La LOGIQUE
(1) de la naissance de l'informatique aux applications actuelles
(2) en Intelligence Artificielle et
(3) en sécurité des Systèmes Informatiques

Christian Retoré

Professeur, Université de Montpellier et LIRMM

Responsable master intelligence artificielle et science des données

Responsable axe « logiques » du LIRMM

18 janvier 2024
Lycée Louis Feuillade
LUNEL

Etudes et thèse maths pures 1993
chercheur maths-info (INRIA) 1994-2003
prof en info (univ Bordeaux puis Univ Montpellier) 2003→

Qu’est-ce que l’informatique?

- Science?
- Technologie?

Applications
« visibles » (analyse traitement d’image)
« impressionnantes », (chatGPT),
« plaisantes » (jeux)
mais disent-elles la nature de l’informatique?

- Citation de Fellows Parberry (attribuée à Dijkstra)
- **« Computer science is no more about computers than astronomy is about telescopes »**
- **« L’informatique ne s’occupe pas plus des ordinateurs que l’astronomie ne s’occupe des télescopes. »**

PAS de démo… car c’est trompeur.

- Par exemple démo en biologie de traitement contre le covid par un antiviral
  1. Photo d’un patient mal en point
  2. Photo du patient prenant un comprimé d’antiviral
  3. Photo du patient guéri.
- C’est impressionnant….
- Mais ça ne m’apprend rien sur
  1. comment le virus infecte les cellules, ni sur
  2. comment fonctionne le traitement….

- Les **démos** en info c’est pareil!
- C’est impressionnant…
- Mais **ça n’explique rien** de ce qu’est l’informatique mise en œuvre.

Mathématiques et informatique

- Maths & info ≠
  physique, chimie biologie, économie, linguistique,…
- Maths & info: PAS des sciences empiriques.
- Etudes de constructions abstraites, inventées par l'esprit humain.
- Implémentées sur des machines inventées et construites par l'être humain.

Pour les nombres entiers c'est discutable. Il y a au moins une intuition forte.

Comparaison avec la mécanique:

- *On **observe** qu'en jetant un caillou il suit une parabole…*
- *Puis Newton, la gravité, la mécanique classique, propose un modèle qui corrobore les observations.*

- *En informatique?*
- *Gödel **invente** la notion de calculabilité et de codage*
- *Turing (ingénieur) et Von Neumann en font un ordinateur avec l'électronique existante (1945?)*

La logique dans l'enseignement

- *En tout cas, en France, la logique est assez peu enseignée:*
  - *Philosophie? (un peu en terminale)*
  - *Mathématiques? (rare)*
  - ***Informatique? (dans toutes les licences et bcp de masters)***
  - *Linguistique (sémantique) et philosophie du langage?*

Computer science vs informatique

- *Calcul (~petits cailloux avec lesquels on calculait)*
- *Information (représentation des données et connaissances).*

- *Dès le départ, il y a les 2 aspects.*
- *Représentation des données par des entiers sur lesquels on peut calculer. (entier ~ suite de 0 et 1, parfait pour l'électricité)*

Qu’est ce que l’informatique?
D’où vient elle, historiquement?

- *Il y a des machines à calculer mécaniques:*
  - *Abaques boulier chinois grec abyssinien indien… antiquité*
  - *Blaise Pascal ~ 1650*
  - *Charles Babbage ~1850*
- *Mais ce n’est pas de là que proviennent:*
  - *L’idée de calcul et résolution de problèmes mécanisable*
  - *Les premières machines*
- *Origine de l’info:*
  - *Maths*
  - *Et surtout logique mathématique.*

Logique, histoire résumée

Aristote

L'antiquité après Aristote

Le Moyen-Âge et la scolastique

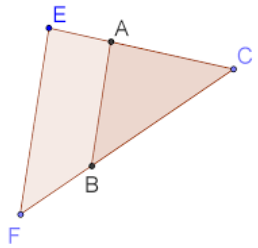
La logique algébrique (XVIIe XVIIIe XIXe)

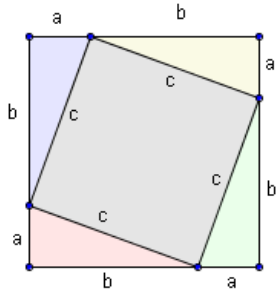
La logique est elle sulfureuse?

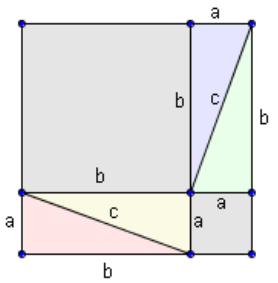
Lucifero: "Forse tu non pensavi ch'io LOICO fossi. Dante Alighieri (1265-1321) Comedia, Inferno XXVII

Une traduction pourrait être: "*sans doute ne savais tu pas que j'étais aussi bon logicien".*

