Résolution de problèmes

Plan

- **Introduction sur la résolution de problème**
  - Notion de problème
  - Types de problèmes
- **Démonstration automatique**
  - Brève histoire
  - Test de Türing
  - Les objections
  - Exemples fameux
- **Les méta-concepts de la logique**
  - Problème de déduction
  - Approches sémantiques vs formelles
  - Décidabilité

Introduction sur la Résolution de problèmes

Notion de problème

- **Eléments constitutifs d'un problème**
  - Un problème comporte des données, des objectifs ou une tâche à accomplir
  - La recherche des solutions est faite sous contraintes
- **Les approches**
  - En informatique classique : des méthodes déterministes
  - En intelligence artificielle :
    - utilisation de connaissances
    - résolution par la recherche
- **Démonstration automatique**

  On s'intéresse ici aux problèmes qui nécessitent de la part d'un humain qui les résout une *capacité à raisonner*

Notion de problème

- **Problèmes « preuve de théorèmes »**

  Recherche de *solutions exactes* :

  - Formalisation logique
  - Stratégies de résolution générales

- **Problèmes fortement combinatoires**

  Recherche de *"bonnes" solutions* :

  - Utilisation de connaissances
  - Méthodes de recherche qui construisent une solution
  - Application de stratégies spécifiques
  - Efficacité plutôt qu'optimalité

Dans tous les cas, nécessité d'une représentation explicite du problème !

Types de problèmes

Catégories de problèmes

- résolus un temps raisonnable (P)
- résolus en un temps non raisonnable (E, NP)
- sans solution (I)

- *Complexité :*

  La complexité d'un algorithme est la borne supérieure du nombre d'actions qu'il nécessite, rapportée à la taille des données

  La complexité d'un problème est la complexité du meilleur algorithme connu pour le résoudre

- *Calculabilité d'un problème :*

  Un problème est *calculable* s'il existe un algorithme capable de le résoudre *en un temps fini*

Types de problèmes

Problèmes polynomiaux

Un problème est *polynomial* s'il peut être résolu par un algorithme de complexité égale à un polynome de degré constant

- Ils constituent la *classe P*
- Approche algorithmique déterministe
- Exemples : programmation linéaire, tri d'une liste d'entiers, recherche d'un circuit eulérien, d'un plus court chemin dans un graphe, …..

Problèmes exponentiels par nature

Complexité, rédhibitoire, au moins en x^n

- Ils constituent la *classe E*
- Liste des sous-ensembles d'un ensemble

Types de problèmes

Problèmes NP-complets

Caractérisation de la classe NP-complet, à l'existence hypothétique :

- Ces problèmes appartiennent à la classe NP (meilleur algorithme connu en x^n)
- Tous les problèmes NP s'y ramènent

« L'IA est l'étude des techniques utilisées pour résoudre des problèmes NP-complets en un temps polynomial »

Exemples : problème du voyageur de commerce, détermination d'un circuit hamil-tonien, diagnostic, etc ………

Types de problèmes

Problèmes indécidables

Problèmes posés sous forme de question et pour lesquels il n'existe pas d'algorithme capable d'y répondre *en un temps fini*

- Ces problèmes appartiennent à la *classe* I
- Forme particulière de non-calculabilité

Exemples de problèmes indécidales :

- Vérification de l'arrêt d'un programme, …..
- Théorématicité en Calcul des Prédicats
- Equations diophantiennes
- Hypothèse du continu….

Problèmes décidables :

- Théorème de Fermat
- Conjecture de Goldbach ?

