UNIVERSITE BADJI MOKHTAR – ANNABA
FACULTE DES SCIENCES DE L'INGENIORAT
Département D'Informatique

Master 1 IATI 2019/2020
Méthodes de l'intelligence artificielle
Chargé Module : **Dr Djebbar Akila**

Chapitre 1 : Raisonnement à base de connaissances

Introduction :

L'un des buts de l'intelligence artificielle (I.A) est de concevoir des systèmes capables de reproduire le comportement de l'être humain dans ses activités de raisonnement. Le terme de système expert disparaîtra sans doute progressivement, au profit du concept plus général de système à bases de connaissances que l'on retrouve dans divers champs d'activité de l'IA.

1. Notions de base

1.1. Le système à base de connaissances (SBC):

Le SBC est fondé sur une séparation entre les connaissances nécessaires pour résoudre un problème et les mécanismes de raisonnement exploitant ces connaissances (appelés selon les cas structures de contrôle, interpréteurs, moteurs d'inférence).

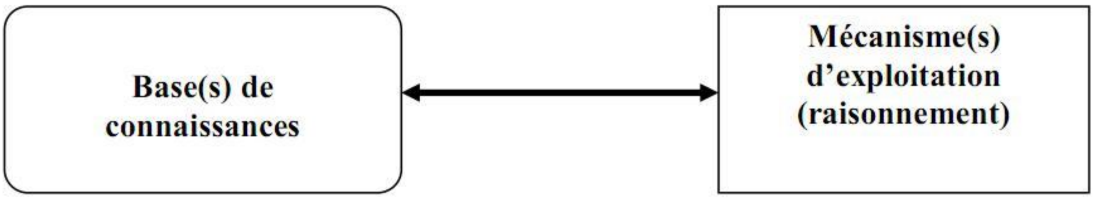


Fig.I.1 : Schéma simplifié d'un système à bases de connaissances

Connaissance : « ensemble des notions et des principes qu'une personne acquiert par l'étude, l'observation ou l'expérience et qu'elle peut intégrer à des habiletés »

Inférence : production d'une connaissance

Avec représentation de la connaissance (déclarative)
Sans représentation de la connaissance (incorporée)

Raisonnement : enchaînement d'inférences avec un objectif

Démontrer de la connaissance => démontrer une capacité à mobiliser des informations pour agir ou produire d'autres capacités à agir (connaissances)

Connaissance = information + mode d'emploi dans un contexte donné
En IA: Connaissance = Information + Sémantique

Pas de classement universel des différents types de connaissances

La représentation des connaissances n'étant pas le but ultime, il faut que les connaissances représentées puissent être manipulées par un mécanisme de raisonnement approprié capable de fournir une solution au problème posé. En effet, représentation et raisonnement sont, d'une manière ou d'une autre, très liés

1.2. Objectif d'un SBC

- Inscrire les connaissances en tant que connaissance (pas seulement en tant qu'information) dans un système :

  Pour « conserver » des savoirs, des savoir-faire et leur sémantique associée

  Disposer d'un « moteur » permettant d'enchaîner des inférences sur ces inscriptions de connaissances :

  Pour « exploiter » les savoirs et savoir-faire ainsi « conservés »

1.3. Rôle d'un SBC

- Inscrit des connaissances issues de l'expertise ou/et de la pratique (on dit que les connaissances sont « représentées » dans un système informatique).
- Est donc spécialisé sur une expertise ou une pratique donnée
- Fonde le « raisonnement » sur des mécanismes d'inférence logique ou analogique
- Intègre une représentation symbolique
- Autorise parfois une certaine prise en compte de l'incertitude
- Les heuristiques sont des connaissances spécifiques au domaine qui guident la recherche de solutions

Est orienté décision, résolution de problème et doit fournir des explications

1.4. SBC à partir de règles

Il s'agit des SBC historiques que l'on appelle « systèmes experts » s'ils sont spécialisés

Les connaissances expertes sont représentées par des règles de la forme Si (prémisses) Alors (conclusions)

Prémisses = conditions de déclenchement de la règle Conclusions = effets du tirage de la règle

Les connaissances sont déclaratives (révisables en principe) L'ensemble des règles forme « la base de connaissances ».

