Enjeux Environnementaux et Sociétaux du Numérique :

Christine Solnon

INSA de Lyon - 4IF

2023 / 2024

Présentation de l'EC EESN

6 cours en amphi :

- Comment la pensée algorithmique remodèle le monde (C. Solnon)
- Perspective socio-historique sur le matériel info. (F. De Dinechin)
- Le numérique dans les limites planétaires (M.-P. Escudier et L. Morel)
- Législation, licences et RGPD (F. Biennier)
- Ecrans et santé humaine (G. Beslon)
- Genre et informatique (C. Abry-Durand et C. Solnon)

4 TP de 4h :

- Prospective
- Modélisation du monde
- Dark Patterns
- Vie privée

Résolution de problèmes concrets :

- Estimer la surface d'un terrain, Répartir équitablement un ensemble d'objets, ...
- Numération (en base 60)
  $\rightsquigarrow$ langage avec 2 symboles
- Utilisation de tables pour calculer

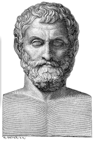
Picture by M0tty (table d'inverses)

Quelques éléments historiques...

Introduction de l’abstraction et du raisonnement

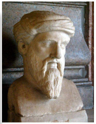
Introduction d’objets mathématiques abstraits :

Droite, cercle, triangle, ...

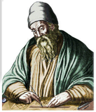


Démonstration de théorèmes :

Thalès (500 BC.), Pythagore (500 BC), Euclide (300 BC), ...



- Théorèmes démontrés en appliquant des règles de déduction à des axiomes (ou des théorèmes déjà démontrés)
- Raisonnements sur des ensembles infinis



Conception d’algorithmes :

Euclide (300 BC), Eratosthène (200 BC), ...

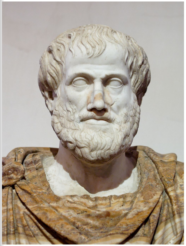
Quelques éléments historiques...

Introduction de la logique

Organon (Aristote, 300 BC) :

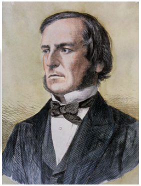
- Syllogismes : Prémisses $\Rightarrow$ Conclusion
- Exemple :
  - Tous les hommes sont mortels
  - Socrate est un homme

  Conclusion : Socrate est mortel
- Logique utilisée en **philosophie**, et non en mathématiques



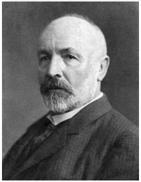
Quelques éléments historiques...

Formalisation de la logique



Boole, 1847 : Algèbre de Boole

Variables booléennes et connecteurs logiques ($\wedge, \vee, \neg$)
$\rightsquigarrow$ Satisfiabilité d'une formule propositionnelle



Cantor, 1874 : Théorie des ensembles

Notions de cardinalité d'un ensemble et d'équipotence entre ensembles $\rightsquigarrow$ Infinité de tailles d'ensembles infinis

Frege, 1879 : Logique des prédicats axiomatiques

- Introduction des quantificateurs ($\forall, \exists$)
- Séparer l'intuition (axiomes) de l'inférence logique (règles de déduction)

$\rightsquigarrow$ **Logicisme** : Mathématiques réductibles à la logique !?

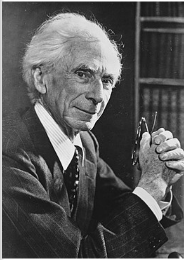
Quelques éléments historiques...

Notions de cohérence et complétude

1902 : Russel revisite le paradoxe d'Épiménide le Crétois

- Soit $\mathcal{E}$ l'ensemble de tous les ensembles
  - Certains ensembles se contiennent eux-mêmes
    Par ex. : $\{e \in \mathcal{E} : \#e \geq 1\}$
  - D'autres ne se contiennent pas
    Par ex : $\{e \in \mathcal{E} : \#e \leq 4\}$, ou $\{i \in \mathbb{N} : \exists j \in \mathbb{N}, i = 2j\}$
- Soit x l'ensemble des ensembles ne se contenant pas
  $x = \{e \in \mathcal{E} : e \notin e\}$
- $x \in x$ ou $x \notin x$?

cf Russel, 1906 : Les paradoxes de la logique



Propriétés désirables pour un système formel

- **Cohérence** : La négation d'un théorème ne peut pas être un théorème
  $\rightsquigarrow$ On ne peut pas prouver des théorèmes contradictoires
- **Complétude** : Toute formule peut être soit prouvée vraie, soit réfutée

Programme de Hilbert, 1920 :

Définir un système formel cohérent et complet pour les mathématiques $\rightsquigarrow$ Méta-mathématiques

> *We are not speaking here of arbitrariness in any sense. Mathematics is not like a game whose tasks are determined by arbitrarily stipulated rules. Rather, it is a conceptual system possessing internal necessity that can only be so and by no means otherwise.*

cf Hilbert's Program

Hilbert & Ackermann, 1928 : Problème de la décision (*Entscheidungsproblem*)

Remplacer le raisonnement par un algorithme :

- Entrée : une formule x (syntaxiquement correcte)
- Sortie : vrai si x est démontrable ; faux si x est réfutable

$\rightsquigarrow$ Si cet algorithme existe alors le système est cohérent et complet !