Démonstration automatique

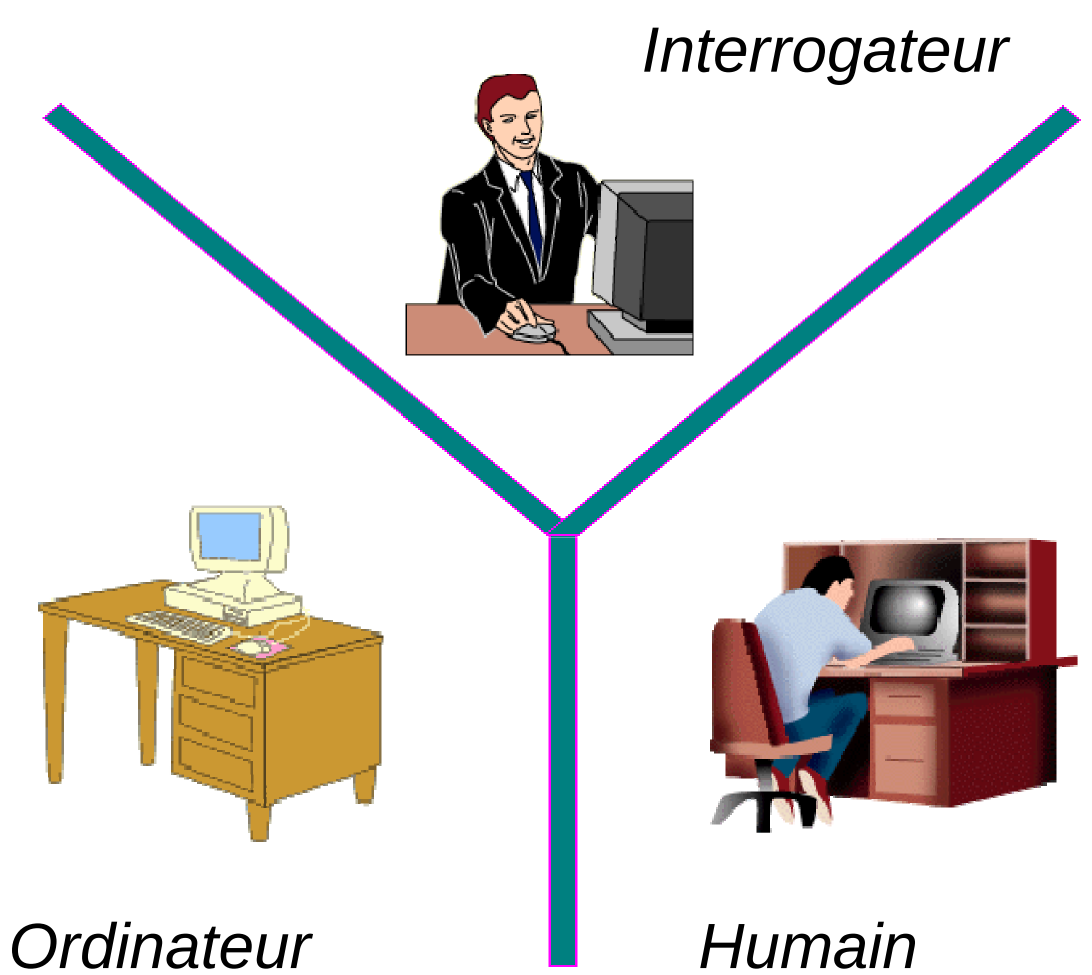
Brève histoire

- **Les précurseurs**
  - Aristote, Descartes, Boole….
  - Herbrandt (1930), Türing, Von Neumann.....
- **Les premiers développements**
  - Logic Theorist (1956)
  - General Problem Solver (1958)
  - Geometry Theorem Proving Machine (1959)
  - Principe de Résolution (1965)
  - Eliza (1966)
  - Les micro-mondes (Shrdlu 1972)

Brève histoire

- **La programmation logique**
  - Prolog (1972)
  - Prolog d'Edimbourg (1975)
  - Prolog + contraintes (1982)
- **Le traitement des connaissances**
  - Le calcul symbolique
    - Macsyma (1971)
  - Les systèmes experts
    - Dendral (1969), Mycin (1980)
  - Les systèmes de planification
  - L'ingénierie des connaissances