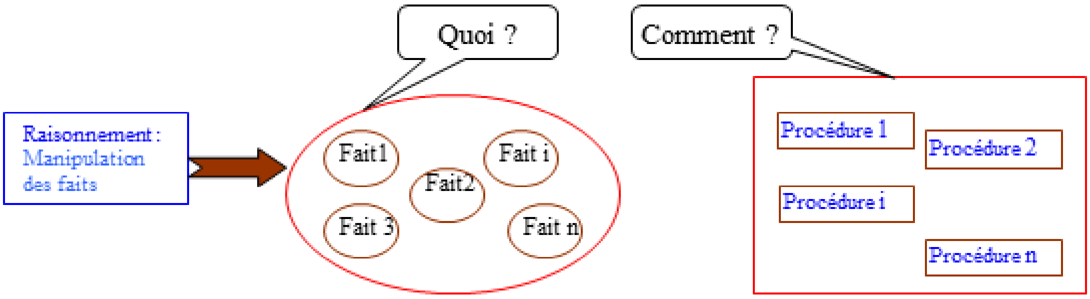
Les faits décrivent ce qui est vrai dans la situation d'exploitation de la base de règles (base de faits)

2. Les types de représentation des connaissances :

Il existe deux grands types de représentation de connaissances, souvent complémentaires.

. **Les représentations déclaratives**, dans lesquelles les connaissances sont décrites indépendamment de leur exploitation ultérieure; elles sont à l'origine de la programmation déclarative d'un système à bases de connaissances ;

.**Les représentations procédurales**, qui contiennent la façon dont les connaissances codées doivent être utilisées.



Représentation déclarative **représentation procédurale**

- Il y a plusieurs raisons pour préférer la manière déclarative à la manière procédurale :
  - ✓ La représentation déclarative peut être modifiée facilement par ajout ou suppression de nouveaux faits.
  - ✓ Une même collection de faits peut être utilisée de différentes façons moyennant quelques simples modifications et ce, selon le problème à résoudre.
  - ✓ Mieux encore, la représentation déclarative est souvent extensible au-delà de ce qui est explicitement représenté et ce, parce qu'elle est 'liée' (malgré tout) à des processus de raisonnement adéquats qui permettent la dérivation d'une connaissance additionnelle.

- La représentation procédurale a, toutefois, la particularité d'être indispensable à toute implémentation sur machine.
- En effet, il y a toujours des connaissances difficiles à exprimer de manière déclarative et même celles, de nature déclarative, doivent à un moment ou un autre être manipulées par une procédure opératoire qui représente le mécanisme de raisonnement utilisé
- Autrement dit, il est nécessaire de spécifier la manière procédurale (le comment) d'utilisation des connaissances déclaratives dans le but de trouver une solution au problème posé.
- Il en résulte qu'en I.A, on a souvent tendance à marier ces deux types de représentation avec, toutefois, une relative prédominance du déclaratif.

3. Les modes de représentation de connaissances :

Le raisonnement sur des connaissances dans un système d'I.A implique leur formalisation selon un certain mode de représentation, qui peut être défini comme un ensemble de méthodes de codage symbolique. Un mode de représentation associe ainsi deux aspects imbriqués, même parfois confondus :

. Une structure de données permettant de représenter la connaissance à coder ;

. Une ou plusieurs méthodes d'exploitation de cette connaissance permettant, par le mécanisme de raisonnement, de produire de nouvelles connaissances.

3.1. Les catégories des connaissances

On distingue 6 catégories des connaissances qui sont :

. **Connaissances de définition**

Exemple : "Un triangle est un polygone ayant exactement trois cotés ".

. **Connaissances évolutives**

Exemple : " Samir est de taille 1 m 52 " (aujourd'hui).