Presburger, 1929 : Arithmétique des nombres entiers avec addition

Version simplifiée de l'arithmétique de Presburger :

- Soit N l'ensemble défini récursivement par :
  - $0 \in N$
  - $\forall x \in N, s(x) \in N$
- Axiome : $\forall x \in N, p(0, x, x)$
- Règle : $\forall x, y, z \in N, p(x, y, z) \Rightarrow p(s(x), y, s(z))$
- Exemples de théorèmes : $p(0, 0, 0)$, $p(0, s(0), s(0))$, $p(s(0), s(s(0)), s(s(s(0)))$, ...

Et aussi :

- Skolem, 1930 : Arithmétique des nombres entiers avec multiplication
- Tarski, 1930 : Arithmétique des nombres réels avec addition et multiplication

Quelques éléments historiques...

Incomplétude et indécidabilité

Either mathematics is too big for the human mind, or the human mind is more than a machine

Gödel, 1931 : Théorèmes d'incomplétude

- Un système formel cohérent pour l'arithmétique des nombres entiers ne peut être complet

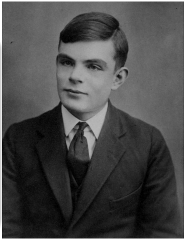
  Peut-on compléter un système incomplet en ajoutant des axiomes ?

  $\rightsquigarrow$ Lire Delahaye, 2009 : Presque tout est indécidable !

- Un système formel ne peut démontrer sa propre cohérence

Turing, 1936 : Le problème de l'arrêt est indécidable

- Entrées :
  - un programme P ayant un paramètre en entrée
  - une valeur V pour ce paramètre en entrée
- Question: L'exécution de $P(V)$ termine-t-elle ?



Preuve de l'indécidabilité du problème de l'arrêt

Supposons qu'il existe le programme *halt(P,V)* tel que :

- Données en entrée de *halt* :
  - un programme P ayant un paramètre en entrée
  - une valeur V pour ce paramètre
- Postrelation : *halt(P,V)* retourne vrai si $P(V)$ termine, et faux sinon

Définissons le programme *diag(X)* suivant :

- Si $halt(X, X)$ = vrai alors boucler à l'infini
- Sinon retourner vrai

Exécutons *diag* sur lui-même :

diag(diag) termine $\Rightarrow$ *halt(diag,diag)* = vrai $\Rightarrow$ *diag(diag)* ne termine pas
$\leadsto$ **Contradiction !**

Formalismes Turing-complets

Calculabilité d'une fonction f:

Il existe un algorithme pour calculer $f(x)$ à partir de x en un nb fini d'étapes
$\rightsquigarrow$ Quel langage/formalisme considérer pour exprimer cet algorithme ?

Deux formalismes introduits en 1936 :

- Church, 1936 : λ-calcul
- Turing, 1936 : Machine de Turing

$\rightsquigarrow$ Même pouvoir d'expression $\Rightarrow$ Formalismes **Turing-complets**

Exemples d'autres formalismes Turing-complets :

- Jeu de la vie (cf Gardner, 1970 : Conway's game "life")
  Rendell, 2011 : A universal Turing machine in Conway's Game of Life
  Zucconi, 2020 : Let's build a computer in Conway's game of life
- F. Fages, G. Le Guludec, 2017 : Réseaux d'interactions biochimiques

Le concept de calculabilité est-il absolu ?

Thèse de Church (*aka* thèse de Church-Turing)

A-t-on atteint un concept absolu de calcul ?

$\rightsquigarrow$ La calculabilité est-elle indépendante d'un formalisme ?

Forme psychologique de la thèse de Church :

Tout algorithme qu'un **être humain** est capable d'exécuter pour résoudre un certain problème peut être exprimé par un ensemble de règles de calcul

Forme physique de la thèse de Church :

Tout algorithme qu'un **système physique** est capable d'exécuter pour résoudre un certain pb peut être exprimé par un ensemble de règles de calcul

Les mathématiques seules ne peuvent prouver/réfuter ces thèses...

