

Références et Développements

7 juin 2019

1 Références

- [Han] Handbook of Theoretical Computer Science - LEEUWEN - ?
Contient un bon chapitre par CROCHEMORE pour l'algorithmique du texte
- [Cor] Algorithmique - CORMEN - à la BU/LSV
La bible de l'algorithmique, avec toutes les bases. Attention, les calculs avec des probas sont parfois faux.
- [Wol] , Introduction à la calculabilité : cours et exercices corrigés - WOLPER - à la BU/LSV
Appréciable pour sa pédagogie, et la compréhension des concepts majeurs.
- [Bea] Éléments d'algorithmique - D. BEAUQUIER, J. BERSTEL, Ph. CHRÉTIENNE - à la BU/LSV
Bonne référence pour l'algo, pleins de dessins et de preuves. Un peu vieillissant et devenu rare.
- [Cro2] Algorithms on string - Crochemore - à la BU/LSV

- [Sip] Introduction to the Theory of Computation - Michael SIPSER - Bu
En anglais. Intuition de la preuve donnée avant chaque preuve.
- [Cro] Jewels of stringology - CROCHEMORE - à la BU/LSV

- [Sak] Éléments de théories des Automates - SAKAROVITCH - à la BU/LSV
Style austère, mais très complet, et bonne source de développements. Beaucoup de hors programme.
- [Aho] Compilers : Principles, Techniques, and Tools - AHO et. al - à la BU/LSV
La référence pour la compilation, surnommé le dragon. En réalité, seul deux/trois chapitres sont utiles.
- [Kri] Lambda-calcul, types et modèles - Jean-Louis Krivine -
Semble mieux que le Barendregt.
- [Go] Proof Theory and Automated Deduction - Jean GOUBAULT-LARRECQ - à la BU/LSV
Le must pour la logique
- [Da] Introduction à la logique - R. DAVID, K. NOUR, C. RAFFALLI - à la BU/LSV
L'autre must pour la logique
- [Per] Complexité algorithmique - PERIFEL - pas en BU
La référence parfaite pour la complexité. Formel, clair. C'est essentiellement une traduction formel du ARORA.
- [Win] The formal semantics of programming languages - WINSKEL - à la BU/LSV

- [Dow] Les démonstrations et les algorithmes - Introduction à la logique et à la calculabilité - Gilles DOWEK - à la BU ?
Très bien pour prendre du recul.
- [Aro] Computational Complexity : A Modern Approach - Arora BARAK - au LSV
En anglais. Les premiers chapitres couvrent les notions nécessaires et donnent souvent une bonne intuition. Manque parfois de formalisme, à croiser avec le PERIFEL.
- [Cor] Logique mathématique Tome 1/2 - René CORI, Daniel LASCAR - pas en BU
Typo assez ancienne, pas toujours agréable à lire.
- [Car] Langages formels, calculabilité et complexité - CARTON - à la BU/LSV
Très bonne référence couvrant beaucoup de bases. Se méfier de certaines preuves faites un peu rapidement.