La logique



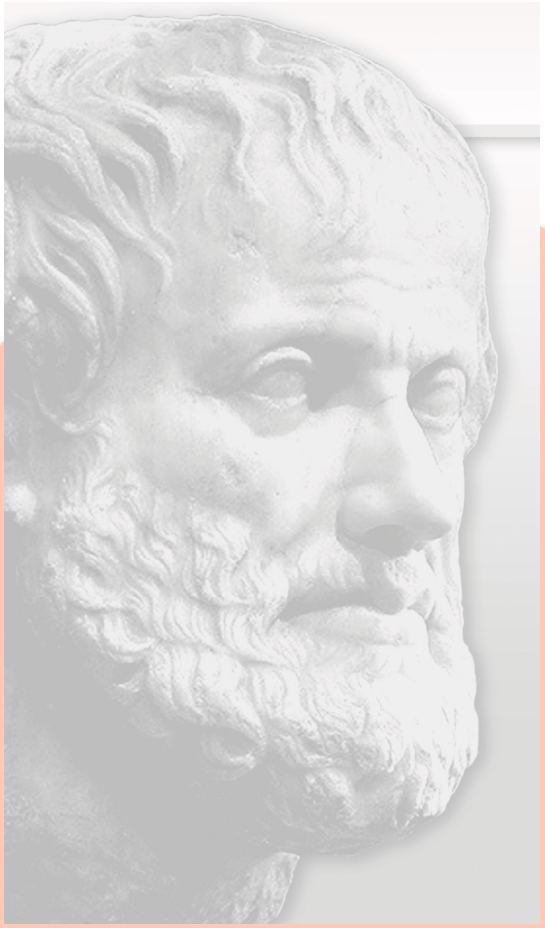




- Art de raisonner correctement
- Avec la rigueur des raisonnements mathématiques (Thalès, Pythagore,… VIIe siècle av. J.C)
- Dériver correctement des énoncés …mais à partir de quels axiomes?
- Etude de la vérité dans une situation particulière, mais cela est plus récent.

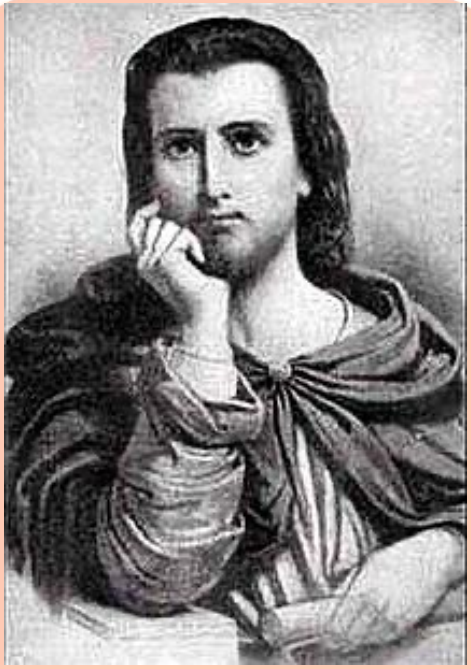
Aristote (III av JC) l'antiquité & la scolastique (moyen âge)

- Certains types d'énoncés:
  - A Tout A est B
  - E Certains A sont B
  - I Aucun A est B
  - O Tous les A ne sont pas B. (ou Certains A ne sont pas B, mais le **thème** est différent)



La scolastique (Antiquité et Moyen-Âge)

- Les fameux syllogismes (règles de déduction)
- Barbara :
  - *tout M est P,*
  - *or tout S est M,*
  - *donc tout S est P;*
- Baroco :
  - *tout P est M,*
  - *or quelque S n'est pas M,*
  - *donc quelque S n'est pas P*

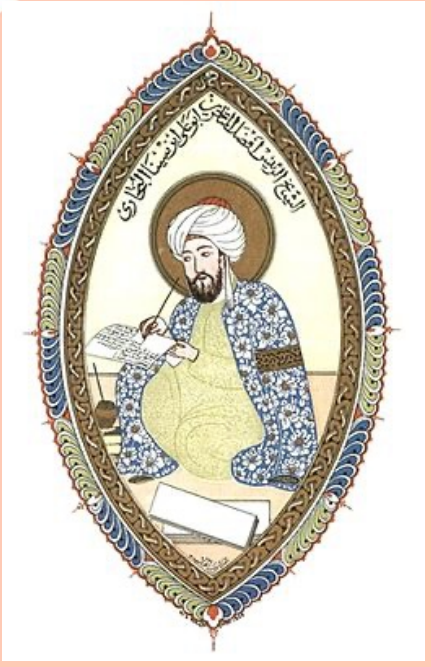


Pierre Abélard (1079-1142)

Principes (Aristote, Avicenne)

Avicenne ibn Sina (980-1037)

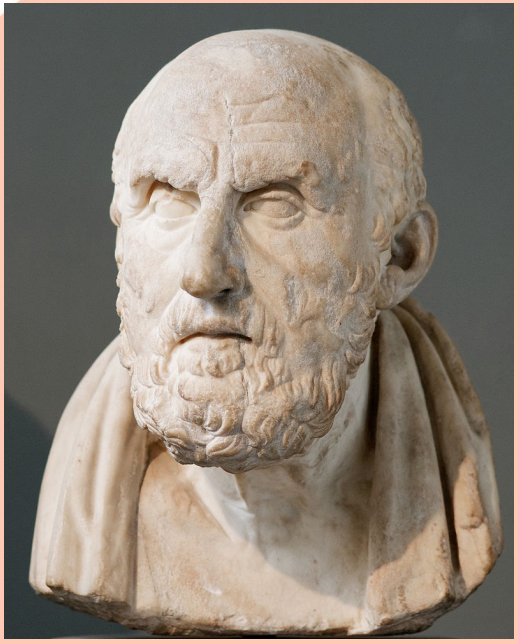
- Identité: Tout A est A
- Non contradiction NON (G et NON G) *"Tout personne niant le principe de non contradiction devrait être battue et brûlée jusqu'à ce qu'elle admette qu'être battu n'est pas la même chose que ne pas être battu, et qu'être brulé n'est pas la même chose que ne pas être brûlé"* Avicenne (980-1037) en réponse à des religieux souhaitant accommoder ce principe.
- Tiers exclus: pour tout énoncé G on a (G ou NON G) **(tertium non datur)**



Principes (Stoïciens)

- Stoïciens (calcul propositionnel)
- Modus ponens:
  - Si A alors B
  - Or A
  - Donc B.
- Modus tollens:
  - Si A alors B.
  - Or NON B.
  - Donc NON A.
- **Ex falso quodlibet sequitur**

Chrysippe de Soles
logicien stoïcien
(280—206 av. JC, Anatolie).



