

Logique et Fondements de l'Informatique

Attendus du cours en 2023–2024

David A. Madore

21 janvier 2025

INF110

Git : c81bc04 Tue Jan 21 10:58:32 2025 +0100

Sont au programme du cours les notions suivantes :

- **Calculabilité** : fonctions primitives récursives, fonctions générales récursives, numérotation (notamment la notation $\varphi_e(i)$), théorème s-m-n, astuce de Quine, existence d'une fonction universelle pour les fonctions générales récursives (et inexistence pour les primitives récursives), théorème de la forme normale et possibilité de lancer des calculs en parallèle, théorème de récursion de Kleene ; indécidabilité du problème de l'arrêt, théorème de Rice ; machines de Turing et équivalence avec les fonctions générales récursives ; parties décidables et semi-décidables, équivalence entre semi-décidable et « image d'une fonction calculable » ; la notion de réduction many-to-one et de Turing ; le λ -calcul non typé, β -réduction, théorème de Church-Rosser, redex extérieur gauche, entiers de Church, équivalence du λ -calcul avec les fonctions générales récursives, combinateur Y.
- **Typage** : λ -calcul simplement typé, et sa version enrichie par les types produits, sommes, 1 et 0 ; terminaison des programmes écrits dans ce dernier (sans détails) ; correspondance de Curry-Howard entre λ -calcul simplement typé enrichi et calcul propositionnel intuitionniste ; algorithme de Hindley-Milner (basique).
- **Calcul propositionnel** : règles de logique en déduction naturelle, et au moins une présentation des preuves (arbre de séquents, ou drapeau) ; écriture et vérification des λ -termes de preuve (sans entrer dans le détail pointilleux des notations) ; différence entre logique intuitionniste et logique classique ; notion de calcul des séquents et de preuve sans coupure (sans détails) ; notions des axiomes de Hilbert / combinateurs S, K, I (sans détails).
- **Continuations et call/cc** : la notion générale de continuation, l'idée générale de la fonction call/cc et son rapport avec la loi de Peirce, l'idée générale du continuation-passing-style (sans détails).
- **Sémantiques du calcul propositionnel intuitionniste** : *au moins une* des quatre sémantiques vues en cours (Kripke, ouverts, réalisabilité propositionnelle, problèmes finis), sa correction, et comment on s'en sert pour montrer qu'une formule propositionnelle n'est pas démontrable.
- **Quantificateurs** : règles *générales* d'introduction et d'élimination du \forall et \exists , et λ -termes de preuve correspondants (sans entrer dans le détail pointilleux des notations) ; logique du premier ordre pure, logique du premier ordre avec égalité.

- **Arithmétique du premier ordre** : les axiomes de Peano ; l'idée générale que Curry-Howard sur l'arithmétique de Heyting permet d'extraire des algorithmes des preuves ; la possibilité de formaliser $\varphi_e(i) \downarrow$ en arithmétique de Heyting/Peano ; le fait que vérifier si une preuve est valable est décidable, mais que savoir si un énoncé est un théorème est seulement semi-décidable ; l'énoncé du théorème de Gödel et au moins une certaine idée de la preuve par machines de Turing.

Ne sont explicitement pas exigibles les notions suivantes :

- Les détails de la fonction d'Ackermann ; les détails de la notion d'arbre de calcul (autre que l'énoncé du théorème de la forme normale) ; la notion de degré many-to-one ou de Turing ; les notions de β -réduction autres qu'extérieur gauche, les subtilités de l'ordre d'évaluation, le combinateur Z ou sa différence avec Y.
- Les détails du typage de quelque langage de programmation que ce soit (autres que les variantes du λ -calcul simplement typé vus en cours, Hindley Milner, et les parties de Coq vues en TP), notamment rien de ce qui concerne Scheme, Haskell ou quelque autre langage mentionné en passant dans le cours ; le sous-typage, le polymorphisme ad hoc, les types dépendants, ou les autres fonctionnalités de certains systèmes de typages mentionnés en passant dans le cours. Les subtilités de l'algorithme de Hindley-Milner (problème du polymorphisme du let , restriction de valeur).
- Les subtilités des règles structurales en calcul des séquents. Le fonctionnement de l'élimination des coupures ou sa preuve. Le $\bar{\lambda}$ -calcul (juste mentionné en cours). Le détail de l'équivalence entre déduction naturelle et calcul des séquents. Le détail de l'élimination des λ grâce aux combinateurs S, K, I.
- Le fonctionnement détaillé de la fonction call/cc. Les détails du continuation-passing-style (conversion systématique) ou de son typage. Le $\lambda\mu$ -calcul (juste mentionné en cours).
- La complétude de telle ou telle sémantique du calcul propositionnel intuitionniste. Les subtilités de la réalisabilité propositionnelle (p.ex., la réalisabilité de la formule de Tseitin). La sémantique des problèmes finis.
- Le λ -cube de Barendregt, les subtilités de la différence entre \exists et types sommes, la notion de prédictivité/imprédictivité.
- Les détails de Curry-Howard pour quoi que ce soit d'autre que le calcul propositionnel intuitionniste.
- Les subtilités des différences et rapports entre Heyting et Peano (sauf s'il s'agit, par exemple, de vérifier si une démonstration donnée utilise un raisonnement par l'absurde).
- Les détails de la démonstration du théorème de Gödel, les systèmes précis auxquels il s'applique.