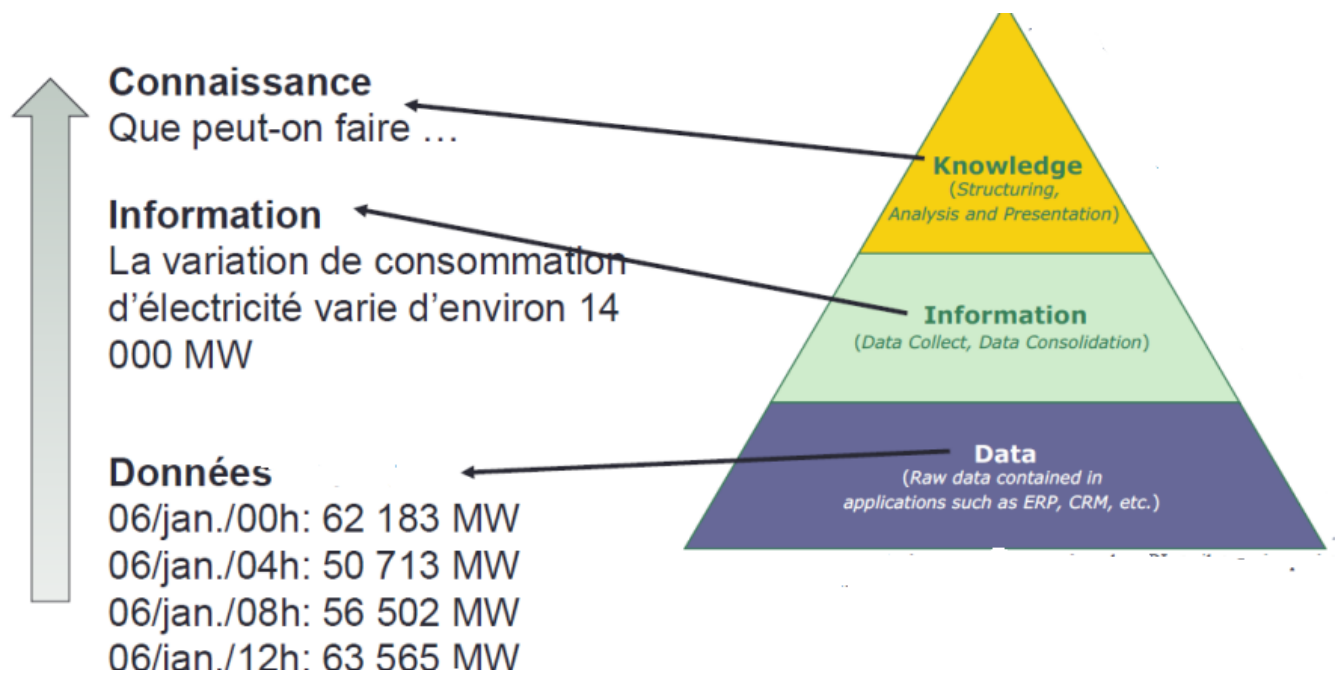


Université Mohamed BOUDIAF de M'SILA
Faculté des Mathématiques et de l'Informatique
Département de l'Informatique