. **Connaissances incertaines**

Exemple : " Mohamed est né en 15 après Omar ".

. **Connaissances vagues**

Exemple : " les jeunes élèves sont turbulents ". (Jeunes =? turbulents = ?).

. **Connaissances typiques**

Exemple : " habituellement, chaque période d'enseignement est consacrée à une seule matière".

. **Connaissances ambiguës**

Exemple : "Avant le conseil de classe, le professeur savait que trois élèves redoubleraient". Est- ce le nombre global, ou bien 3 cas particuliers connus individuellement ? Est-ce trois exactement, ou au moins trois ?

1 Représentation des connaissances

La représentation des connaissances, comme son nom le suggère, a pour but l'étude des formalismes qui permettent la représentation de toutes formes de connaissances. En général, ce sont des langages logiques ou probabilistes. Dans ce chapitre, nous allons nous concentrer sur les langages logiques. Nous commençons donc par un petit rappel des bases de la logique propositionnelle et la logique du premier ordre, puis nous considérons quelques autres logiques.

Rappel : Logique propositionnelle

La logique propositionnelle est une logique très simple qui se trouve à la base de presque toutes les logiques qui sont étudiées aujourd'hui. Les éléments de bases sont des *propositions* (ou variables propositionnelles) qui représentent des énoncés qui peuvent être soit vrais soit faux dans une situation donnée. Nous pourrions par exemple avoir une proposition p qui représente "Marc est un étudiant" ou une proposition q qui représente l'énoncé "Deux est un nombre pair". Des formules complexes peuvent être construites à partir des propositions en utilisant les connectives logiques : $\wedge$ ("et"),

$\vee$ ("ou"), et $\neg$ (negation). Par exemple, nous pourrions avoir les formules $p \wedge q$ ("Marc est un étudiant *et* deux est un nombre pair"), $p \vee q$ ("Marc est un étudiant *ou* deux est un nombre pair"), et $\neg p$ ("Marc *n*'est *pas* un étudiant"). Souvent pour simplicité nous introduisons aussi les connecteurs $\rightarrow$ et $\leftrightarrow$. $p \rightarrow q$ est une abbréviation pour $\neg p \vee q$, et $p \leftrightarrow q$ est juste une façon plus concis d'écrire $(p \wedge q) \vee (\neg p \wedge \neg q)$.

Dans la deuxième partie du chapitre, nous allons avoir besoin des notions de littéral et de clause. Je vous rappelle qu'un *littéral* est une proposition ou la négation d'une proposition. Par exemple, p et $\neg q$ sont des littéraux. Une *clause* est une disjonction de littéraux. Des formules p et $p \vee \neg q \vee r$ sont des exemples de clauses.

Un *modèle* M est une fonction qui donne une valeur de vérité (soit $vrai$, soit $faux$) à chaque proposition du langage. Une proposition p est satisfaite dans un modèle M si elle réçoit la valeur $vrai$, et elle n'est pas satisfaite si elle reçoit la valeur $faux$. Dans le premier cas, nous écrivons $M \models p$ (M est un modèle de p), et dans le deuxième cas, nous écrivons $M \not\models p$ (M n'est pas un modèle de p). La *satisfaction* des formules dans des modèles est entièrement détérminée par les valeurs de vérité des propositions de la formule :

- $M \models \phi \wedge \psi$ si et seulement si $M \models \phi$ et $M \models \psi$
- $M \models \phi \vee \psi$ si et seulement si $M \models \phi$ ou $M \models \psi$ (ou les deux)
- $M \models \neg\phi$ si et seulement si $M \not\models \phi$

Une formule est *valide* si elle est satisfaite par tout modèle, et elle est dite *satisfiable* si elle est satisfaite dans au moins un modèle. Une formule ϕ est une *conséquence logique* de ψ (ce que l'on note $\phi \models \psi$) si chaque modèle qui satisfait ϕ satisfait aussi ψ, ou autrement dit, s'il n'y a pas de modèle de $\phi \wedge \neg\psi$.