Place de la logique dans l'enseignement

- A étudier en premier pour raisonner correctement (Organon, Catégories,..)
- *« Celui qui souhaite atteindre la perfection humaine doit d'abord étudier la logique, puis les diverses branches des mathématiques dans l'ordre qui convient, puis la physique et enfin la métaphysique. »* (Maimonides, XIIe)

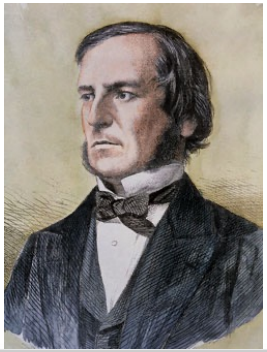
Logique algébrique Leibnitz (XVIIIe) et ses successeurs Boole, De Morgan, Pierce

- Précurseur: Leibniz (1646-1716)
- Lois et calculs
- Calcul propositionnel: tables de vérité
  - X → VRAI : VRAI
    (Si Rome est en Chine alors Paris est en France)
  - FAUX → X : VRAI
    (Si Rome est en Chine alors Paris est en Chine)
  - VRAI → FAUX : FAUX
    (si Paris est en France alors Rome est en Chine)

Logique algébrique anglo-américaine Boole, De Morgan, Pierce (XIXe)

- Pour les prédicats des règles parfois fausses
  « si toute personne est (F ou H)
  alors (toute personne est F ou toute personne est H) »

- C'est évidemment FAUX !



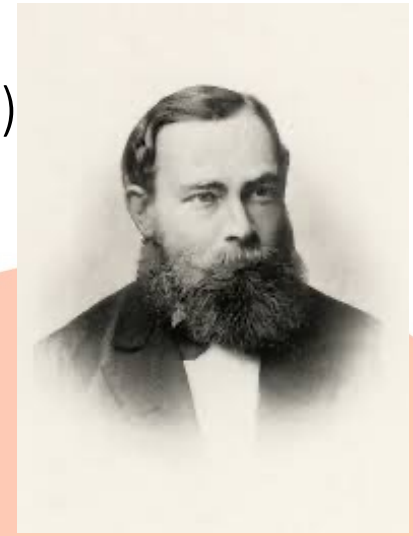
Le XXe siècle

La crise des fondements des mathématiques

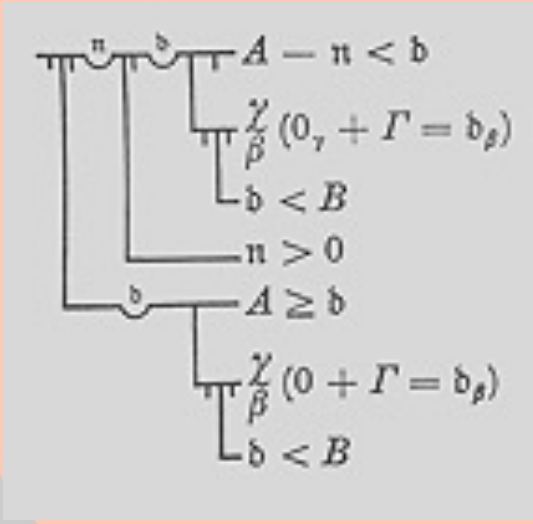
Les débuts de la logique mathématique

La logique du premier ordre

Calcul des prédicats (formules quantifiées)
Gottlob Frege (1848-1925), David Hilbert (1862-1943)



- Incluent strictement les énoncés A E I O d'Aristote
- *Tout entier est la somme de quatre carrés:*
  pour tout entier n
  il existe quatre entiers a b c et d tels que $n=a^2+b^2+c^2+d^2$
- Idéographie: notation (horrible) pour les formules et les preuves



Formalisation des quantificateurs

- Tout x est P: noté ∀x P(x)
  « tout », « tous les », « chaque »,
  (et même « un »:
  « *un homme averti en vaut deux* »)
  - Tout A est B
  - Pour tout X, si X est A alors il est B
- Au moins un x est P: noté ∃x P(x)
  « certains » « quelques » « des » « un »
  - Au moins un A est B.
  - Il existe un X qui est A et qui est B.

David Hilbert

Jacques Herbrand

Déductions formelles

- Des axiomes, par ex.
  - Si A alors A
- Des règles par ex.
  - SI A et (si A alors B) alors B
  - Si on a A en supposant B, alors on a A=>B
  - Si A et Si B alors A et B
  - Si pour tout x A(x) alors A(y) pour un y particulier.
  - Si A(y) pour une variable qui n'a rien de spécial alors « pour tout x A(x) »
  - Si A(t) est vrai pour un t particulier, alors il existe x tel qu A(x)
  - … que des évidences!

Calcul des prédicats: vérité dans un modèle
Leopold Löwenheim (1878-1957)
Thoralf Skolem (1887-1963)

- Quand on fixe
  - Le domaine par ex. les élèves du Lycée Philippe Lamour
  - Les propriété par ex. l'élève x est en 2nde 3 ou en 1ère 4 etc.
  - Les relations binaires par ex. l'élève x connaît l'élève y
  - (ça s'appelle un modèle du langage)
- Alors les phrases quantifiées deviennent vraies ou fausses (dans ce modèle):
  - Chaque élève de 2nde 3 connait au moins un élève de 1ère 4.
  - Vrai cette année / peut-être faux l'an prochain.