Matière : Intelligence artificielle- 3SI

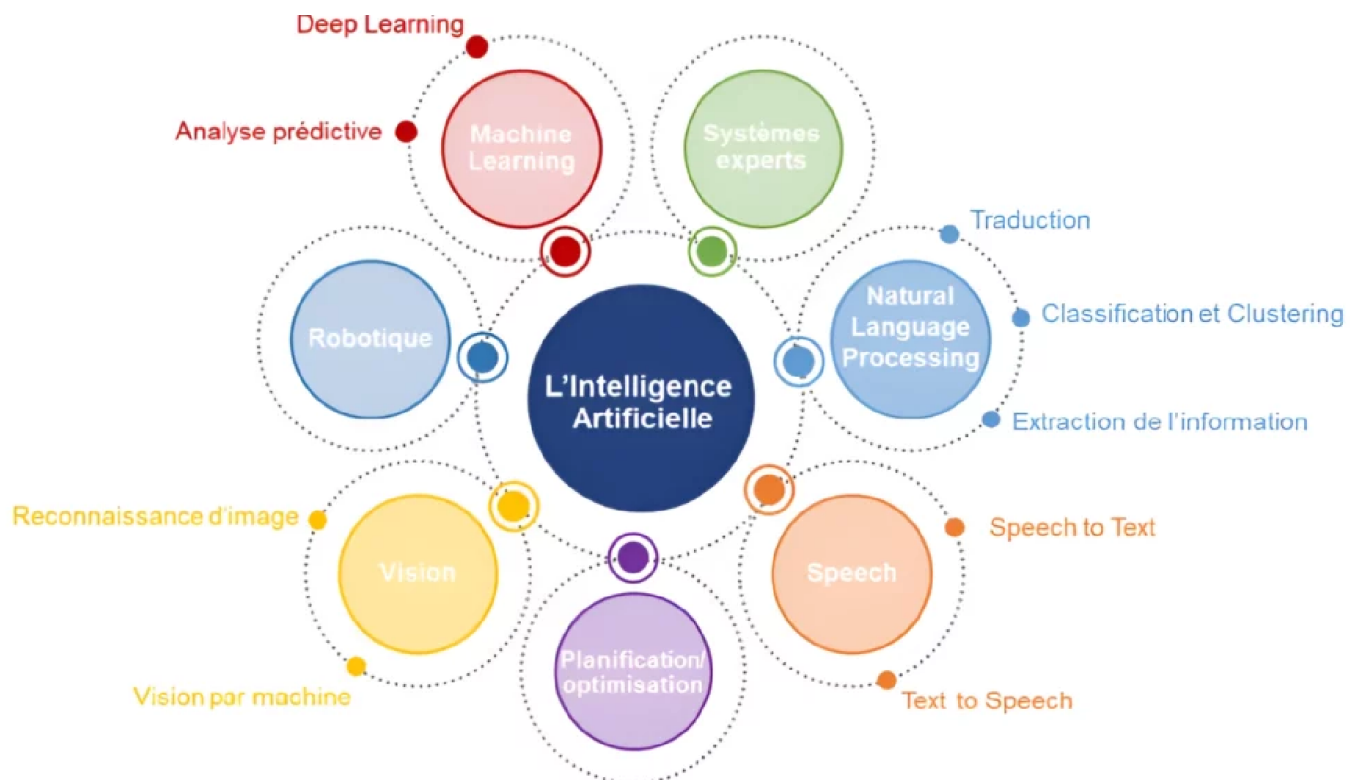
Cours 2 : Les systèmes experts



Connaissance :

système expert : coder les règles...

approche guidée par les données : Apprentissage automatique (machine learning).



Autres domaines :

Data analysis

Data science

Data mining

Big data...

1. Introduction

Les êtres humains ne savent pas toujours **déduire des faits d'autres faits** qui leur sont connus et nécessitent l'aide d'un expert.

Exemple : panne automobile,

>> déterminer l'origine de la panne ??

se tourne vers le garagiste (l'expert).

grâce à ses connaissances >> trouver la panne (et la réparer, normalement).

Domaines : Les médecins, les assureurs, les agents immobiliers...

Système expert définition : programme informatique qui jouera le rôle de l'expert.

Dans certains cas comme la médecine, cet outil pourra devenir une aide au spécia-

liste lui-même car le domaine étudié est très vaste.

- ☐ Un système expert est un système informatique qui utilise les connaissances d'un expert pour obtenir une performance de décision élevée dans un domaine spécifique. ☐
- **MYCIN** : est l'un des premiers systèmes experts développé pour aider les médecins à diagnostiquer les infections sanguines.
- Des tests ont démontré que ce système est aussi précis que des experts humains.

Qu'est-ce qu'un expert?

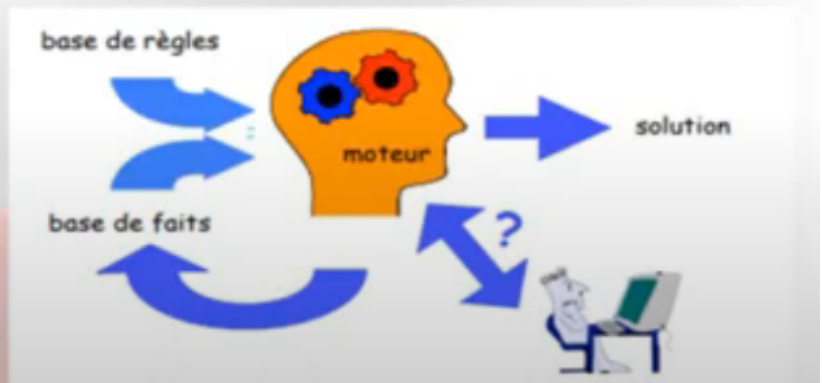
- ☐ Un expert est une personne qui a des connaissances, jugements, expériences et méthodes sur un domaine précis (un expert dans une entreprise n'est pas nécessairement un expert dans une autre entreprise) et capable de les exploiter pour donner des conseils et résoudre des problèmes.
- ☐ L'expert est responsable de donner les connaissances nécessaires pour que le système expert fasse sa tâche car il sait quelle information est importante et comprend les relations entre les différents faits.
- Un expert doit être capable de résoudre un problème et avoir un niveau de performance significativement plus élevé que la moyenne. ☐

Qu'est-ce qu'une expertise?

- □ C'est la connaissance spécifique à une tâche que connaît l'expert. □
- Le niveau d'expertise détermine la performance d'une décision.
- L'expertise est souvent obtenue par la formation, la lecture et l'expérience.

Qu'est-ce qu'un système expert?

- **Objectif** : remplacer les experts humains d'un domaine ou les aider dans leurs expertises
- **Principe** : On dispose de connaissances sur un domaine et d'un moteur d'inférences. À partir d'informations contextuelles factuelles, on infère de nouvelles connaissances sur le problème examiné.
- **Domaines d'application** :
Les systèmes experts peuvent être appliqués dans de nombreux domaines.