La logique propositionnelle est *decidable*, c'est-à-dire qu'il existe des algorithmes pour déterminer si une formule est valide ou non. Le problème de validité est co-NP-complet, et le problème dual de satisfiabilité est NP-complet. Ces résultats de complexité montrent que nous ne pouvons pas ésperer de trouver des algorithme de validité ou de satisfiabilité qui terminent dans un temps raisonnable sur toutes les formules [1]. C'est plutôt une mauvaise nouvelle, mais il faut se rappeller aussi qu'il s'agit de la complexité dans le pire cas. En pratique, la recherche actuelle fournit des algorithmes de plus en plus performants (grace aux heuristiques) et peuvent aujourd'hui traiter des formules de taille importante qui n'auraient pas été résolubles il y a quelques années.

1. Sauf peut-être dans le cas où P=NP, ce qui est considéré comme très improbable.

Rappel : Logique du premier ordre

La logique du premier ordre est une logique très expressive et bien étudiée. Les formules en logique du premier ordre sont formée des éléments suivants :

- les symboles pour constants, e.g. $Marie$, $gareMarseille$
- les variables, e.g. x, y
- les symboles pour fonctions, e.g. $méreDe$
- les symboles pour prédicats, e.g. $humain$, $dans$
- les connectives logiques : $\wedge$, $\neg$, etc.
- la quantificateur universelle $\forall$
- la quantificateur existentielle $\exists$
- la symbole d'égalite $=$

Notez que chaque symbole pour fonction et chaque symbole pour prédicat a un *arité* qui détérmine combien d'arguments elle peut prendre (0, 1, 2, ...), e.g. $mèreDe$ a un arité de 1, $dans$ a un arité de 2. Les *termes* sont des symboles pour constantes, des symboles pour variables, ou bien des symboles pour fonctions appliquées à d'autres termes, e.g. $Marie$, $méreDe(x)$, $méreDe(méreDe(Marie))$. Les *atomes* sont de deux types : soit $P(terme_1, ..., terme_n)$ où P est un symbole pour prédicat avec arité n et les $terme_i$ sont des termes, soit $terme_1 = terme_2$ où $terme_1$ et $terme_2$ sont des termes. Nous pourrions par exemple avoir des atomes $humain(Marie)$, $humain(x)$, $dans(Marie, gareMarseille)$ ou bien $Marie = méreDe(x)$. Les formules sont soit des atomes, e.g. $humain(Marie)$, soit des combinaisons booléenes de formules, e.g. $dans(Marie, gareMarseille) \wedge humain(Marie)$, soit les formules préfixées par $\forall x$ ou $\exists x$ (où x est un variable), e.g. $\exists x.humain(x)$ ou $\forall x.humain(x)$.

Un modèle en logique du premier ordre est composé d'un ensemble non-vide U (appelé l'univers) et d'une fonction d'interpretation I. La fonction I associe à chaque symbole pour constante C un élément de l'univers $I(C) \in U$, à chaque symbole pour fonction F (où l'arité de F est n) une fonction n-aire de U à U, et à chaque symbole pour prédicat P (où l'arité de P est n) un prédicat n-aire sur U. Pour traiter des variables (qui ne sont pas fixés par des modèles), nous faisons appel aux valuations, qui sont des fonctions qui associent à chaque variable un membre de l'univers U. Dans la définition de la satisfaction, nous aurions aussi besoin de la notation $v(x/d)$, qui donne la valuation qui est la même que v sauf pour la variable x, à laquelle on associe l'élément de l'univers d. La satisfaction d'une formule à par rapport un modèle $M =< U, I >$ et une valuation v comme ceci [2] :

- $\models_{M,v} t_1 = t_2$ ssi t_1 et t_2 réfère au même élément de U
- $\models_{M,v} P(t_1, ..., t_n)$ ssi le tuple d'éléments associés à $(t_1, ..., t_n)$ est dans $I(P)$