Calcul des prédicats vérité dans un modèle

- Il y a des formules vraies dans TOUT modèle:
  - **SI** il existe X tel que pour tout Y
    X soit en relation R avec Y
    (par ex. quelqu'un connait chaque personne)
    ALORS pour tout Y il existe un X
    tel que X soit dans la relation R avec Y
    (chaque personne est connue d'une personne
    par ex. celle qui connait tout le monde)
  - C'est-à-dire $\exists x \, \forall y \, R(x,y) \Rightarrow \forall y \, \exists x \, R(x,y)$

Complétude

- **Validité:** Toute formule démontrable formellement est vraie dans tout modèle: par ex. « pour tout x, si x dort alors x dort »
- **Complétude (Gödel, 1929) :**
  Toute formule vraie dans tout modèle est formellement démontrable

Giuseppe Peano (1858-1932)
Axiomatisation de l'arithmétique

- Esperanto mathématique …
- Arithmétique: symboles:
  - 0 (zéro)
  - fonction S: successeur (l'entier suivant)
  - + (addition)
  - x (multiplication)

Axiomes de l'arithmétique

- S(n) successeur de n, c'est-à-dire n+1 (l'entier qui suit n).

- ∀n S(n)≠0
- ∀n si n≠0 alors il existe p tel que S(p)=n
- ∀n ∀p si S(n)=S(p) alors n=p
- ∀n n + 0 = n
- ∀n ∀p n+S(p) = S(n+p)
- ∀n n * 0 = 0
- ∀n ∀p n * S(p) = (n * p) + n

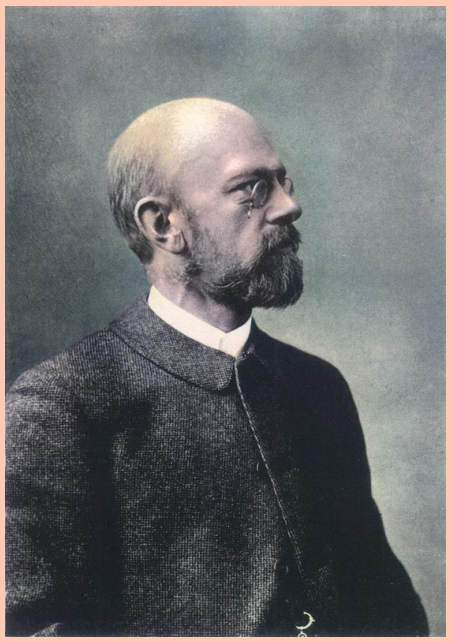
… ainsi que le schéma d'axiomes de **récurrence**!

- Récurrence:
  pour établir que
  « la propriété P(…) est vraie de tout entier »
- il suffit de
  - Montrer qu'elle est vraie de 0
  - Montrer qu'elle passe au successeur:
    si P(n) alors P(S(n)) — avec S(n)=n+1.
- C'est un **schéma** d'axiome:
  il faut UN axiome
  pour CHAQUE propriété P(…) des nombres.

Programme de Terminale

David Hilbert (1862-1943) et le programme éponyme (1900)

- En 1900 à Paris 23 problèmes majeurs
- Programme de fondements des mathématiques:
  - Etablir la cohérence des mathématiques
  - Un langage et une axiomatique minimales



David Hilbert (1862-1943) et son programme: les preuves formelles à l'honneur

Montrer en raisonnant sur les preuves formelles que les axiomes

ne conduisent jamais
à une proposition fausse

(par ex. 0=1)
par les règles

de déduction formelle

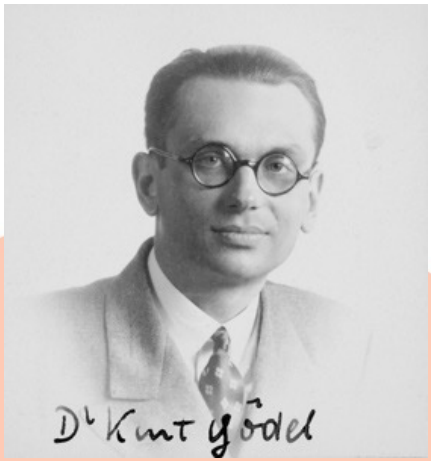
Mécanisation du raisonnement, notamment arithmétique.

C'est dans ce cadre qu'apparaît l'**informatique** !!!

Logique, informatique et calculabilité

Les ordinateurs sont issus de la notion de calculabilité et non l'inverse.

Kurt Gödel (1906-1978)



- Montre que les mathématiciens de son époque se trompent: il n'y a pas de procédé mécanique pour démontrer toutes les propriétés de l'arithmétique.
- Définit la première notion de calculabilité:
  - que veut dire résoudre mécaniquement?
  - Quelles fonctions de N dans N sont calculables? Pas toutes!
- Définit le codage des suites finies de caractères par des entiers: un tableau d'entiers, une formule, une preuve, une image pixélisée -> un entier!