- Un système expert utilise la connaissance correspondante à un domaine spécifique dont le but est de fournir une performance comparable à l'expert humain lors de la résolution d'un problème donné.
- Les concepteurs de systèmes experts effectuent l'acquisition des connaissances grâce à un ou plusieurs interviews avec l'expert ou les experts du domaine.
- Le système expert est enrichi avec les connaissances des humains et leurs heuristiques obtenues grâce à l'utilisation de leurs connaissances.

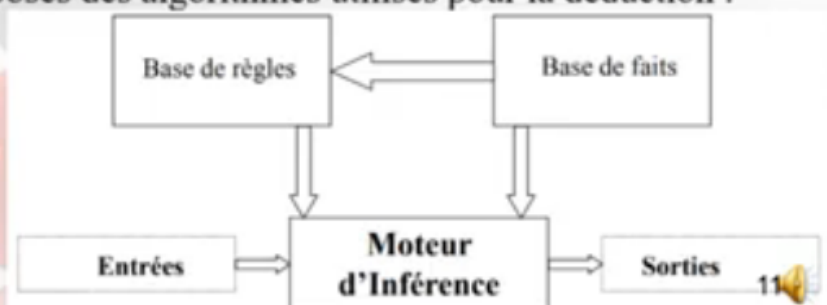
Qu'est-ce qu'un système expert?

- C'est un outil capable de reproduire les mécanismes cognitifs d'un expert, dans un domaine donné.
- Un système expert est composé de :
 - la base de connaissances :
 - base de faits : code la connaissance sur l'étude en cours Son état évolue en cours d'expertise (mémoire de travail)
 - base de règles : code la connaissance sur le domaine.

Règles : SI condition ALORS action

- le moteur d'inférences : composés des algorithmes utilisés pour la déduction :

- chaînage avant
- chaînage arrière
- calcul des faits déductibles
- calcul de question
- chaînage mixte



Bénéfices des systèmes experts

- ❑ **Augmentation de la production et la productivité** ❑: Les systèmes experts fonctionnent plus rapidement que les humains. ❑
- ❑ **Temps de décision réduit** :❑ Avec les recommandations faites par un système expert, un humain peut prendre des décisions plus rapidement. ❑
- ❑ **Amélioration de la qualité** ❑ : Un système expert peut améliorer la qualité en donnant des conseils consistants et en réduisant la fréquence et l'ampleur des erreurs.
- ❑ **Réduction des temps morts** :❑ Plusieurs systèmes experts peuvent être utilisés pour diagnostiquer des défaillances et proposer des réparations.
- ❑ **Amélioration de la qualité des décisions** :❑ Les systèmes experts fonctionnent 24/7. Ils ne sont pas fatigués. La décision ou solution offerte n'est pas influencée par des facteurs humains.
- ❑ **Élimination de machinerie complexe** ❑: Un système expert peut faire les même tâches qu'un humain sans avoir besoin d'instrumentation complexe pour afficher toute sorte d'information.

Limites des systèmes experts

- ☐ Connaissances pas toujours facilement disponibles. ☐
- ☐ Difficulté d'obtenir l'expertise d'un humain ☐
- ☐ Domaine d'utilisation restreint ☐
- ☐ Difficulté d'obtenir des vérifications indépendantes aux solutions proposées ☐
- ☐ Le vocabulaire et termes techniques peut être difficile à convertir à un système expert. ☐
- ☐ Manque de confiance dans les systèmes experts ☐

PROGRAMME

Les logiciels traditionnels, notamment ceux qui sont réalisés dans le domaine de l'éducation, sont composés en général d'un ensemble de données sous forme des connaissances procédurales et d'un ensemble de programmes qui les gère d'une façon spécifique.



SYSTEME EXPERT

Un système expert est un système d'aide à la décision basé sur **un moteur d'inférence** et sur une **base de connaissances**.

Moteur d'inférence : Partie d'un système expert qui effectue **la sélection et l'application des règles** en vue de la résolution d'un problème donné.



"Un système expert est un programme conçu pour simuler le comportement d'un humain qui est un spécialiste ou un expert dans un domaine très restreint"
P. Denning (1986).

Le Système expert est un programme :

- ☐ Incorporant à la fois des connaissances formelles des jugements personnels de l'expert (sous forme d'heuristiques)
- ☐ Expliquant à la fois le raisonnement conduisant aux réponses aux questions le savoir qu'il contient
- ☐ Capable d'incorporer du nouveau savoir dans le savoir existant de manière incrémentale
- ☐ Séparant les connaissances, le contrôle (moyen d'utiliser les connaissances)
- ☐ "Dialoguant" (systèmes évolués) avec l'utilisateur dans un langage proche du sien (langue naturelle, "jargon" du domaine)

38

ETAPES DE CONSTRUCTION D'UN SYSTEME EXPERT

Cinq étapes principales sont généralement identifiables pour la réalisation d'un système expert :

1. Etape d'identification : où l'on veillera à déterminer le rôle attribué à chaque intervenant dans le projet et également à définir clairement et précisément les objectifs.

1. Etape de conceptualisation : qui consiste à expliciter les concepts. les connaissances relevés lors de la phase d'identification. Il s'agit d'élaborer une base conceptuelle sur laquelle se fonderont les futures étapes du développement.

