Logique mathématique et Informatique
Où sont les interactions?

Richard Lassaigne
IMJ/Logique mathématique
CNRS-Université Paris Diderot

Quelques grandes questions pour la logique mathématique

- Quel est le pouvoir d'expression d'un langage (formel) ?

En mathématique, informatique, linguistique

- Qu'est-ce qu'une preuve dans un système formel ?

Etude des systèmes de déduction

- Qu'est-ce qu'un algorithme, un modèle de calcul ?

Etude des fonctions calculables, des propriétés décidables

- Qu'est-ce qu'un algorithme efficace ?

Etude de la complexité algorithmique, structurelle, descriptive

- Qu'est-ce que l'infini (d'un point de vue mathématique) ?

Etude des cardinaux et des ordinaux transfinis

Petite histoire des fondements de l'informatique

- La notion informelle d'algorithme remonte à l'antiquité. L'utilisation de ce concept peut être attribuée aux Grecs : **Erathostène**, **Euclide**
- Le terme *algorithme* provient d'une déformation latine du nom du mathématicien **Al Khwarizmi**, au 9e siècle. Il a écrit plusieurs livres sur l'algèbre, l'astronomie et la géographie, mais le plus célèbre de ses travaux est certainement celui sur l'algèbre, introduite ici pour la résolution de problèmes de la vie réelle
- **Pascal** (1642) invente une machine à calculer capable d'additionner et de soustraire des nombres de huit chiffres
- **Leibniz** (1673) améliore la machine de **Pascal** (1642) en y ajoutant la multiplication et la division, et s'intéresse au calcul binaire
- Ada **Lovelace** (1840) contribue à la création de la *machine analytique* de **Babbage** et conçoit peut-être le 1er langage informatique
- **Boole** (1854) publie son livre *The Mathematical Analysis of Logic*

Petite histoire des fondements de l'informatique (suite)

- **Turing** (1936) publie son article *On Computable Numbers...* dans lequel il définit un modèle mathématique pour la notion d'algorithme. Il prouve également l'impossibilité du problème général de décision, posé en 1928 par David **Hilbert**
- **Church** (1936) définit le *lambda-calcul* dont le pouvoir d'expression est équivalent à celui du modèle de Turing
- **Turing** travaille (1940-1944) dans le service anglais de décryptage des messages secrets allemands, codés suivant le système *Enigma :* il réalise plusieurs machines électromécaniques avec **Newman**
- **Von Neumann** (1945) écrit un rapport où il propose l'architecture interne d'un calculateur universel, utilisée dans la quasi-totalité des ordinateurs
- **Shannon** (1948) publie sa *Théorie mathématique de l'information*
- Grace **Murray Hopper** (1951) conçoit le 1er compilateur

Repères logiques pour les fondements de l'informatique

- La calculabilité : modèles et limites

  A.Turing, S.Kleene, A.Church (1936,...)

  K. Gödel (1931,1932)
- Les systèmes de déduction : G.Gentzen (1935)

  Déduction naturelle et calcul des séquents
- Le modèle relationnel des Bases de Données : E.F.Codd (1970)
- La complexité structurelle des problèmes : S.Cook (1971), R.Karp (1972)

  Complexité du problème SAT et réduction entre problèmes
- La preuve de programmes : J.Y.Girard, J.L.Krivine, T.Coquand (1985,...)

  Correspondance de Curry-Howard, systèmes de types
- La vérification : A.Pnueli (1977), E.M.Clarke(1981), M.Vardi

  Logique temporelle et model checking

Plan de l'exposé

1. Panorama (historique) de la logique mathématique

2. Déduction et preuve de programmes

3. Vérification par model checking

4. Complexité et réduction entre problèmes

5. Logique du 1er ordre et bases de données

6. Menu des cours à venir

Logique, vous avez dit logique?

Epimondes a dit:

« Tous les Crétois sont des menteurs »

Or Epimondes était Crétois

Le barbier du village a une règle d'or:

il rase les habitants du village

qui ne se rasent pas eux-mêmes

Le barbier se rase-t-il?

Livre de R. Smullyan:

Quel est le titre de ce livre?