Codage

- Programmes que sur des entiers… pas fun!
- Programme sur des messages, des images, des videos,… ça c'est plus fun.
- Un codage (codes trop grands mais simple:
  - Caractère → un nombre
    - 0→0 …. A→10 B → 11….
  - Suite de caractères s1 s2 s3 …→ un nombre
    - $2^{[S1]}\ 3^{\{s2]}\ 5^{[s3]}$
  - Codage de NxN dans N : < ,>
  - Codage de NxNx…xN dans N
  - Codage des suite finies d'entiers:
  - <p,<<<n1,n2>,n3,…>

Toute DONNÉE est représentée par UN NOMBRE
→Suite de 0 et de 1
→Traitement informatique possible

Résultat (difficiles)

- Une formule → un entier
- Une démonstration formelle est une suite de formule → un entier.
- Pr_{PA} (d,f) d est le code d'une démonstration D dans PA qui démontre la formule F dont le code est f
- Il existe d, Pr_{PA} (d,f) : f est démontrable dans PA.
- Gödel: il existe des formules G telles que
  - PA ne démontre pas G
  - PA ne démontre pas « non G »
  - G est démontrable dans PA est indécidable (**PAS calculable!**)
- En particulier PA ne démontre pas que PA ne démontre pas 0=1
- **Vous trouvez ça compliqué… vous avez raison!**
- ***Les plus grands mathématiciens de l'époque ont mis des années à comprendre la signification de ces résultats.***

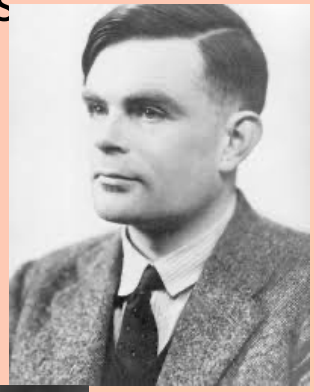
C'est la première notion de CALCULABLE!

Et de suite on affirme que tout n'est pas calculable!

Et en particulier, on ne pourra jamais remplacer les mathématiciens par des ordinateurs!

Et l'informatique dans tout ça?

- Gödel en visite puis installé à Princeton,
- Church notion de calculabilité en composant des fonctions (cf. langages fonctionnels: Lisp, CaML, Haskell, Scala,…)
- Turing ingénieur anglais, vient faire sa thèse avec Church
- Machine de Turing
- Machine de Von Neumann… -> Ordinateurs actuels.



Logique et Sécurité

Applications en Génie Logiciel:

Spécification, conception et vérification de programmes et de systèmes cyber-physiques

Programmes certifiés: les preuves vues comme des programmes

- Santé, aéronautique, finances, protocoles de connexion…
  il faut des programme sûrs:
  - Vérification
    - Model checking
    - Preuve de programme
  - Extraction de programmes depuis les preuves

Model Checking / vérification

- On spécifie le le système informatique POSSIBLEMENT parallèle (à la différence d'un programme)
- On utilise une logique qui permet de dire:
  - Dans toutes les situations à partir de maintenant on a …
  - Dans au moins une situation à partir de maintenant on a …
- On cherche à montrer que les bonnes propriétés sont toujours vérifiées:
  - A tout instant Si l'ascenseur monte ou descend, alors les portes sont fermées
  - A tout instant Si les portes sont ouvertes alors l'ascenseur est en face d'un étage.
  - A tout instant Il y aura plus tard un instant ou l'ascenseur sera au rez-de-chaussée.

Preuves de programmes

- On spécifie le programme:
- Par ex si l'entrée est une liste d'entiers,
  alors la sortie est la liste des mêmes entiers, mais triée par ordre croissant.
- On prouve formellement que le programme s'arrête et que la sortie est bien la liste des même entiers ordonnée par ordre croissant.

Extraction d'un programme certifié à partir d'une preuve formelle

- **Procédure:**
  1. On écrit la spécification du programme: (équations)
  2. On démontre formellement que
     pour tout x de type A
     il existe un y de type B tel que P(x,y)
     en utilisant les règles de la logique et les équations.
  3. Cette preuve peut être vu comme un programme qui sera totalement correct:
     qui satisfait exactement la spécification.

Le programme obtenu (lent mais 100% sûr) fait exactement ce qu'il est supposé faire.

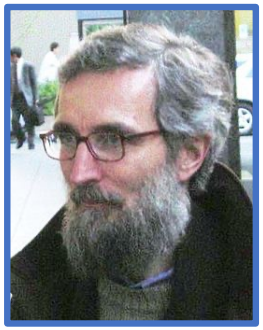
Basé sur la correspondance logique ⇔ programme fonctionnel

LOGIQUE	INFORMATIQUE
Formule A	Type A
Preuve d de A	Programme d de type A
Normalisation de d (remplacement des hypothèses par leur preuve)	Evaluation de d (substitution des variables par leur valeur)

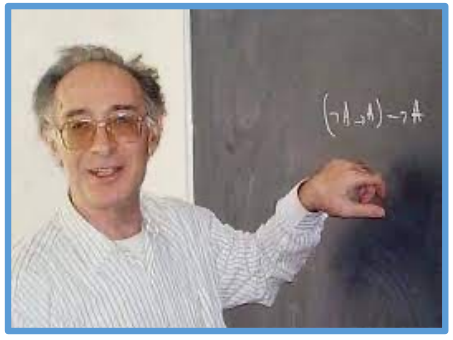
Preuves: des fondements des maths à la logique du calcul

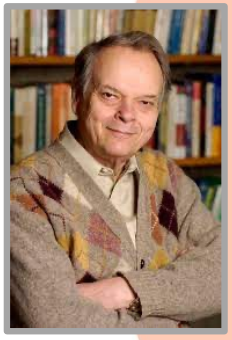
- Initialement: pour la cohérence des mathématiques
- Contenu calculatoire (dans une logique particulière)
- Programmes fonctionnels typés
  - Temps de calcul élevé
  - Mais totalement sûrs (important par ex. circuits aéronautiques)
- Ce sont des langages où tout programme termine.
  C'est lié aux propriétés des preuves de la logique utilisée.