1. Etape de formalisation : qui permet de rassembler toutes les connaissances acquises à l'issue de la conceptualisation. Il faut ensuite choisir un mode de représentation technique en fonction du type de problème pour faire la traduction des connaissances. En général, on utilise le formalisme des règles de production

40

4. Etape d'implantation : qui consiste à traduire les connaissances en utilisant l'outil choisi. Cette implantation donnera le jour au premier prototype du système expert.

5. Etape de test du prototype : où le système est confronté avec des experts humains en utilisant des données réelles. À cette étape, on peut raffiner le système en combinant des ensembles de règles, en ajoutant de nouvelles règles etc. La qualité du travail effectué à ce moment détermine finalement le niveau final de performance du système et son utilité réelle.

Exemple d'illustration : un système expert en polygones.

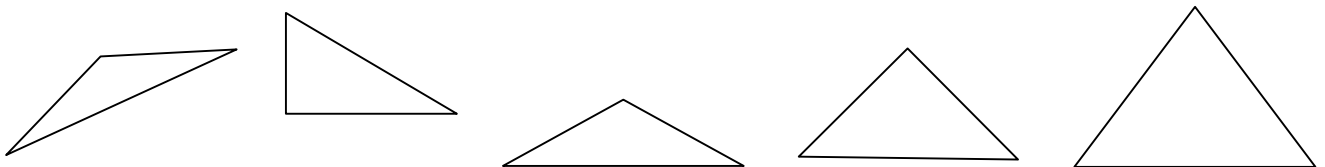
Le but est de déterminer le nom d'un polygone (par exemple un rectangle) en fonction de caractéristiques sur la forme (le nombre de côtés, les angles droits...).

Un polygone est défini comme une forme géométrique possédant au moins trois côtés. L'ordre d'un polygone correspond au nombre de ses côtés.

Triangles

Si l'ordre d'un polygone vaut 3, il possède donc trois côtés et il s'agit d'un triangle.

Celui-ci peut être quelconque, rectangle, isocèle, rectangle isocèle ou équilatéral.



Triangle quelconque : possède trois côtés de tailles différentes et aucun angle droit.	Triangle rectangle : possède trois côtés de tailles différentes et un angle droit.	Triangle isocèle : possède deux côtés de même taille, mais pas d'angle droit.	Triangle rectangle isocèle : cumule les deux côtés égaux du triangle isocèle et l'angle droit du triangle rectangle.	Triangle équilatéral : possède trois côtés de même taille (et ne peut pas posséder d'angle droit).
--	--	---	--	--

3. Mise en place

Comment mettre en place un Système Expert dans un milieu professionnel. Cela se fait en 5 étapes :

1. Etude de faisabilité

*La mise en place d'un SE ne peut se faire que dans le cadre d'un domaine d'expertise dont les connaissances et le **savoir-faire** est **formalisable**. Tous les domaines d'expertise ne **sont pas formalisables**. Le premier travail est donc d'évaluer le **domaine et les risques d'échecs** de la mise en place et de **succès de l'outil** auprès des professionnels et futurs utilisateurs.*

2. Extraction des données

*Extraire de ce dernier toutes ses connaissances et son savoir-faire. Il devra pour cela faire preuve **d'une grande compréhension des informations** qui lui seront transmises (**les experts n'étant pas forcément bon pédagogue**) et d'un certain sens **de la psychologie** pour faire*

parler un expert qui aura tout naturellement le sentiment de se faire très prochainement remplacer par un système informatique.

3. Formalisation

Formaliser les connaissances (les développeurs et autres professionnels techniques de l'informatique afin de commencer à définir la base de connaissance et les règles d'inférence.

4. Design et développement

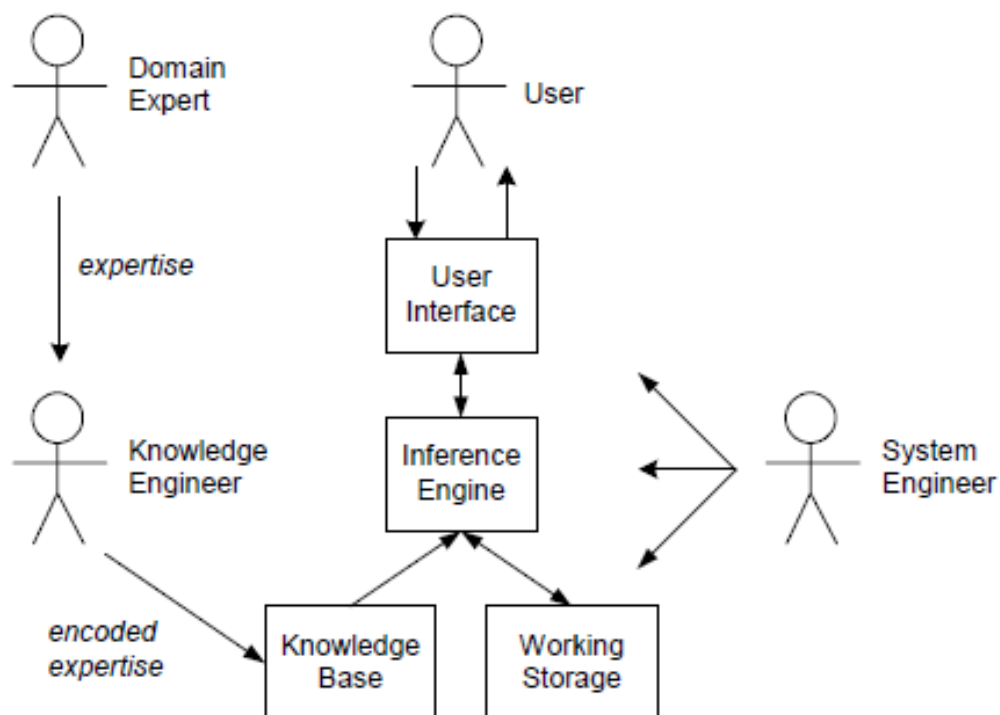
Définir l'architecture technique nécessaire.