Logique, Mathématique et Informatique

- Naissance

  G. Boole, G. Frege, B. Russell, D. Hilbert, …

  Crise des Fondements (Cantor, Russell)

- Panorama de la Logique mathématique

  Zermelo-Fraenkel, K. Gödel, Gentzen,...

  Cohérence, Incomplétude, Preuves

- Problèmes et résultats

  K. Gödel, P. Cohen, Y. Matiyasevich,...

  Indépendance, Indécidabilité

- Logique et Informatique

  J.Y. Girard, J.L. Krivine, A. Pnueli, S. Cook, R. Karp,..

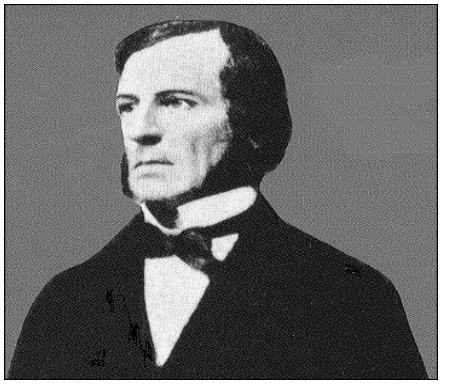
  Preuves et Types, Vérification, Complexité

Naissance

Le calcul propositionnel (calcul booléen)

G. Boole, 1847

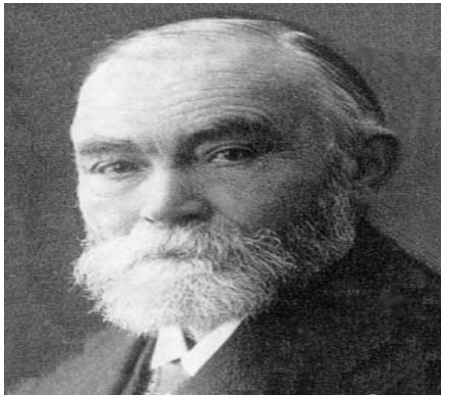
Présentation algébrique du calcul propositionnel



Formalisation du raisonnement

G. Frege, 1879

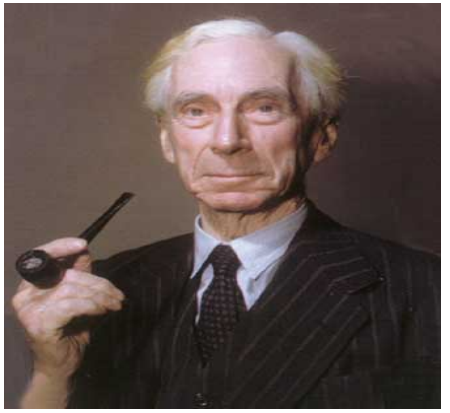
Notion de système formel



Principles of mathematics

B. Russell, 1903

Rejoint les conclusions de G. Frege



Fondements des mathématiques

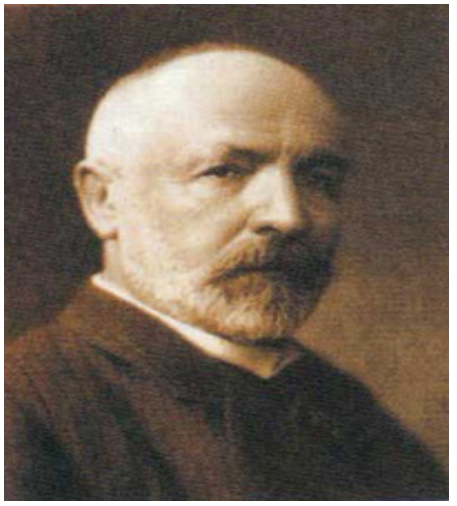
Axiomatisation des mathématiques: 1873-1895

- Analyse -------> Arithmétique Weierstrass, Dedekind,Cantor, Bolzano
- Arithmétique Dedekind, Frege, Peano,...
- Ensembles Cantor
- Géométrie euclidienne D. Hilbert, 1899

Théorie cantorienne :

Cardinalité des ensembles de réels

Etude mathématique de l'infini



Projets

- Logiciste:

  Réduction des mathématiques à une logique cohérente

  Principia Mathematica Russell, Whitehead 1910-1913

- Formaliste:

  Application des mathématiques à leur propre langage

  Programme de Hilbert, 1904

- Intuitionniste:

  Objections aux conceptions ensemblistes (Baire, Lebesgue, Borel)

  Poincaré

  et aux procédés non constructifs (thèse de Brouwer, 1907)

Logique et fondements des mathématiques

Questions posées par D. Hilbert (1900, 1904 et 1928)

- Les mathématiques sont-elles complètes ?

