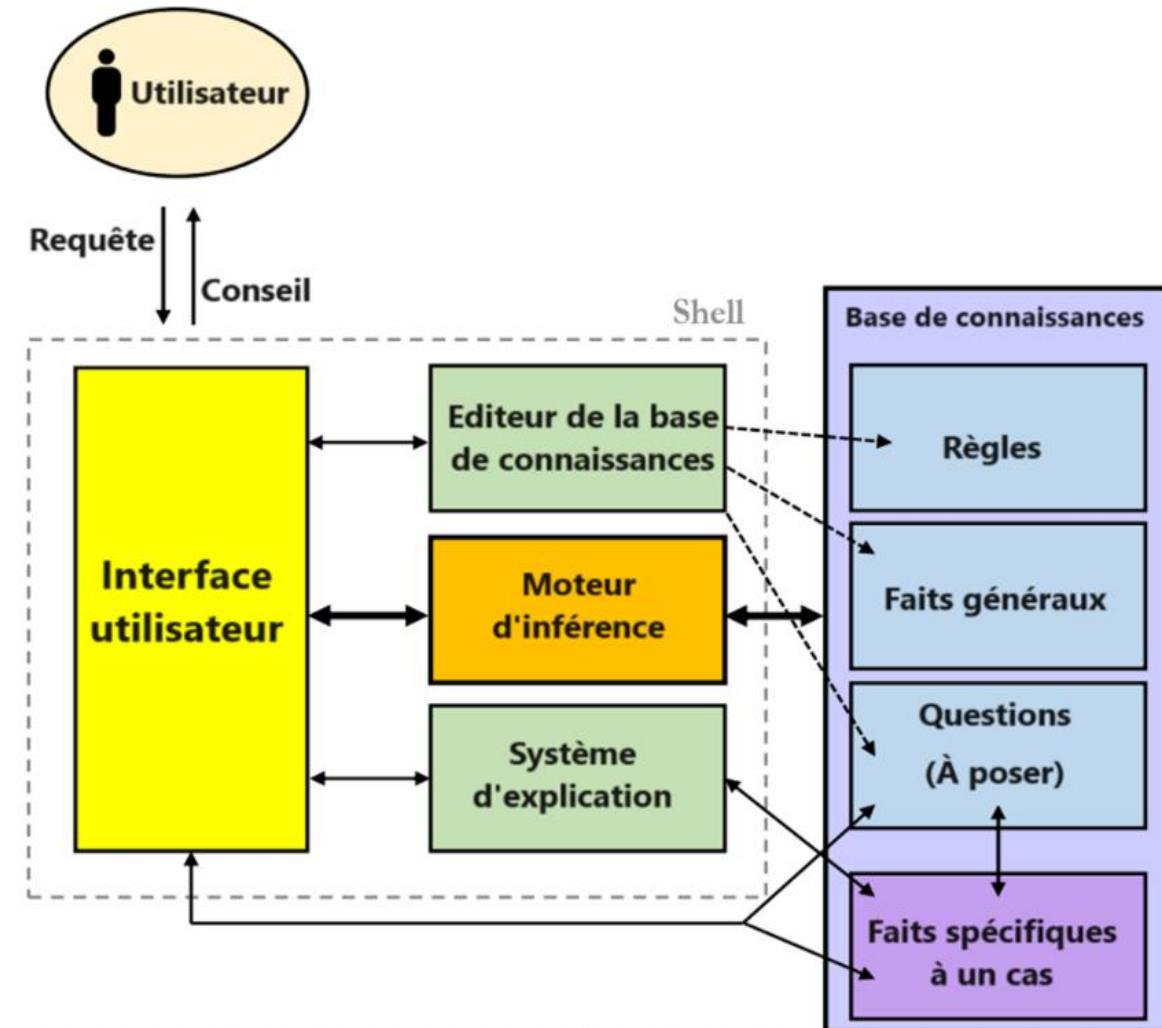
Composants supplémentaires du système expert