5. Tests et optimisations

Une série de tests auprès des experts et d'utilisateurs.

4. Contenu d'un système expert

Les principaux rôles des individus qui interagissent avec le système.



3. Composants d'un système expert

3.1. Expert du domaine : les individus qui sont actuellement des experts résolvant les problèmes que le système est censé résoudre ;

3.2. Ingénieur des connaissances l'individu qui **encode les connaissances** de l'expert dans une forme déclarative qui peut être utilisée par le système expert ;

3.3.Utilisateur - l'individu qui consultera le système pour obtenir des conseils qui auraient été fournis par l'expert du domaine.

4.Composants d'un S E

Un système expert est constitué de différentes parties liées entre elles :

4.1. Base de règles

Un système expert contient un ensemble de règles nommé **base de règles**.

Celles-ci représentent les connaissances de l'expert sur le domaine.

Ces règles sont toujours de la forme :

SI (ensemble de conditions) ALORS nouvelle connaissance

Les conditions d'application d'une règle sont appelées les **prémisses**. Il peut y avoir plusieurs prémisses, elles sont alors reliées par une coordination **ET**, signifiant qu'elles doivent **toutes être vraies** pour que la règle s'applique.

Les nouvelles connaissances sont appelées **conclusions**.

Pour notre système expert sur les formes géométriques, voici les règles concernant les triangles :

SI (ordre vaut 3) ALORS c'est un triangle

SI (triangle ET 1 angle droit) ALORS
c' est un triangle rectangle

SI (triangle ET 2 côtés de même taille
) ALORS c'est un triangle isocèle

SI (triangle rectangle ET triangle iso-
cèle) ALORS c'est un triangle rec-
tangle isocèle

SI (triangle ET côtés tous égaux)
ALORS c'est un triangle équilatéral

En particulier, **les prémisses et conclusions**
peuvent être demandées sous la forme at-
tribut (valeur), par exemple ordre (3) ou
angle Droit (1). La règle du triangle rec-
tangle dans une telle représentation pour-
rait être :

SI (ordre (3) ET angle Droit (l))

ALORS polygone (Triangle Rectangle)

4.2. Base de faits

Les prémisses d'une règle peuvent être de deux types :

- Des connaissances sur le problème fournies par l'utilisateur du système : ce sont les **entrées**.
- Des connaissances **issues** de l'application de règles : ce sont les faits **inférés**.

Ces deux types de connaissances doivent être enregistrés dans une **base de faits** qui contient donc toutes les informations sur le problème, quelle que soit leur origine. Lorsqu'on lance un système expert, la base

ne contient initialement que les connaissances de l'utilisateur (les entrées) et se remplit petit à petit des faits inférés.

Exemple :

une forme d'ordre 4, possédant 4 côtés parallèles, de même taille et 4 angles droits. Ce sont nos connaissances initiales.

À ces faits en entrée va se rajouter le fait que la forme est un quadrilatère (elle est d'ordre 4) et un parallélogramme (un quadrilatère avec 4 côtés parallèles), qu'il s'agit plus précisément d'un losange (un parallélogramme dont les 4 côtés sont de même taille) et qu'il s'agit aussi d'un rectangle (un parallélogramme avec des angles droits). Enfin, on ajoute le fait qu'il

s'agit d'un carré (car il s'agit d'un losange et d'un rectangle).

Généralement, c'est le dernier fait ajouté qui intéresse vraiment l'utilisateur : il s'agit du **but du programme**.

Cependant, on peut aussi concevoir un système expert qui permet de savoir si un fait est **vrai ou non**. Dans ce cas, on regarde si le fait donné est dans la base de faits après application des règles.

Dans l'exemple de notre quadrilatère précédent, au lieu de demander quel est le nom de ce polygone (il s'agit d'un carré), on pourrait demander s'il s'agit d'un losange ou s'il s'agit d'un triangle rectangle. On obtiendrait une réponse affirmative

dans le premier cas (un carré est un losange particulier), et négative dans le deuxième (un carré n'est pas un triangle).

4.3. Moteur d'inférences

Le moteur d'inférences (ou système d'inférences) est le cœur du système expert. Le moteur va permettre de sélectionner et d'appliquer les règles. Cette tâche n'est pas forcément aisée car il peut exister des milliers de règles.

C'est aussi le moteur d'inférences qui va ajouter les nouveaux faits à la base de faits, ou y accéder pour vérifier qu'un fait est déjà connu.