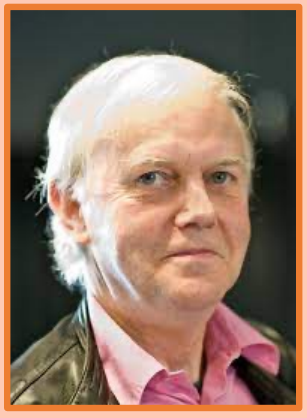
Preuves



- Fondements des maths / informatique dès 1970
  - Jean-Yves Girard (1947-…) *mon directeur de thèse*
  - John Reynolds (1935-2013)
  - Per Martin-Löf (1942-…) Vladimir Voevodski (1966-2017)
  - Jean-Louis Krivine (1939-…)
  - Gérard Huet (1947-…)







Muddy Children

Résolution d'un puzzle de type jeux en logique modale

Les enfants sales (muddy children) 1/2

- 3 enfants A B C jouent dehors, il y a de la boue dehors.
- Certains ont de la boue sur le front.
- Chaque enfant voit si les autres ont de la boue sur le front.
- Aucun n'enfant ne voit si lui-même à de la boue sur le front.

Les enfants sales (muddy children) 2/2

1. Le père va voir les enfants
2. Le père leur dit « l'un au moins d'entre vous a de la boue sur le front ».
3. Le père dit si l'un d'entre vous a de la boue sur le front, qu'il lève la main.
4. Aucun enfant ne lève la main.
5. Le père dit si l'un d'entre vous a de la boue sur le front, qu'il lève la main.
6. Aucun enfant ne lève la main.
7. Le père dit si l'un d'entre vous a de la boue sur le front, qu'il lève la main.
8. Tous les enfants lèvent la main.

Pourquoi? Comment?
Lesquels ont de la boue sur le front?

Modélisation

a. L’enfant A a de la boue sur le front.
b. L’enfant B a de la boue sur le front
c. L’enfant C a de la boue sur le front.

Modalités:

Modalités:

K_a l'enfant A sait que

K_b l'enfant B sait que

K_c l'enfant C sait que

Relations: situations indiscernables pour un enfant:

R_a l'enfant A ne fait pas de différence

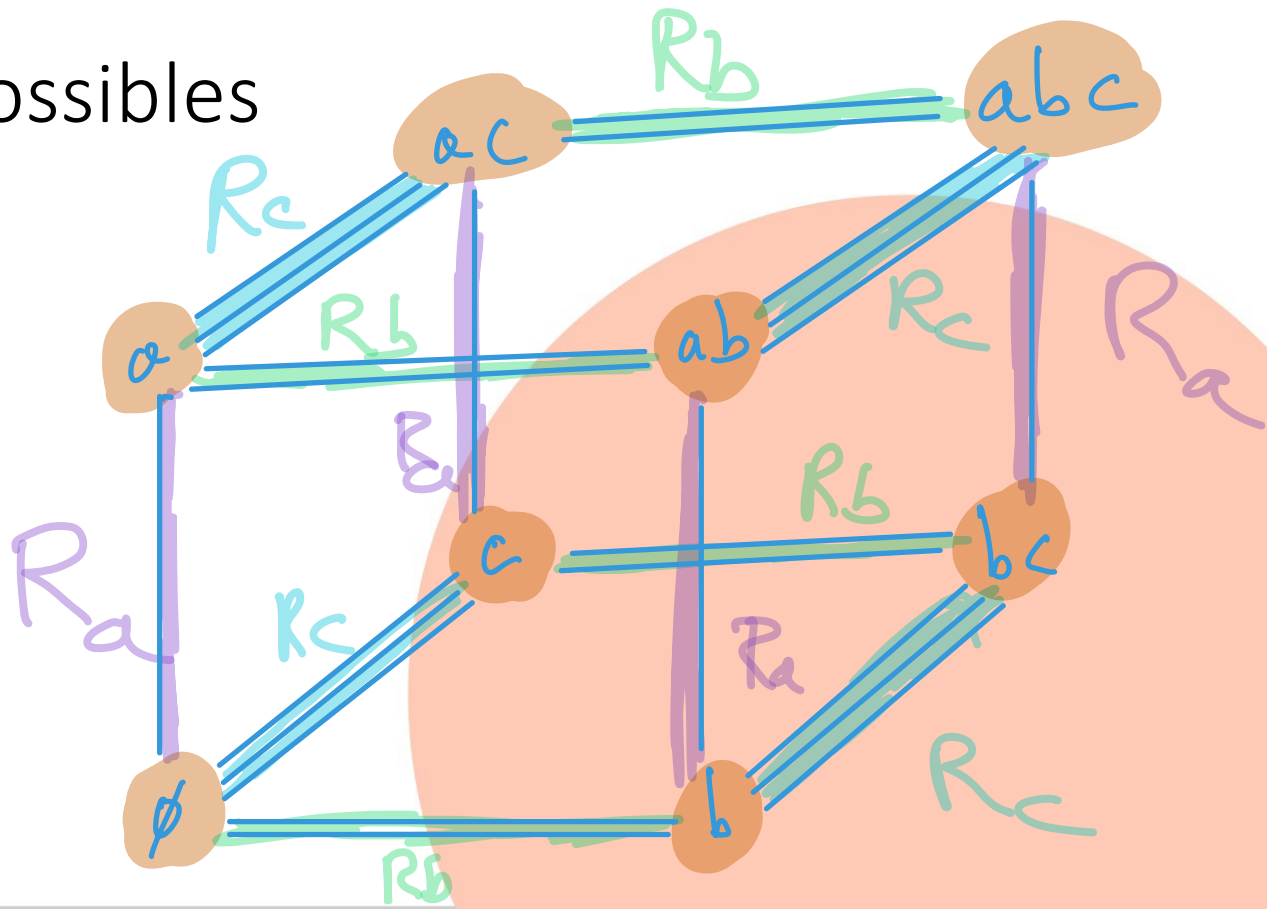
R_b l'enfant B ne fait pas de différence

R_c l'enfant C ne fait pas de différence

K_xG vraie dans une situation S
lorsque G est vraie
dans toutes les situation S' indiscernables de S par x.