  Tout énoncé mathématique peut-il être soit prouvé, soit réfuté ?

- Les mathématiques sont-elles cohérentes ?

  Comment montrer la cohérence d'un système élémentaire,

  comme l'arithmétique ?

- Les mathématiques sont-elles décidables ?

  Existe-t-il un algorithme permettant de décider

  tout énoncé mathématique ?

Quelques résultats importants

- Incomplétude de l'Arithmétique

  Tout système formel suffisamment puissant

  et cohérent est incomplet

  K. Godel (1931)

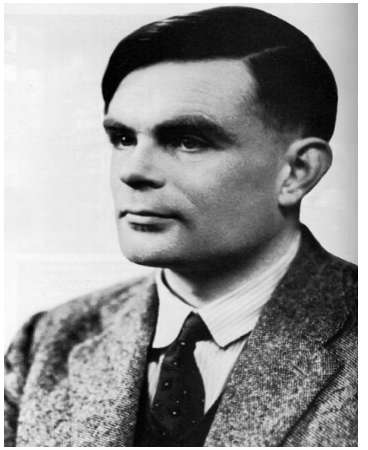
- Calculabilité et indécidabilité:

  Définition et construction d'un modèle de calcul

  Preuve de l'indécidabilté du problème de l'arrêt

  et de la logique du premier ordre

  A. Turing (1936)



Panorama de la Logique mathématique (suite)

- Développement de nouvelles branches (ou de nouvelles méthodes)
- Théorie de la démonstration :

  Déduction naturelle, calcul des séquents

  G. Gentzen 1935, Heyting
- Théorie des Modèles:

  Sémantique, construction de modèles

  Tarski 1931, Henkin, A. Robinson, Mal'cev, Morley, Vaught
- Théorie des Ensembles:

  Méthode du Forcing

  Cohen 1962, Solovay, Silver, Martin, Jensen
- Théorie de la Complexité:

  SAT, P=NP?, problèmes NP-complets, complexité descriptive

  S. Cook 1971, R. Karp, R. Fagin, N. Immermann

Résultats

- Thèse de A. Church:

  Calculabilité = Récursivité = Turing-Calculabilité

  = lambda-définissabilité = …

- Indécidabilité et Incomplétude de l'Arithmétique:

  K. Gödel, 1931

- Indépendance de l'axiome du choix et

  de l'hypothèse du continu:

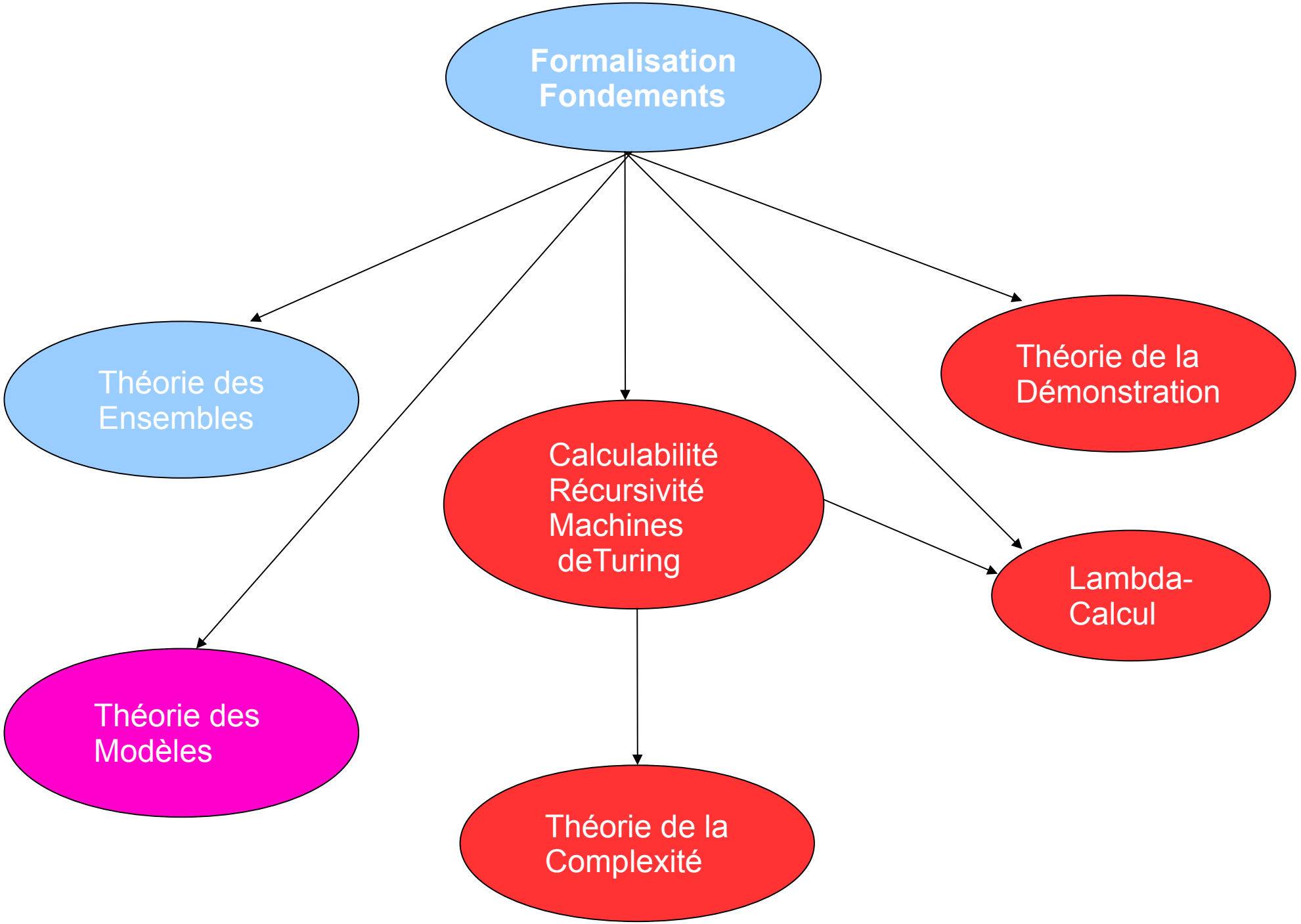
  K. Gödel, 1938, P.J. Cohen, 1962

- Indécidabilité des problèmes diophantiens:

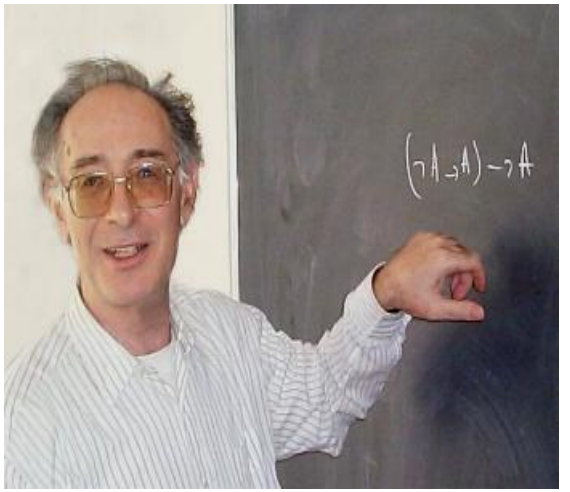
  M. Davis, A. Robinson, Y. Matijasevich , 1970

- Complexité du problème SAT S. Cook, 1971

  Réduction entre pb difficiles R. Karp, 1972


Formalisation
Fondements
Théorie des
Ensembles
Théorie de la
Démonstration
Calculabilité
Récursivité
Machines
deTuring
Lambda-
Calcul
Théorie des
Modèles
Théorie de la
Complexité