- Outre ces deux éléments clés, il existe d'autres éléments communs à un système expert :
 - **Interface utilisateur :**
 - Mécanisme par lequel l'utilisateur et le système communiquent
 - **Éditeur de la base de connaissances :**
 - **Aussi connu comme :** Dispositif d'acquisition de connaissances
 - Moyen automatique pour l'utilisateur d'introduire des connaissances dans le système en contournant la nécessité d'une expertise explicite en matière de codage.
 - **Système d'explication :**
 - **Aussi connu comme :** Justificateur
 - Explique le raisonnement du système expert à l'utilisateur

Schéma général d'un système expert

- Voici un schéma que j'ai élaboré pour essayer de mieux saisir les différents aspects d'un système expert typique:
 - Toujours à l'extérieur, nous avons un **utilisateur** qui peut poser des questions et obtenir des conseils en retour. Il interagit avec une interface utilisateur.
 - L'**interface utilisateur** est connectée de manière importante au moteur d'inférence et éventuellement à un éditeur de base de connaissances et à un système d'explication s'ils sont disponibles dans l'implémentation du système expert.
 - Tous ces éléments appartiennent notamment à un shell.
 - Le **shell** est une partie réutilisable d'un système expert.
 - Le **moteur d'inférence** est connecté à la base de connaissances.
 - La **base de connaissances** est constituée de *règles*, de *faits généraux*, de *questions* qu'il pourrait être nécessaire de **poser à l'utilisateur** et d'un espace de mémoire temporaire pour les faits spécifiques à un cas ou les nouveaux faits qui peuvent être ajoutés lors de l'examen d'une requête spécifique de l'utilisateur.



Pourquoi développer/utiliser un système expert ?

- Expertise humaine :
 - Peu disponible (peu d'experts sur la tâche)
 - Coûteux à développer ou à embaucher
 - Difficile à obtenir rapidement
 - Expertise difficile à appliquer rapidement (tâches complexes)
 - Il n'est pas pratique d'avoir un être humain constamment disponible
 - Peut être perdue :
 - Retraite/quitter son emploi
 - Décès
- Le system expert remplit les fonctions suivantes en ce qui concerne l'expertise :
 - Documenter/préserver
 - Reproduire/mettre à disposition
 - Enseigner/diffuser
- Les ordinateurs ne s'ennuient pas/ne se fatiguent pas/ne se frustreront pas/**ne s'effraient pas**

Grandes catégories de systèmes experts

- En termes d'application, il existe de nombreuses catégories de systèmes experts.
- Par exemple:
 - ils ont été conçus pour traiter les problèmes de **configuration**, c'est-à-dire pour assembler correctement les composants d'un système.
 - ils ont été appliqués au **diagnostic**. En d'autres termes, ils permettent de déduire les problèmes sous-jacents sur la base des éléments observés
 - Etc.

Classe	But de l'application
Configuration	Assembler les composants d'un système de la manière appropriée
Diagnostic	Inférer les problèmes sous-jacents sur la base des éléments observés
Instruction	Enseignement intelligent permettant à l'étudiant de poser des questions sur le pourquoi, le comment et le si, comme s'il était enseigné par un être humain.
Interprétation	Expliquer les données observées
Surveillance	Comparer les données observées aux données attendues pour juger de la performance
Planification	Concevoir des actions pour obtenir un résultat souhaité
Pronostic	Prédire l'issue d'une situation donnée
Remède	Prescrire un traitement pour un problème
Contrôler	Réguler un processus. Peut nécessiter une interprétation, un diagnostic, un suivi, une planification, un pronostic et des remèdes.

Exemples d'applications



- Médecine :
 - Diagnostic des maladies
 - Systèmes d'alerte médicale
 - Gestion des soins de santé
- Finance :
 - Systèmes d'analyse de crédit :
 - Utilisés dans les banques pour l'évaluation de la gestion des risques
 - Conseillers en fonds de pension :
 - Aident les employés à planifier leur retraite grâce à leur analyse qui calcule leurs prestations de retraite en fonction de différents scénarios
 - Services d'assistance automatisés :
 - Permet aux petites entreprises de traiter les demandes des clients de manière plus efficace.
 - Systèmes de surveillance du marché
 - Systèmes de réingénierie des processus d'entreprise
- Fabrication
 - Diagnostic des défauts des machines
 - Robotique
 - Configuration des ordinateurs
 - Identification des déversements de matériaux
- Autres ressources humaines
 - Ressources humaines
 - Traitement des données
 - Données sonar ou mesures géophysiques
 - Systèmes de sécurité intérieure