L'avantage des systèmes experts sur de nombreuses autres techniques d'intelli-

gence artificielle est leur capacité à **expliquer le raisonnement suivi** : les moteurs implémentent donc souvent un mécanisme permettant de retrouver toutes les règles utilisées pour arriver à un fait donné. L'utilisateur obtient le **raisonnement en plus du résultat**, ce qui peut être très important dans certains domaines.

Certains langages ont été créés dans le but d'implémenter des moteurs d'inférences. Ils correspondent à la famille des langages de **programmation logique**, dont fait partie **Prolog**.

Interface utilisateur

Le dernier élément d'un système expert est l'interface utilisateur. En effet, il faut que

l'utilisateur puisse simplement **entrer les données** qu'il possède, **soit en une seule fois** avant de lancer le processus, **soit au fur et à mesure des besoins** du moteur d'inférences.

Comme pour tout logiciel, si l'interface **n'est pas agréable à utiliser** ou si elle est **trop complexe**, voire contre-intuitive, le système sera peu ou pas utilisé.

Dans le cas du système expert sur les polygones, il y a peu de chances d'erreurs, car les règles sont basées sur **le nombre de côtés, les angles droits, les côtés parallèles et les tailles des segments**. Pourtant, une interface **demandant l'ordre du**

polygone sans préciser qu'il s'agit **du nombre de côtés** ne serait pas adéquate.

Il est donc important de travailler sur **l'ergonomie** du système expert avec de futurs utilisateurs ou des représentants des utilisateurs, de manière à savoir quels **termes seraient les plus adaptés**, ainsi que la disposition des **écrans**, pour limiter les erreurs.



Deductive

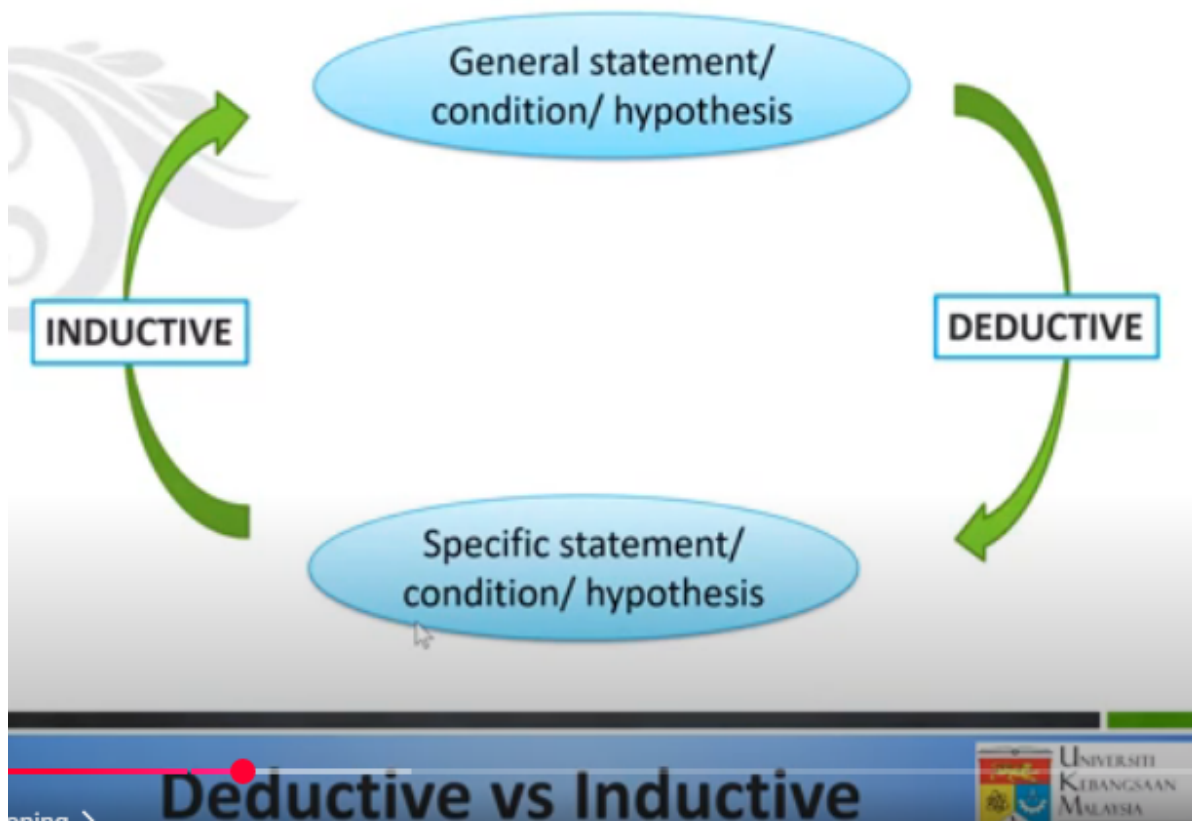
الاستنتاج

- Starts with a general idea/statement
- Then make a specific conclusions

If cond then conclusion

Example

- All insects has 6 legs and 3 body segments
-> ants are insects
- All ManU supporter wears red shirts
->Ali isManU supporter as he wears red shirts
- Even numbers are numbers that can be divided to 2
->28 is even number
- If square has 4 equal sides,
->Then  is a square



Inductive

استقرائي

Généraliser ...découvrir :théorie...règles

Abductive

الاستدلال التخميني

- Abductive is like a guessing game.
- So, to use **A** as a reasoning for conclusion **B**,
 - We need to make sure **B** is enough, but not a necessary for **A**.
 - and **B** is one of the accepted reasons for **A**.
- Thus, it does not promise a correct conclusion.