Le test de Türing



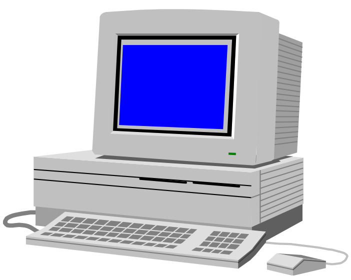
Test réussi si l'interrogateur ne peut distinguer l'homme de l'ordinateur

Les objections

Envisagées par A. Türing (1950)

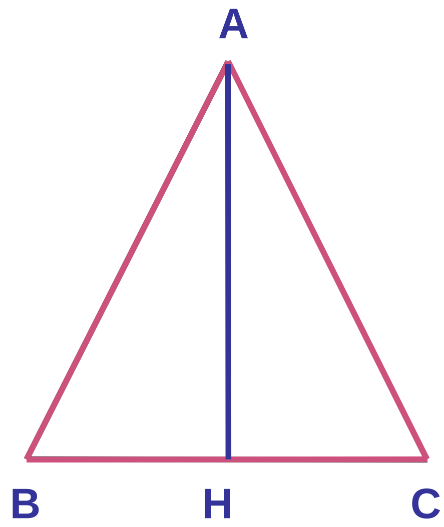
- objection théologique
- objection de l'autruche
- objection du caractère non formalisable du comportement
- objection mathématique....
- et celle de Lady Lovelace :

  « La Machine Analytique n'a pas la prétention de créer quoi que ce soit. Elle peut faire tout ce que nous savons lui ordonner de faire »



Exemples fameux

Le triangle isocèle



Hypothèse

$AB = AC$

Conclusion

$\widehat{ABC} = \widehat{ACB}$

Démonstration classique

$AB = AC$, $AH = AH$ et $\widehat{AHB} = \widehat{AHC}$

donc Tr ABH = Tr ACH

d'où $\widehat{ABC} = \widehat{ACB}$

Démonstration automatique

$AC = AB$, $AB = AC$ et $BC = CB$

donc Tr ABC = Tr ACB

d'où $\widehat{ABC} = \widehat{ACB}$

Les méta-concepts propres à la logique

Problème de déduction

Le problème de déduction (Prd)

Situation envisagée : étant donné un ensemble d'assertions, il s'agit de statuer sur la validité d'une conjecture

Problème : soit *E* un ensemble de fbfs de L0 (ou L1) (hypothèses). Soit une fbf *A* de L0 (resp. L1) (conclusion). Supposons que toutes les fbfs de *E* soient vraies. Alors, *A* est-elle vraie ?

Reformulation logique de Prd :

Soit I une interprétation commune à *E* et à *A*, modèle de *E*. I est-elle alors un modèle de *A* ? Ou encore : *A* est-elle une conséquence logique de *E* ?

Problème de déduction

Approche sémantique en L0

– Dans l'approche sémantique, Prd est abordé en termes de vérité ou de fausseté

– Il se ramène au célébrissime problème de la satisfiabilité d'un ensemble de fbfs de L0

– Faisabilité technique :

  - le nombre d'interprétations est fini
  - des algorithmes : Quine, réduction, Davis et Putman,….
  - « Le raisonnement est un calcul » (Boole)

– *Exemple d'approche sémantique :*

Soit $E = \{ p \vee q \; ; \; \neg q \}$. Soit $A = p$. A est-elle conséquence logique de E ?

Problème de déduction

Approche syntaxique en L0

L'approche syntaxique (axiomatique) consiste à déduire *A* à partir de *E* (et d'axiomes) par un enchaînement d'inférences

- Voir « systèmes formels »
- Utilisation de règles d'inférence :
  - *modus ponens*

    $\{ p \Rightarrow q \,;\, p \} \vdash q$
  - *modus tollens*

    $\{ p \Rightarrow q \,;\, \neg q \} \vdash \neg p$
- *Exemple d'approche syntaxique :*
  Soit $E = \{p \vee q \,;\, \neg q\}$. Soit $A = p$. Peut-on déduire formellement *A* de *E* ?