2. Pour gagner de la place, j'utilise "ssi" comme abbréviation pour "si et seulement si".

- $\models_{M,v} \phi \wedge \psi$ ssi $\models_{M,v} \phi$ et $\models_{M,v} \psi$
- $\models_{M,v} \phi \vee \psi$ ssi $\models_{M,v} \phi$ ou $\models_{M,v} \psi$
- $\models_{M,v} \neg\phi$ ssi $\not\models_{M,v} \phi$
- $\models_{M,v} \forall x.\phi$ ssi pour tout $d \in U$ nous avons $\models_{M,v(x/d)} \phi$
- $\models_{M,v} \exists x.\phi$ ssi il existe $d \in U$ tel que $\models_{M,v(x/d)} \phi$

Une formule est valide si elle est satisfaite par tout modèle et toute valuation.

La logique du premier ordre est beaucoup plus puissante que la logique propositionnelle, et cette expressivité accrue se paie par une difficulté du calcul. En effet, à l'inverse de la logique propositionnelle, la logique du premier ordre n'est pas decidable : il n'existe pas d'algorithmes qui retournent oui si une formule est valide, et non si elle ne l'est pas.

D'autres logiques

La logique propositionnelle et la logique du premier ordre sont de loin les deux logiques les plus étudiées, mais on constate un intérêt grandissant pour d'autres logiques qui sont parfois plus adaptées pour représenter certains types de connaissances. En voici quelques-unes :

Logiques de description Cette famille de logiques se trouve à mi-chemin entre la logique propositionnelle et la logique du premier ordre, offrant ainsi une expressivité beaucoup plus importante que celle de la logique propositionnelle mais avec une complexité du raisonnement moindre que pour la logique du premier ordre (en particulier, les logiques de description sont souvent decidables). Ces logiques possèdent deux types de formules : des axiomes qui décrivent des relations entre des concepts et des assertions qui expriment les caractéristiques des individus et les relations entre individus. Par exemple, nous pourrions avoir des axiomes $Oiseau \sqsubseteq Animal$ (= chaque oiseau est un animal) et $Oiseau \sqsubseteq \forall enfant.Oiseau$ (= les enfants des oiseaux sont aussi des oiseaux) et des assertions $Oiseau(Tweety)$ (=Tweety est un oiseau) et $enfant(Tweety, Paul)$ (=Paul est un enfant de Tweety). Nous pourrions alors inférer que Paul est un oiseau et que Tweety et Paul sont tous les deux des animaux. Les logiques de description sont utilisées dans plusieurs domaines d'application (e.g. la médecine, le traitement du langage naturel, etc.) et sont à la base de la web sémantique (qui est censé être la prochaine incarnation du Web).

Logiques temporelles Ces logiques sont obtenues à partir de la logique propositionnelle en ajoutant un certain nombre de quantificateurs temporels, e.g. “maintenant”, “dorénavant”, “toujours”, qui permettent de parler des propriétés qui sont vraies à différents moments dans le temps. On pourrait par exemple exprimer dans une telle logique le fait “Marc est un étudiant maintenant mais un jour il ne sera plus étudiant” ou “Deux est toujours un nombre pair”. L'une des applications principales des logiques temporelles est la vérification de programmes où ces logiques sont utilisées pour décrire des propriétés qui devraient être vraies lors d'une exécution d'un programme.

Logique floue Dans les logiques classiques, les propositions sont soit vraies soit fausses. Néanmoins, il existe des propriétés qui semblent pouvoir être vraies à un certaine degré. Prenons par exemple la propriété “grand”. Il serait un peu bizarre de séparer les êtres humains entre deux classes, ceux qui sont grands, et ceux qui ne le sont pas – comment choisir une hauteur telle que toutes les personnes avec une taille supérieur à cette hauteur sont grands, et les autres pas ? La logique floue tente de répondre à ce problème en permettant aux propositions de prendre les valeurs entre 0 et 1, où 0 signifie que la proposition n'est pas satisfaite du tout, et 1 signifie la satisfaction complète de la proposition, et les valeurs entre 0 et 1 signifie des niveaux de satisfaction de la proposition. Nous pourrions donc dire que quelqu'un est grand à niveau 0.3 (ce qui veut dire que la personne n'est pas tellement grande mais pas complètement petite) ou à niveau 0.8 (ce qui dit que la personne est plutôt grande). La logique floue a été appliquée avec succès à de nombreux problèmes dans l'industrie.