Example:

- Grasses will be wet when it rains

Now: grasses are wet

Conclusion: was it rained?

- Every Malaysian has a BCG scar
- Now: a corpse was found with BCG scar
- Conclusion: is the corpse a Malaysian?

Analogy

القياس – التشابه – المثل

- Finding a similarities/ difference between 2 or more event

Tiger:

Group : Animal

Num_leg : 4

diet : meat

habitat : SEA and India

Colour : orange and black stripes



shutterstock - 761196088

Analogy

Statistical

احصائي - تكراري

- Frequency or statistical based
- The price for essentials such as oil and electricity can be increased as the usage trend shows that it was never decreasing.*

Types d'inférences

Les moteurs d'inférences peuvent enchaîner les règles de différentes façons : c'est ce que l'on appelle le chaînage. Les deux principaux chaînages sont le **chaînage avant** et le **chaînage arrière**.

Chaînage avant

Principe

Un moteur à chaînage avant est aussi appelé un moteur à inférences dirigé par les **données**. Dans ce mode de chaînage, **on part des données disponibles en base de faits**, et on teste pour chaque règle si elle peut s'appliquer ou non. Si oui, on l'applique et on rajoute la conclusion à la base de faits.

Le moteur explore donc **toutes les possibilités**, jusqu'à trouver le fait recherché ou

jusqu'à ne plus pouvoir appliquer de nouvelles règles.

Ce mode de chaînage est celui proposé par défaut dans des langages de type CLIPS (C Language Integrated Production System), spécialisés dans la construction de systèmes experts.

Exemple les polygones,
supposons que nous partions des faits suivants :

- L'ordre vaut 3.
- Il y a un angle droit.
- Deux côtés sont de même taille.

On commence par appliquer la règle suivante, qui ajoute dans la base de faits que la forme est un triangle :

SI (ordre vaut 3) ALORS c'est un triangle

On peut ensuite en déduire que c'est un triangle rectangle grâce à la règle suivante :

SI (triangle ET 1 angle droit) ALORS c'est un triangle rectangle

De même, on sait que c'est un triangle isocèle :

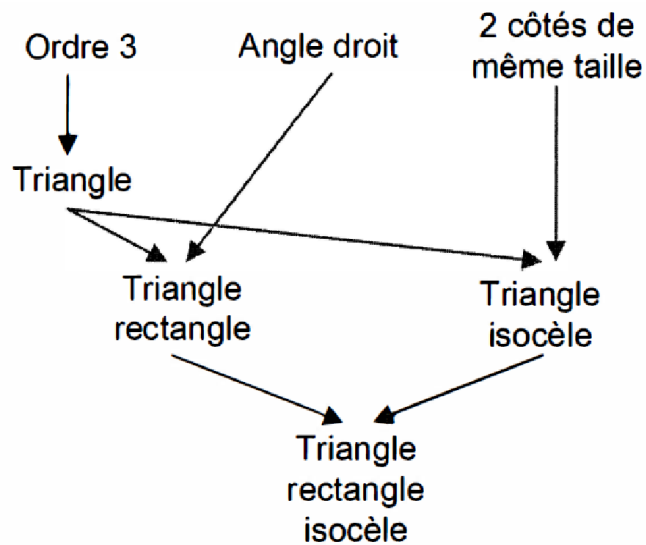
SI (triangle ET 2 côtés de même taille) ALORS c'est un triangle isocèle

Enfin, en sachant qu'il s'agit d'un triangle rectangle et d'un triangle isocèle, on peut appliquer :

SI (triangle rectangle ET triangle isocèle) ALORS c'est un triangle rectangle isocèle

On rajoute donc enfin le fait qu'il s'agit d'un triangle rectangle isocèle. Comme il n'y a plus de règles applicables, le moteur d'inférences s'arrête. L'utilisateur est informé que son polygone est un triangle rectangle isocèle.

On peut résumer la logique du moteur par le schéma suivant :



Chaînage arrière

Principe

Les moteurs d'inférences à chaînage arrière sont aussi dits dirigés par le **but**. On part des faits que l'on **souhaiterait obtenir** et on **cherche une règle** qui pourrait permettre d'obtenir ce fait. On rajoute alors toutes les prémisses de cette règle dans les nouveaux buts à atteindre.

On **réitère**, jusqu'à ce que les **nouveaux buts à atteindre soient présents dans la**

base de faits. Si un fait est absent de la base de faits ou prouvé comme faux, alors on sait que la règle ne peut pas s'appliquer. Ces moteurs ont donc un mécanisme (le backtracking) leur permettant de passer à une nouvelle règle, qui serait un nouveau moyen de prouver le fait. Si plus aucune règle ne peut mener au but recherché, alors celui-ci est considéré comme faux.