- Besoin de connaissances en psychologie pour la forme psychologique
- Besoin de connaissances en physique pour la forme physique

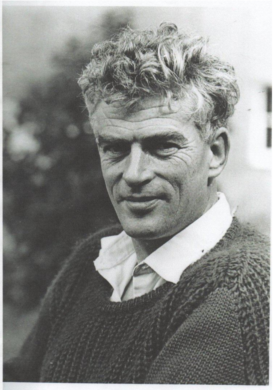
Les notions d'états et de transitions sont-elles les mêmes pour un humain, un système physique, et une machine de Turing ?

Le concept de calculabilité est-il absolu ?

Gandy, 1980 : Démonstration de la forme physique de la thèse de Church

Hypothèses :

- Espace physique = espace géométrique à 3 dimensions
- Finitude de la densité de l'information : Un système physique de taille finie ne peut être que dans un nombre fini d'états différents
- Finitude de la vitesse de transmission de l'information : L'état d'un système ne peut influencer l'état d'un autre système qu'après un délai proportionnel à la distance entre ces deux systèmes



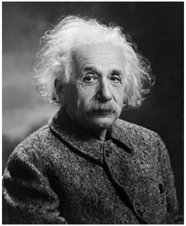
Lire aussi Dershowitz and Gurevich, 2008 : A Natural Axiomatization of Computability and Proof of Church's Thesis

La nature est-elle mathématisable ?

Points de vue de Galilée et Einstein

Galilée, 1623

(...) le livre de l'Univers (...) est écrit dans la langue mathématique et ses caractères sont des triangles, des cercles et autres figures géométriques, sans le moyen desquels il est humainement impossible d'en comprendre un mot. Sans eux, c'est une errance vaine dans un labyrinthe obscur.



Einstein, 1934

Ce qui est incompréhensible, c'est que le monde soit compréhensible

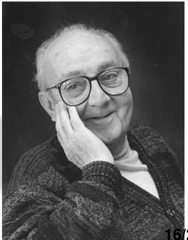
Pourquoi existe-t-il des modèles mathématiques pour décrire des phénomènes naturels ?

Quelques éléments de réponse à discuter...

- Parce que les concepts mathématiques sont des abstractions construites à partir d'observations de la nature ?
- Parce qu'on se focalise sur les phénomènes naturels qu'on arrive à modéliser et on ignore les autres ?
- Parce que pour décrire un phénomène naturel, on fait abstraction de tout ce qui nous gène pour le modéliser ?

Lire aussi Hamming, 1980: The Unreasonable Effectiveness of Mathematics

Box, 1919-2013 : Tous les modèles sont faux, certains sont utiles



De l'importance du choix des axiomes

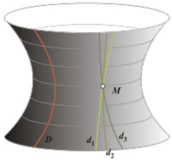
5ème axiome (postulat) de la géométrie euclidienne (300 BC) :

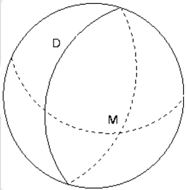
- Par un point distinct d'une droite passe une et une seule droite parallèle

Cet axiome est il une conséquence des 4 premiers ?

Géométries non euclidiennes :

- Lobachevski (1829) et Bolyai (1832) :
  $\rightsquigarrow$ (...) une infinité de droites parallèles
  $\rightsquigarrow$ Géométrie hyperbolique
- Riemann (1854) :
  $\rightsquigarrow$ (...) aucune droite parallèle
  $\rightsquigarrow$ Géométrie sphérique





Les axiomes définissent les concepts étudiés ...

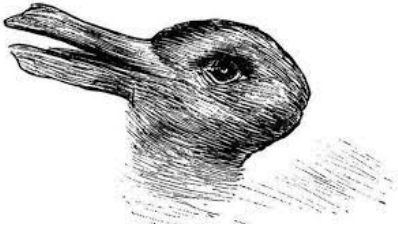
- Géométrie sphérique : droite = grand cercle
- Géométrie hyperbolique : droite = géodésique

La trahison des images (... et des modèles ...)

René Magritte, 1929

Un modèle représente une réalité
Il n'est pas cette réalité

Joseph Jastrow, 1892

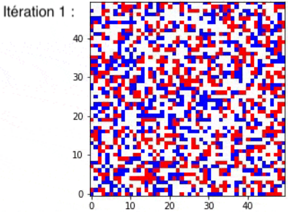


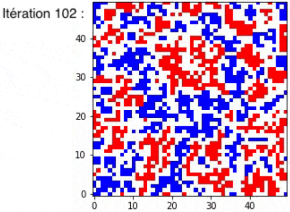
L'abstraction introduit de l'ambiguïté

Exemple 1 : Modèle de ségrégation de Schelling, 1971

Modèle individu-centré et synchronisé sur une grille :

- Chaque cellule = un logement occupé (rouge ou bleu) ou vide (blanc)
- Un individu déménage si moins de 33% de voisins de sa couleur





- Conclusion de Schelling : une forte ségrégation spatiale peut résulter de préférences individuelles ne visant pas une telle ségrégation
- Et vous, quelles sont vos conclusions ?