Logiques de connaissances et/ou croyances L'une des caractéristiques du raisonnement humain est sa faculté de raisonner sur ses propres connaissances/croyances et sur les connaissances/croyances des autres. Un certain nombre de logiques ont été proposées pour formaliser ce type de raisonnement. Dans ces logiques, nous pouvons exprimer les informations de type "je sais que Marie a un enfant, et je crois que c'est une fille, mais je ne suis pas sûr" ou bien "je crois qu'il croit que je sais qui a volé le diamant, mais je ne le sais pas". Le raisonnement sur les connaissances devient un sujet de plus en plus populaire grâce à son utilité dans de nombreux domaines tels que l'IA, la linguistique, l'informatique distribuée, et l'économie.

Logiques non-monotones Les logiques classiques sont appelées *monotones* parce que si $\phi \models \psi$ il est forcément le cas que $\phi \wedge \zeta \models \psi$. Autrement dit, le fait d'ajouter de l'information supplémentaire ne peut pas causer de perte d'informations. Or cette propriété ne semblent pas toujours vérifiée dans les raisonnements humains. Pour prendre un exemple classique, supposons que je vous dit que Tweety est un oiseau, et je vous

demande si Tweety vole. Vous allez probablement me dire que oui car les oiseaux volent. Mais si maintenant j'ajoute le fait que Tweety est un pingouin, vous allez maintenant me dire que Tweety ne vole pas. C'est un exemple d'un raisonnement non-monotone car l'ajout d'une hypothèse (Tweety est un pingouin) fait perdre l'une des conséquences (Tweety vole). Ce type de raisonnement ne satisfait pas donc les lois de la logique classique mais il est rationnel tout de même car il nous permet de raisonner en l'absence d'informations complètes. Les logiques non-monotones essaient de formaliser ce type de raisonnement. Dans la logique des défauts, par exemple, nous pourrions traiter l'exemple de Tweety en utilisant une règle de type "Si X est un oiseau *et* s'il n'est pas connu que X est un pingouin *alors* conclure que X vole". La conclusion que Tweety vole reste valide tant qu'on ne sait pas que Tweety est un pingouin.

5. Différents modes de raisonnements

Trois modes de raisonnements sont utiles. Il convient de bien comprendre les différences entre eux :

5.1. Premier mode de raisonnement : la déduction

Définition : Raisonnement déductif : C'est le type de raisonnement le plus étudié. Il s'agit de dériver les conséquences d'un ensemble d'informations données. Le raisonnement déductif permet de déduire "Marie est un docteur" `a partir des informations "Marie est un docteur ou un professeur" et "Marie n'est pas un professeur", ou bien que "Tweety a des ailes" a` partir de "Tweety est un oiseau" et "Tout oiseau a des ailes". L'une des caractéristiques du raisonnement déductif est que si nous dérivons une conséquence d'un ensemble d'informations qui sont vraies, alors nous sommes sûrs que la conséquence est vraie aussi.

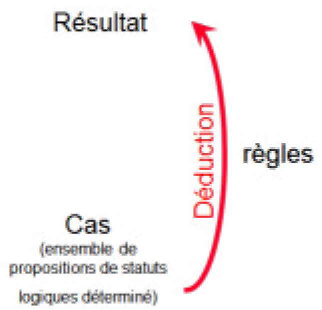
Exemple :

Le premier mode de raisonnement s'appelle la **déduction**. Peirce l'avait illustré avec la première partie de sa démonstration à partir d'un cas et d'une règle.

Le **cas** : un haricot est tiré d'une boîte N° 1.

La **règle,** connue au préalable : tous les haricots de la boîte N° 1 sont blancs.