Ce mode de chaînage est celui présent par défaut dans le langage Prolog, dédié aux systèmes experts.

Exemple

On reprend l'exemple précédent, à savoir un polygone pour lequel :

- L'ordre vaut 3.

- Il y a un angle droit.
- Deux côtés sont de même taille.

On demande au logiciel si le triangle est rectangle isocèle. C'est notre premier but. Le moteur recherche une règle permettant d'obtenir ce fait, il n'y en a qu'une :

SI (triangle rectangle ET triangle isocèle) ALORS c'est un triangle rectangle isocèle

Le moteur ajoute donc les buts « triangle rectangle » et « triangle isocèle » à sa liste de buts. Il commence par chercher une règle permettant de prouver qu'il s'agit d'un triangle rectangle. Là encore, nous n'avons qu'une règle :

SI (triangle ET un angle droit)

ALORS c'est un triangle rectangle

Il obtient ainsi deux nouveaux buts : s'agit-il d'un triangle et a-t-il un angle droit ? La présence d'un angle droit est déjà en base de faits, ce but est donc atteint. Pour le triangle, il a besoin de la règle suivante :

S I (ordre vaut 3) ALORS c'est un triangle

La base de faits précise que l'ordre a une valeur de 3, la règle est donc prouvée.

Le fait triangle peut ainsi être ajouté à la base de faits et enlevé des buts à atteindre, tout comme le fait « triangle rectangle ». Il ne lui reste alors plus que « triangle isocèle » à prouver.

De la même façon, il cherche une règle possédant ce but :

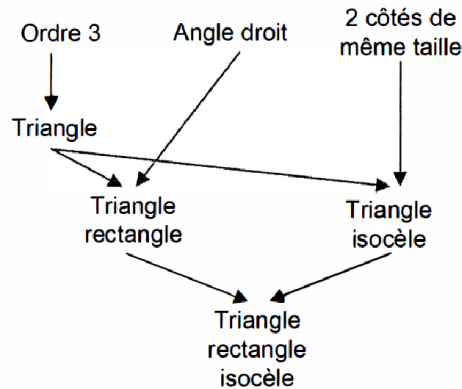
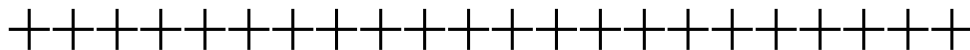
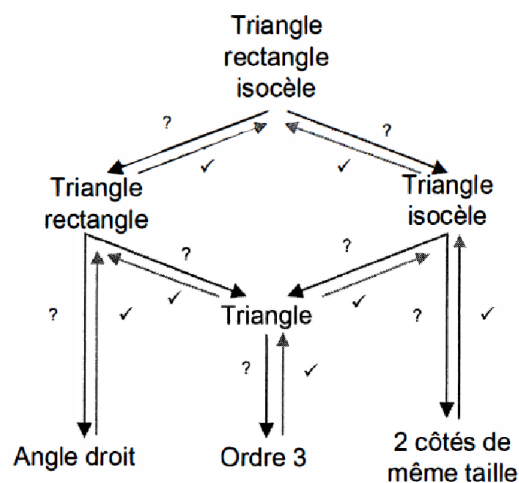
SI (triangle ET 2 côtés de même taille)

ALORS c'est un triangle isocèle

Le fait que la forme soit un triangle est déjà en base de faits (on l'a obtenu juste avant) et la présence de deux côtés de même taille aussi. On rajoute que c'est un triangle isocèle.

Le programme finit par retourner à son but initial, à savoir s'il s'agissait d'un triangle rectangle isocèle . Comme les faits « triangle rectangle » et « triangle isocèle » sont maintenant prouvés, il peut en conclure que oui, la forme est un triangle rectangle isocèle, et l'indiquer à l'utilisateur.

La logique est donc ce coup-ci la suivante : on part du but à atteindre et on essaie de prouver que les prémisses sont vraies.



Chaque mode de chaînage a ses avantages et ses inconvénients :

- Le chaînage avant permet de découvrir en permanence de **nouveaux faits**, mais il risque d'appliquer et de tester de nombreuses règles qui ne concernent pas l'information recherchée par l'utilisateur. Il est donc plus adapté à **l'exploration**.
- Le chaînage arrière permet de se concentrer sur un but précis (ou plusieurs buts), mais il va tester de nombreuses possibilités qui seront finalement démontrées comme fausses. Ainsi, il va essayer de prouver des règles qui ne pourront pas l'être. Sa gestion est aussi plus complexe (car il doit gérer la liste des buts).

Le chaînage mixte.

Un mélange des deux chaînages a alors été proposé : le chaînage mixte. Dans ce nouveau chaînage, on va alterner des périodes en chaînage avant (pour déduire de nouveaux faits de ceux que l'on vient juste de prouver) et des périodes de chaînage arrière (dans lesquelles on cherche de nouveaux buts à prouver).

C'est donc un savant équilibre entre les deux méthodes de recherche, en fonction de règles de recherche. De plus, on peut alterner les phases de recherche en profondeur

aux phases de recherche en largeur selon les buts.

Remarque

Le chaînage mixte est cependant peu utilisé, car il ajoute de la complexité au moteur d'inférences. Il est pourtant beaucoup plus efficace.