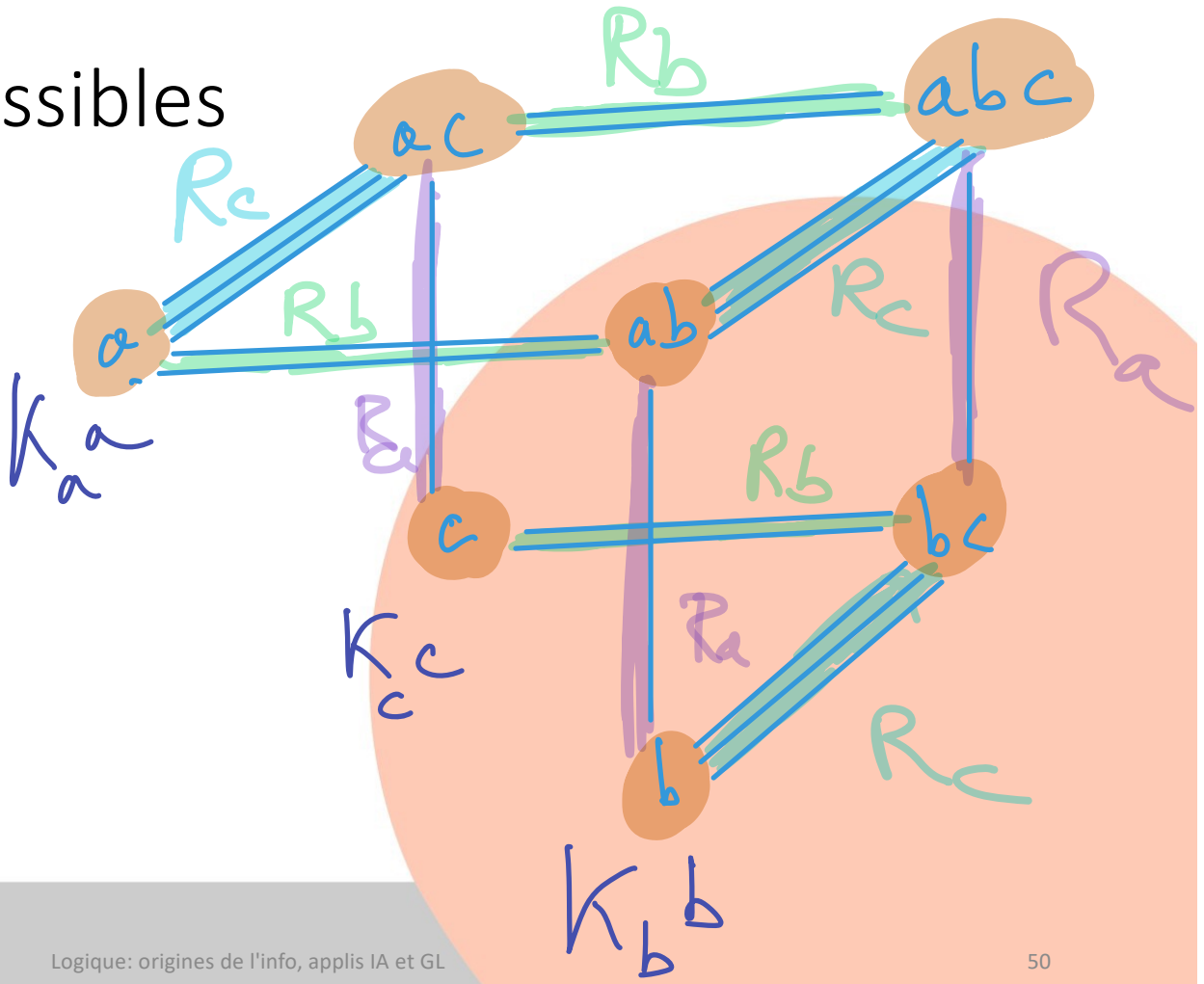
Situations possibles

1. Ø
2. a
3. b
4. c
5. ab
6. ac
7. bc
8. abc



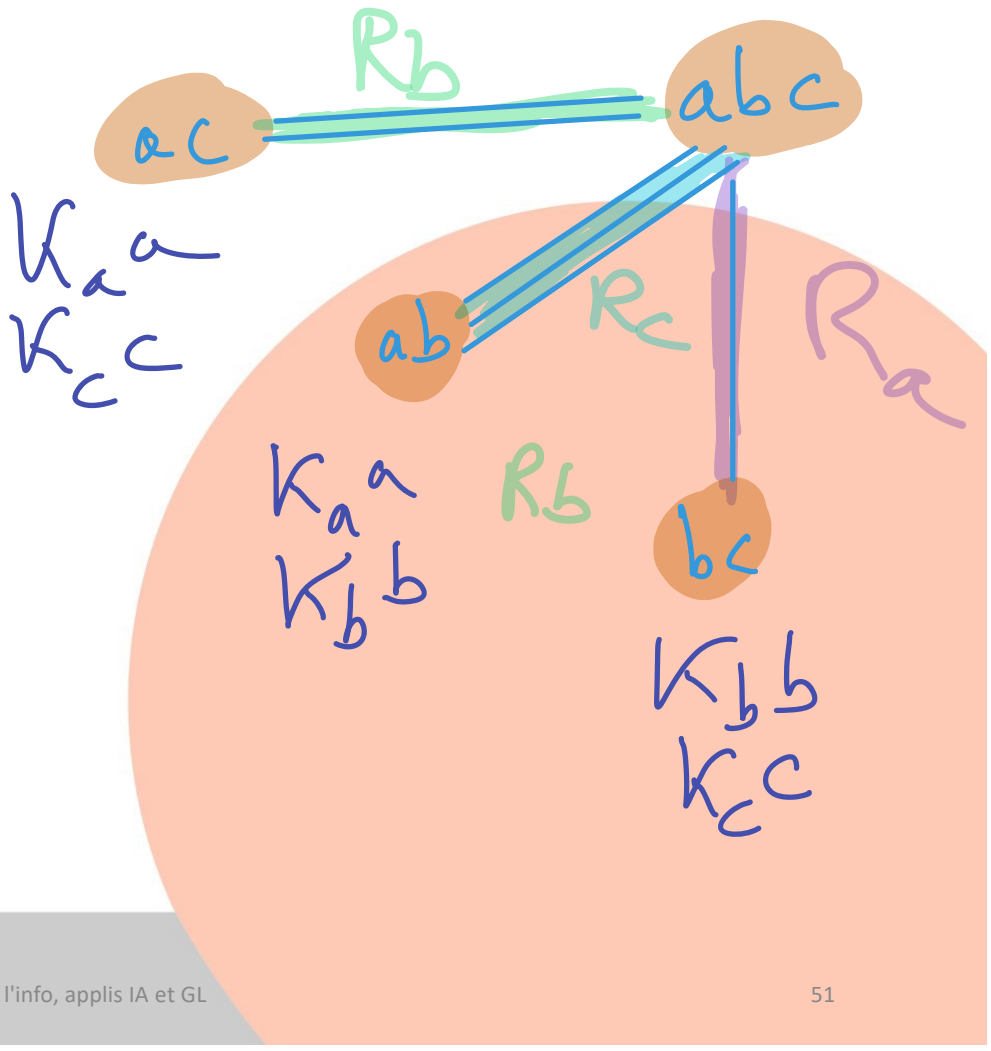
Situations possibles

1. ~~∅~~
2. a
3. b
4. c
5. ab
6. ac
7. bc
8. abc



Situations possibles

1. ~~∅~~
2. ~~a~~
3. ~~b~~
4. ~~c~~
5. ab
6. ac
7. bc
8. abc



Situations possibles

1. ~~Ø~~
2. ~~a~~
3. ~~b~~
4. ~~c~~
5. ~~ab~~
6. ~~ac~~
7. ~~bc~~
8. abc

"Contrariwise,
if it was so, it might be;
and if it were so, it would be;
but as it isn't, it ain't.
That's logic."
Lewis Carroll (1832-1898)

Quelques références:

Quelques références

- **BD LOGICOMIX** de Apostolos Doxiadis, Christos H. Papadimitriou, Alecos Papadatos, and Annie di Donna Vuibert 2010
  Les principaux logiciens du début du XXe sont mis en scène.
  Pas technique, mais des annexes permettent d'aller plus loin.
- ***Video*** *YouTube* **Les théorèmes de Gödel (***David Louapre, directeur scientifique UbiSoft). Science étonnante #34*
- **Roman La déesse des petites victoires** de Yannick Grannec Editions Anne Carrière. 2012 Prix des libraires 2013. *Gödel vu par sa femme. Très bonne reconstitution de la vie scientifique à Vienne puis à Princeton, avec des personnages comme Einstein (l'unique ami de Gödel), von Neuman, Morgenstern,…Très bon roman* per se.
- **Livre** Lewis Carroll **Logique sans peine.** Illustrations de Max Ernst.
- **Livre** Raymond Smullyan **Le livre qui rend fou.**
- **Livre**: Raymond Smullyan **Quel est le titre de ce livre?**

Conseils pour vos futures études

Un seul conseil: faites ce qui vous plait.

