Leçon 928 - Problèmes NP-complets : exemples et réductions

7 juin 2019

1 Extraits du Rapport

Rapport de jury 2018

L'objectif ne doit pas être de dresser un catalogue le plus exhaustif possible ; en revanche, pour chaque exemple, il est attendu que le candidat puisse au moins expliquer clairement le problème considéré, et indiquer de quel autre problème une réduction permet de prouver sa NP-complétude. Les exemples de réduction polynomiale seront autant que possible choisis dans des domaines variés : graphes, arithmétique, logique, etc. Si les dessins sont les bienvenus lors du développement, le jury attend une définition claire et concise de la fonction associant, à toute instance du premier problème, une instance du second ainsi que la preuve rigoureuse que cette fonction permet la réduction choisie et que les candidats sachent préciser comment sont représentées les données. Un exemple de problème NP-complet dans sa généralité qui devient P si on contraint davantage les hypothèses pourra être présenté, ou encore un algorithme P approximant un problème NP-complet.

2 Cœur de la leçon

- P et NP.
- Réduction polynomiale, complétude.
- Exemples de problèmes et de réductions. Théorème de Cook.

3 À savoir

- Acceptation par certificat.
- 3-SAT, 2-SAT, HORNSAT. Mise en forme normal (CNF, DNF).
- INDSET, COVER, Cycle Hamiltoniens, Voyageur de commerce.
- Somme sous ensemble, programmation entière.

4 Ouvertures possibles

- Algorithme d'approximation (par exemple sur le voyageur du commerce).
- Réduction d'un problème NP devenant P.
- SAT-solver.

5 Conseils au candidat

- Il **faut** être précis sur la définition de NP : ce n'est pas juste les machines non déterministes fonctionnant en temps polynomial. Pour une entrée, est-ce que la TM doit terminer pour tout chemin ? Est-ce qu'il faut juste qu'il existe un chemin qui termine ? Est-ce qu'il faut juste qu'il existe un chemin acceptant qui termine ?

- C'est dur d'éviter le côté catalogue. Essayer de varier les domaines, d'avoir des dessins et des exemples de problèmes. Le plus possible, donner précisément la définition des problèmes.
- Dans le développements, il faut absolument avoir des dessins et que tout soit formel.

6 Questions classiques

- Quelle est l'intuition derrière NP ?
- Pourquoi P=NPest une question centrale ?
- En quoi les problèmes NPsont difficile ?
- Telle et telle définitions d'acceptation pour une machine de TURING non déterministe sont-elles équivalentes ?
- Que se passe-t-il si on considère des réductions polynomial non déterministe ?
- Que se passe-t-il si on considère des réductions en espace logarithmique ?
- Justifier que NP est close par réduction polynomial.
- Connaissez-vous un SAT solver ? Utilisé en pratique pour quoi ?
- Pouvez-vous donner l'idée/les détails de cette réduction ?
- Montrer que

$$L = \left\{(M, x, 1^t) \mid \langle M, x\rangle \text{ termine en temps } \leq t, \text{pour } M \text{ non déterministe}\right\}$$

  est NP-complet.

7 Références

- [*Per*] Complexité algorithmique - PERIFEL - pas en BU
  La référence parfaite pour la complexité. Formel, clair. C'est essentiellement une traduction formel du ARORA.
- [*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. La NP-complétude de HamPATH est fausse dedans.
- [*Aro*] Computational Complexity : A Modern Approach - Arora BARAK - au LSV
  En anglais. Les premiers chapitres couvrent les notions nécéssaires et donnent souvent une bonne intuition. Manque parfois de formalisme, à croiser avec le PERIFEL.
- Éviter le PAPADIMITRIOU, qui contient de très nombreuses fausses preuves.

8 Dev

- 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.
- Preuve de NP-complétude d'un problème au choix.