Logique et fondements de l'Informatique



- Preuves et Programmes :
  - Correspondance de Curry-Howard
  - Théorie de la démonstration
  - Systèmes d'aide à la preuve,
  - certification de programmes
- Vérification et Complexité :
  - Algorithmes de model checking
  - Vérification de circuits, protocoles, systèmes distribués,...
- Complexité :
  - Complexité structurelle, descriptive, preuves interactives
  - Preuves de sécurité (cryptographie),...
- Logique du 1er ordre et Bases de Données

Preuves et Programmes

Correspondance Types/Formules

Lambda-calcul typé (types simples):

- Types des termes de base
- Règles de construction des termes

  (abstraction, application)

- Réduction
- Terme général
- Type d'un terme

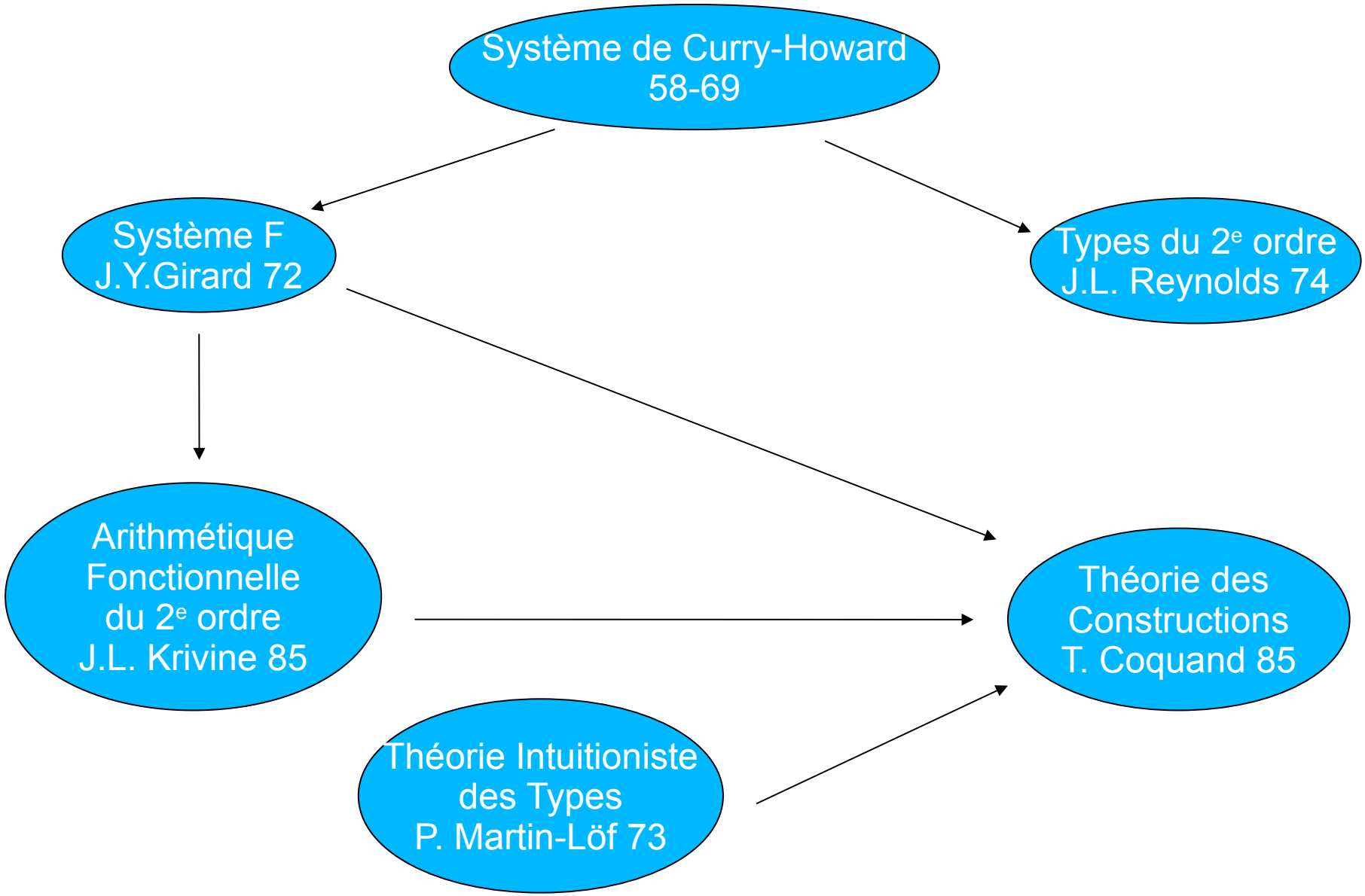
Logique propositionnelle intuitionniste:

- Axiomes Curry
- Règles de déduction concernant l'implication Howard

  (introduction, élimination)

- Elimination d'une coupure
- Preuve
- Formule

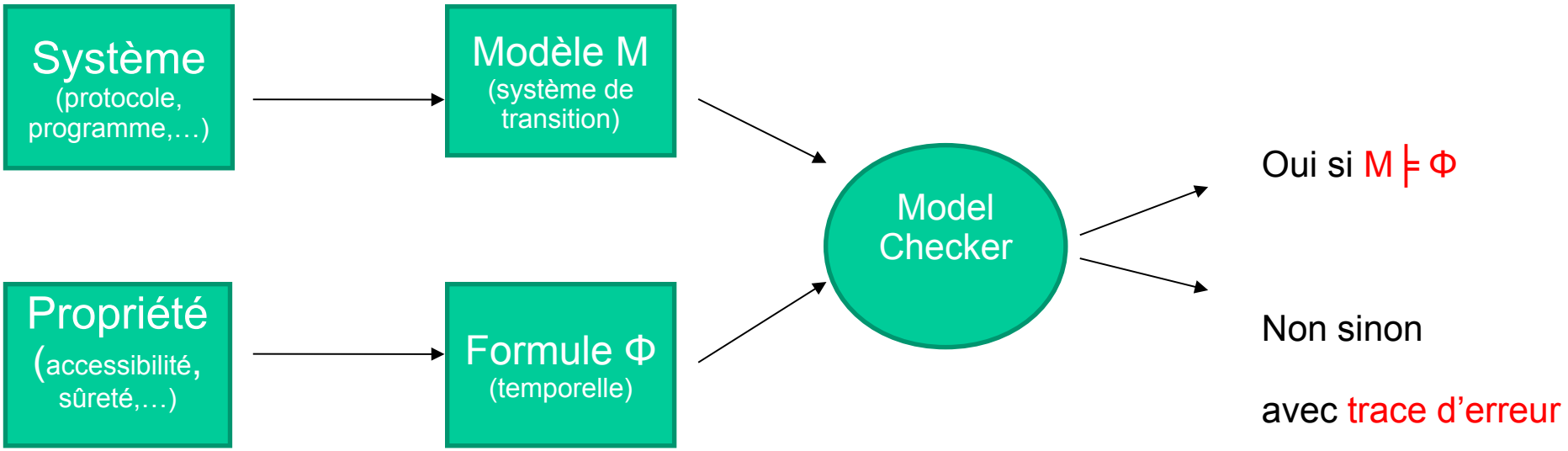
Le paradis (ou l'enfer?) des systèmes de types



Systèmes d'aide à la preuve

Proof Checker	Système de Types
COQ (INRIA, LRI)	Calcul des Constructions (T. Coquand)
PhoX, PML (C.Raffalli, Chambéry)	Arithmétique fonctionnelle du 2e ordre (J.L. Krivine)
Isabelle (L. Paulson, Cambridge) + Theorem Prover	Higher Order Logic (M. Gordon)
PVS (S. Owre,N. Shankar,J. Rushby , Stanford) + Theorem Prover	Théorie des types simples (A. Church)

Vérification par Model Checking



Complexité: linéaire dans la taille du modèle

Problème (modélisation): explosion combinatoire

Remarque: La complexité en espace est bien plus redoutable

(en pratique) que la complexité en temps

Complexité

Un problème célèbre (à 1 million de $): P=NP?