Lire aussi :

- Forsé, Parodi, 2019 : Retour critique sur le modèle de ségrégation urbaine de Schelling
- Jensen, 2018 : Pourquoi la société ne se laisse pas mettre en équations

Exemple 2 : Optimisation d’une tournée de livraison

Critique du modèle utilisé pour le PLD Agile

- Quelles abstractions ont été faites ?
- Pourquoi ?
- Aurait-on pu faire d’autres choix ?
  $\rightsquigarrow$ Avantages et inconvénients ?

Quel est le “meilleur” modèle selon le point de vue

- du programmeur ?
- de l’utilisateur ?
- de l’opérateur du service ?
- de la société ?

Peut-on imaginer des effets rebonds qui changent la donne ?

Exemple 3 : World 3 (Meadows et al, 1972 : The Limits to growth)

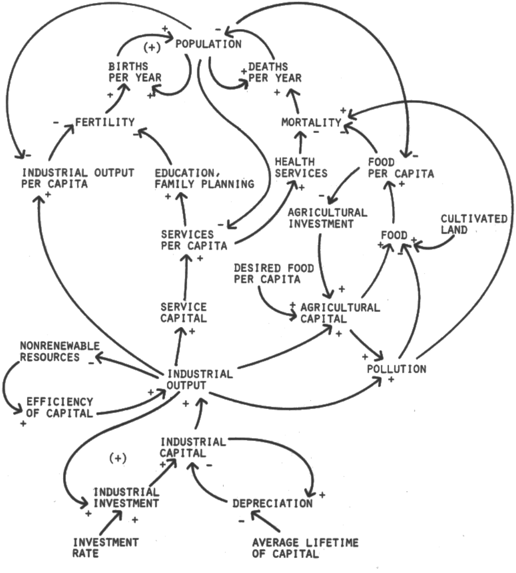
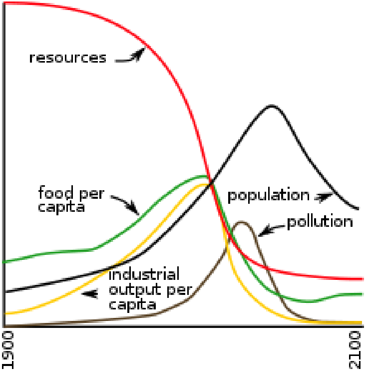


Figure 1-3 Causal-loop diagram of several important feedback loops in World3

Modèle macro stock-flux

- Objectifs ?
- Abstractions ?
- Critiques ?



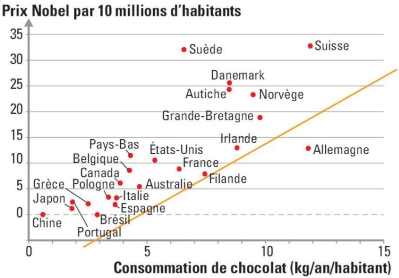
Exemple 4 : Modèles appris à partir de données

Deux niveaux d'abstraction :

- Description d'objets/personnes réels par des vecteurs numériques
- Apprentissage d'un modèle à partir de ces représentations numériques

Quelques questions à débattre...

- Peut-on réduire une personne à des données/traces numériques ?
- Peut-on distinguer causalité et corrélation ?
- Comment vérifier l'absence de biais dans les données ?
- ...



Conclusion ?

Quelques messages à emporter :

- Il existe des limites théoriques à la calculabilité
  $\rightsquigarrow$ Le raisonnement ne peut être totalement remplacé par des algos
- Pour mettre le monde en équations, on introduit des abstractions
  - Modèles génériques pouvant s'appliquer à une infinité d'objets
  - Points de vue subjectifs
- Le choix du langage de modélisation a un impact fort
  $\rightsquigarrow$ Ce qui n'est pas exprimable risque fort d'être ignoré
- L'informatique change notre façon de voir le monde
  $\rightsquigarrow$ Ne pas oublier que le jumeau numérique est une abstraction
- L'informatique change notre façon de faire des sciences
  $\rightsquigarrow$ Y compris notre façon de faire des mathématiques

Références :

- Gilles Dowek, 2007, Les métamorphoses du calcul : Une étonnante histoire des mathématiques, Le Pommier
- Liens (en vert) dans les slides