Introduction aux systèmes experts basés sur des règles

Systèmes experts basés sur des règles

- Les règles sont la représentation la plus courante dans les systèmes experts
 - En particulier, les expressions simples **IF : THEN**
 - Les antécédents des règles ne peuvent être que des conjonctions (**AND**).
- Le moteur d'inférence détermine quels antécédents de règles sont satisfaits.
 - Le côté gauche doit "**correspondre**" à un fait dans la base de connaissances.
 - Le résultat est que le **côté droit** (conséquent) est **activé**.
 - Une règle activée peut générer de nouveaux faits
 - Par conséquent, l'activation d'une règle peut entraîner l'activation d'autres règles.
- Pour des raisons de simplicité, nous nous concentrerons dans ce cours sur les systèmes experts **simples basés sur des règles**.

Quand les systèmes experts basés sur des règles sont-ils préférables ?



- La connaissance est diffuse :
 - Il existe un grand nombre de faits, plus ou moins indépendants les uns des autres.
- La connaissance peut être facilement séparée de son utilisation :
 - Il n'y a pas de dépendance quant à la manière d'utiliser les connaissances.
 - Par exemple, décider des ingrédients d'un produit.
 - Mais nous n'avons pas besoin de décider de la quantité de chaque ingrédients que nous devons mélanger pour obtenir une recette savoureuse.

Systèmes de production

- Il y a un autre terme important à connaître c'est **systèmes de productions**.
- Le terme "**systèmes de production**" est largement issu de l'industrie et, dans ces situations, les conséquences des règles sont généralement des actions.
- Le terme "**systèmes de production**" est généralement associé à la surveillance ou au contrôle des tâches.
- Cependant, cette terminologie n'est pas utilisée de manière cohérente.
- Sur la côte (en Amérique) est, les systèmes de production font généralement référence à des **systèmes experts basés sur des règles** qui utilisent le chaînage avant.
- Mais sur la côte ouest, le terme "**systèmes de production**" peut se référer à n'importe quel type de système expert basé sur des règles.
- Pour ce cours, nous supposons que les systèmes de production font référence aux systèmes experts basés sur des règles qui utilisent le chaînage avant.

Règles de production

- Les règles de production peuvent prendre des expressions **if-then** et les formater de la manière tel que écrit plus bas :
 - if** light is red **then** stop.
 - Nous plaçons l'antécédent sur le côté gauche et le conséquent sur le côté droit.

Règles IF ... THEN

Rule: Red_Light

IF **the light is red** ← Antécédent (côté gauche)

THEN **stop** ← Conséquent (Côté droit)

Rule: Green_Light

IF **the light is green** ← Antécédent (côté gauche)

THEN **go** ← Conséquent (Côté droit)

Règles de production

the light is red ==> **stop** ← antécédent (côté gauche)

the light is green ==> **go** ← Conséquent (côté droit)

Synthèse de la leçon

- Représentation des règles
 - Antécédents et conséquences
 - Formes et relations des règles
 - Valeurs par défaut et exceptions
 - Les bases de l'inférence et de la correspondance
 - Arbres vers règles et vice versa
- Systèmes experts : Quoi, pourquoi, quand ?
 - L'IA pour résoudre des problèmes dans des domaines spécialisés
 - Composants de base
- Systèmes experts basés sur des règles
 - De loin les plus courants et les plus faciles à développer
 - Systèmes de production (et règles)



Sachez qu'être un expert ne se limite pas
à comprendre comment un système est
censé fonctionner. L'expertise s'acquiert
en cherchant à comprendre pourquoi un
système ne fonctionne pas.

— *Brian Redman* —

Travail pratique 3: Rappel

- **Objectifs :**

- Représentation des connaissances, systèmes experts et Experta
- Pratiquer la représentation des cadres
- Apprendre à installer et à utiliser Experta
- Pratiquer la représentation des règles
- Construire un système expert très simple pour le diagnostic
- Apprendre à programmer des questions
- Comprendre le chaînage avant et arrière dans les systèmes experts basés sur des règles

- Le travail de mi-chemin devrait être remis sous peu