Le **résultat**, naturellement est : le haricot est blanc. Plus exactement le haricot tiré de la boîte N° 1 - le cas - est blanc parce qu'on lui applique une règle qui énonce que tous les haricots de la boîte N° 1 sont blancs.



Mécanisme de la déduction | *Informations*

5.2. Deuxième mode de raisonnement : l'induction

Définition : Raisonnement inductif : Il s'agit ici de produire des connaissances générales à partir d'observations. Le raisonnement inductif peut donc être vu comme une forme d'apprentissage. Un exemple d'un raisonnement inductif serait de conclure que "Tous les cygnes sont blancs" étant donné que tous les cygnes que l'agent a vu dans sa vie étaient blancs. Il est important de noter qu'`a la différence du raisonnement déductif, les conclusions du raisonnement inductif peuvent s'avérer fausses (il existe des cygnes noirs !). Néanmoins, ce type de raisonnement est tr`es important car il nous permet d'agir dans un environnement donné, lorsque les informations dont on dispose sur celui-ci sont incomplètes.

Exemple :

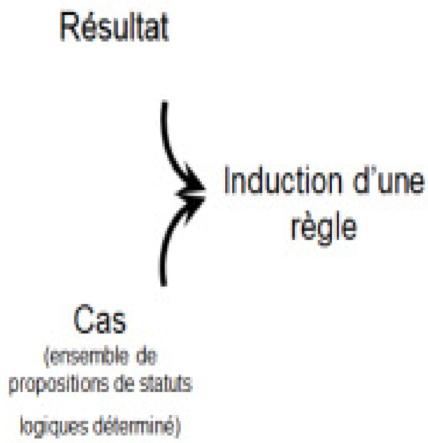
Le deuxième mode de raisonnement est l'**induction**. On part cette fois-ci du cas et du résultat.

Le **cas** : un haricot est tiré de la boîte N° 2.

Le **résultat** : il est rouge.

On peut éventuellement multiplier le nombre de tirages. Le deuxième haricot tiré de la boîte N° 2 et lui aussi rouge, ainsi que le troisième, le quatrième etc.

Puis à partir de cela on peut formuler une **règle** qui est que tous les haricots de la boîte N°2 sont rouges.



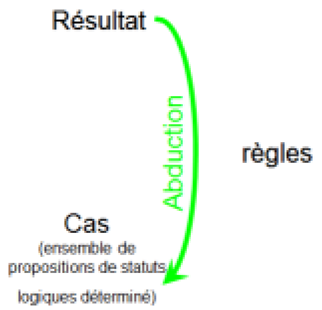
Mécanisme de l'induction | *Informations*

5.3. Troisième mode de raisonnement : l'abduction

Définition : Raisonnement abductif : Le but de l’abduction est d’expliquer des observations. Le raisonnement abductif est utilisé couramment dans la vie de tous les jours, et notamment dans le diagnostic médical : nous cherchons une cause possible (maladie) qui pourrait expliquer les observations (des symptômes de la patiente) en présence de connaissances générales (les connaissances sur la médecine). Un exemple d’un raisonnement abductif pourrait donc être “La patiente a la grippe” étant donné que “La patiente a de la fièvre, de la toux, et mal `a la tête” et “La grippe peut provoquer des fièvres, maux de gorge, maux de tête, fatigue, toux, et des douleurs musculaires”. Notez qu’en général il y a plusieurs explications possibles, e.g. les symptômes de la patiente pourrait aussi être provoqué par une bronchite combinée avec un cancer du cerveau. En général, nous préférons les explications les plus simples, e.g. moins de maladies, ou des plus probables, e.g. des maladies courantes.

Exemple :

Le troisième mode de raisonnement maintenant. Souvenez-vous : un haricot est blanc : d'où provient-il ? Si l'on connaît une règle disant : il y a des haricots blancs dans la boîte N° 1, on peut imaginer un cas qui est que le haricot en question provient bien de la boîte N° 1.



Mécanisme de l'abduction |