2 Dev

- Minimisation de NÉRODE - ([Car], *Sec 1.7 p.49*) - 909
- Immerman Szelepcsényi - ([Per], *p. 121*) - 915
Implique de maîtriser parfaitement NL.
- Interpolation - ([Da],[Go],) - 918
Selon le système de preuve.
- Elimination des quantificateurs dans la théorie des ordres linéaires - (,) - 914,924
- Automate des occurrences - ([Cor], *p.886*) - 907,909,927
Tiens bien en 15 min, mais attention à bien maîtriser les petits calculs. Extension possible vers KMP.
- Théorème de KLEENE - ([Car], *Thm 1.59 p.36*) - 907,909,923
Tout faire est ambitieux, mais cela passe si on prend les constructions les plus basiques (THOMPSON et McNAUGHTON-YAMADA). Si on fait par contre ANTIMIROV, cela peut suffire.
- Complétude d'un système de preuve - ([Da],[Go],) - 918
Faire la complétude du système présenté. Il y a plusieurs preuve pour chaque système.
- Théorème de COOK-LEVIN - ([Car], *p. 191*) - 913,915,916,928
Preuve que SAT est NP-complet. Aller jusqu'à 3-SAT est ambitieux. Bien comprendre la notion de localité du calcul d'une machine de TURING.
- Complexité du tri rapide - ([Cor,Bea],) - 903,926,931
Bien faire attention au proba du Cormen, aller voir Beauquier est pertinent. Avoir une idée de l'écart type des performances.
- Problèmes indécidables pour les grammaires algébriques. - ([Car],) - 914,923
Ambiguïté, universalité. Insister sur la réduction, pas sur la notion de grammaire.
- Une fonction TURING calculable est μ -recursive. - ([Wol],) - 912,913
Preuve précise mais non pédagogique dans le Cori, claire mais non précise dans le Wolper...
- Preuve de correction d'un algorithme - ([Cor,Beau],) - 927, ?
Dijkstra, KMP, unification
- Équivalence entre deux variantes des machines de Turing - ([Sip], p136,) - 912,913
- Calcul de la sémantique d'un ou plusieurs programmes - (?,) - 930
Dans l'idéal, on montre que deux programmes sont équivalents, via des calculs de sémantiques, potentiellement dans des sémantiques différentes.
- Un exemple d'analyse d'un langage jouet - ([Aho], *Ch4.4 ou Ch 4.9.1 p298*) - 923
Le langage d'une calculatrice minimaliste est un classique. L'analyse d'une grammaire LL(1) est plus difficile. Ne pas hésiter à faire des dessins, avec des arbres d'interprétation possible.
- Indécidabilité de l'arithmétique de PEANO - (,) - 914,924
Passer par l'encodage des fonctions calculables. Assez long si on fait tout.
- Équivalence des sémantiques opérationnelle et dénotationnelle - ([Win], *p. 61*) - 930
On peut aussi faire l'équivalence entre petit-pas et grand-pas, mais moins intéressante.
- Complexité et correction du tri par tas - ([Cor], *3rd edition, p.154*) - 901,903
Simple dans le fond, mais à bien faire formellement, et pédagogiquement. Dessins et exemples bienvenue.
- Décidabilité de l'arithmétique de PRESBURGER - ([Car], *Thm 3.63 p.164*) - 909,914,924
Idée générale simple, mais attention aux détails. Réfléchir au codage, à sa sémantique, et à la complexité globale de la construction.
- Montre que X est ou n'est pas récursive primitive - ([Cori],[Car],) - 912
X pour ACKERMANN, avec une preuve un peu compliqué dans le Cori, ou X = isPrime dans le Carton. Attention, ACKERMANN est casse-gueule, il y a plein de lemmes auxiliaires souvent oubliés (e.g, monotonicté de la fonction considérée).

- Preuve de la factorielle en HOARE - ([Win], *ch6.6 p.93*) - 927
Peut être l'occasion de parler des problèmes d'automatisation.
- Borne inférieure sur la complexité d'un tri par comparaison - ([Cor],) - 903,926
Peut être un peu court.
- Hachage Parfait - ([Cor], *p. 258*) - 901,921
- Correction des algorithmes de Prim et Kruskal - (Cor, *p. ? ?*) - 925,927
La preuve du Cormen est général pour ce type d'algorithme.
- Problème de l'arrêt et théorème de Rice. - ([Car],) - 914
Facile, donc on prêtera attention à la propreté de la rédaction.
- Tri bitonique - ([Cor],) - 903
Attention, disponible uniquement dans la seconde édition
- Arbre Binaire de Recherche optimaux - ([Cor], *p. 397*) - 901,921
- Lemme de l'étoile et variantes - ([Car], *Sec 1.9 p.61*) - 909
- Plus longue sous séquence commune - ([Cor], *Ch15,4 p341*) - 907,931
Preuve et exemple. Avoir en tête des applications (séquence d'ADN, commande diff)
- Algorithme de Cocke-Kasami-Younger - ([Hopcroft, Ullman], *Ch7.4.4, p298*) - 923
Uniquement esquissé dans le Carton. Maîtriser la mise en forme normale de Chomsky.
- Correction totale et/ou complexité de DIJKSTRA - (Cor, Beau, *p. ?*) - 925,926,927
Long de faire correction et complexité, doit-être adapté selon la leçon.
- Equivalence entre fonctions récursives et fonctions représentables - ([Kri], *p. 24*) - 912,929
Il suffit de faire le sens récursif est représentable, mais il faut savoir justifier facilement de l'autre direction. Être bien au clair sur le problème des fonctions partielles et des problèmes de terminaison. Savoir exprimer le lambda terme correspondant à une fonction recursive, e.g l'addition, à partir du résultat.
- Indécidabilité de l'arrêt et applications à quelques problèmes indécidables - ([Sip], *p159,172,)* - 912,913
Ne faire Rice que si on sait bien le faire, qu'on le comprend, et qu'on sait l'appliquer.