Problème de déduction

Approche sémantique en L1

- Difficulté inhérente à l'approche sémantique: le nombre d'interprétations est ici *infini* !
- Une méthode qui suppose l'inspection exhaustive des interprétations n'est donc pas envisageable pour Prd
- Selon le théorème d'Herbrandt (1931), on peut ramener le problème Prd à la satisfiabilité d'un ensemble de fbfs propositionnelles. Mais cet ensemble peut lui-même être infini !
- *Exemple d'approche sémantique :* Soit $E = \{ \forall x\ (P(x) \Rightarrow Q(x))\ ;\ P(a)\}$. Soit $A = Q(a)$. A est-elle conséquence logique de E ?

LG1

Problème de déduction

Approche syntaxique en L1

- Principe identique à celui vu pour L0
- Exemples de règles d'inférence :
  - *spécialisation universelle*

    $\forall x\ A(x) \quad |\!-\quad A(c)$

    où c est une constante quelconque
  - *généralisation*

    $A(x) \quad |\!-\quad \forall x\ A(x)$

    où x est libre dans A
- *Exemple d'approche syntaxique :*

  Soit $E = \{\forall x\ (P(x) \Rightarrow Q(x))\ ;\ P(a)\}$. Peut-on déduire formellement $Q(a)$ de E ?

Diapositive 21

LG1 Le Grisly; 21/12/2002

Propriété d'adéquation

Notion d'adéquation

L0 et L1 étant judicieusement axiomatisés : « toute fbf *A* déductible d'un ensemble fini de fbfs *E* est conséquence logique de *E* ». Soit :

Si $E \vdash A$ alors $E \models A$

Ou encore le méta-théorème suivant :

Les systèmes L0 et L1 sont adéquats

Commentaires :

– On dit aussi : correct, ou sain
– Ces méta-théorèmes nous disent donc, en résumé lapidaire : « tout ce qui est déductible est vrai »

Propriété de complétude

Notion de complétude

Les systèmes L0 et L1 étant judicieusement axiomatisés : « toute fbf *A* conséquence logique d'un ensemble fini de fbfs *E* est une fbf déductible de *E* ». Soit :

Si $E \models A$ alors $E \vdash A$

Ou encore le méta-théorème suivant :

Les systèmes L0 et L1 sont complets

Commentaires :

– Il s'agit ici de *complétude sémantique*

– Ces méta-théorèmes d'une extrême importance, dûs à Post (1921) pour L0 et à Gödel (1930) pour L1, nous disent que « tout ce qui est vrai est déductible »

Sémantique / Syntaxique

Théorème de complétude généralisé

Le résultat est tout à fait remarquable : il y a équivalence entre validité et déductibilité !!!

$E \models A$ si et seulement si $E \vdash A$

– *Conséquence* : pour traiter Prd, les approches sémantique et syntaxique sont toutes deux possibles

- pour L0 : plutôt l'approche sémantique
- pour L1 : plutôt l'approche syntaxique

– *Attention* : cette équivalence n'est pas nécessairement vérifiée, en particulier pour certains systèmes plus puissants que L1

Décidabilité

Notion de décidabilité en logique

Existe-t'il un algorithme capable de décider, *en un temps fini*, si une fbf *A* est conséquence logique d'un ensemble de fbfs *E* ?

– *En logique des propositions* **:**

La réponse est positive, compte-tenu de la possibilité d'inspecter exhaustivement les diverses interprétations

La logique des propositions est décidable

– *En logique des prédicats* **:**

La réponse est négative (théorème de Church)

La logique des prédicats est indécidable

Décidabilité

Semi-décidabilité de L1

« Il existe un algorithme qui, en cas d'application à une fbf *A* conséquence logique de *E*, vérifie cette propriété en un temps fini (non borné) ». Soit le théorème suivant :

L1 est semi-décidable

Conséquence pour L1 **:**

Au cas où *A* n'est pas conséquence logique de *E*, cet algorithme peut ne jamais terminer, ce qui interdit alors de connaître le statut de *A* relativement à *E*

Les raisons **:**

Une moindre connaissance du monde des non-théorèmes que de celui des théorèmes