A l'université master bac+5 (ingénieur)

- ≠IUT (bac +2/3) Technicien passerelle compliquée aujourd'hui.
- En informatique la progression des étudiants est souvent bloquée par leur faible niveau **maths** (en NSI, ça se rattrape)
- Pour parcours sup, recrutement en master: maths et info théorique.
- 2 sortes de maths:
  - Calcul (image, apprentissage)
    - matrices, vecteurs
    - Probabilité et statistiques
    - Pas beaucoup d'analyse (études de fonction)
  - Logique (arithmétique, raisonnement mathématique)

Choix difficiles pour un ado… Quelques principes.

- Classe prépa / univ (IUT différent formation pro courte)
  perso bac 16 ans, classe prépa (perso bof bof, contraintes, physique-chimie),
  maths université Paris 6 -> thèse en maths Paris 7
- Faites ce que vous aimez, comme cela vous allez réussir.
- Qu'est ce que vous aimez faire? Question plus difficile que vous croyez.
- N'écouter pas vos amis, ni votre famille
  (pas objectifs, juge et partie, trop d'affect)
- Vous pouvez écouter vos profs qui connaissent les filières et vos aptitudes.
- Sachez que quelle que soit la filière ceux qui réussissent bien
  trouvent un travail intéressant et bien rémunéré.
- La solution est en vous!
- En informatique ceux que l'informatique intéressent vraiment (pas comme simple utilisateur) réussissent bien et trouvent tous un bon travail.
- Mais si vous n'aimez pas l'informatique, surtout ne faites pas d'études d'informatique,
  il y a beaucoup d'autres choses passionnantes
  que ce soit dans les études possibles mais aussi en dehors des études.