- S. Cook (71): le problème SAT est NP-complet
- R. Karp (72): de nombreux problèmes intéressants

(graphes, optimisation, combinatoire,…) sont NP-complets

Exemple: Pb de 3-coloriage d'un graphe

Entrée: Un graphe G=(E,R)

Question: Existe-t-il une fonction de coloriage f telle que

si (a,b) ε R, alors f(a) ≠ f(b)

Remarque:

- On ne connaît pas d'algorithme en temps polynomial pour ces problèmes
- Un d'algorithme en temps polynomial pour l'un de ces problèmes résoudrait

P=NP

Complexité descriptive

Question: Peut-on refléter la complexité (structurelle) d'un problème à l'aide de la complexité logique de la formule définissant ce problème?

Exemple: 3-coloriage d'un graphe

il existe une partition des sommets en 3 sous-ensembles t.q.

les extrémités de toute arête soient de 2 couleurs différentes

Théorème (R. Fagin, 74): Pour toute propriété sur les structures finies,

le problème de décision associé est dans NP ssi

elle est définissable par une formule existentielle du 2^e ordre

Théorème (N. Immermann, M. Vardi 82): Sur les structures finies ordonnées,

$$P \equiv FO + TC$$

FO + TC: logique du 1^{er} ordre + opérateur de clôture transitive

Complexité (suite et fin ?...)

Objectif: Distinguer entre problèmes difficiles (NP-complets, PSPACE-complets)

Méthodes:

- Approximation
- Algorithmes probabilistes

Classe de complexité probabiliste: la classe RP

Un langage L est dans la classe RP s'il existe un algorithme probabiliste A t.q.

- si $x \varepsilon L$, alors Prob[A accepte x] >1/2,
- si non ($x \varepsilon L$), alors Prob[A accepte x] = 0

Exemple: le problème de primalité est dans coRP

Logique du 1er ordre et Bases de Données

- L'algèbre relationnelle est un langage algébrique permettant d'exprimer des requêtes sur une BD : une expression de cette algèbre produit une nouvelle relation à partir des relations de la base de données
- Le calcul relationnel est un langage logique permettant d'exprimer des requêtes par des formules de la logique du 1er ordre
- Le théorème de **Codd** (1972) montre que les pouvoirs d'expression de ces 2 formalismes sont les mêmes et que la traduction d'un formalisme vers l'autre peut se faire en temps polynomial
- Ce résultat est très important car il permet d'utiliser un formalisme déclaratif pour exprimer des requêtes : la logique du 1er ordre ou SQL...
- Ces requêtes sont ensuite compilées via la transformation de Codd en un formalisme algébrique
- Les requêtes algébriques sont ensuite optimisées eu utilisant les propriétés de l'algèbre relationnelle (règles de transformation)
- Les requêtes optimisées sont enfin évaluées, en utilisant le fait que chaque opérateur de l'algèbre relationnelle s'implémente facilement

Plan du cours

1. Les interactions entre logique mathématique et informatique

   Exercices d'introduction à la logique mathématique
2. La logique propositionnelle. Conséquence logique. Formes normales

   Le problème SAT. Exercices
3. Les systèmes de déduction. Un exemple : la déduction naturelle

   La méthode des tableaux. Exercices
4. La logique du 1er ordre. Un exemple de logique pour la vérification :

   la logique temporelle linéaire (LTL). Exercices
5. Une introduction à la complexité : réductions, le problème P=NP?

   Casser la barrière de la complexité : approximation, algorithmes probabilistes

   Exercices

Références

Une référence générale :

- R. Cori et D. Lascar *Logique mathématique (I et II)* Masson (1993)

Deux références sur la complexité :

- M. R. Garey et D. S. Johnsson *Computers and Intractability: A guide to the theory of NP-completeness* Freeman and Co (1979)
- C. Papadimitriou *Computational Complexity* Addison-Wessley (1994)

Les rapports entre la logique, les modèles de calcul et la complexité :

- M. de Rougemont et R. Lassaigne *Logic and Complexity* Springer (2004)

Un polycopié de cours de l'école Polytechnique (2013):

- Olivier Bournez *Fondements de l'informatique : logique, modèles et calculs*

La méthode de vérification par model checking :

- Clarke, Grumberg et Peled *Model Checking* MIT Press (1999)