

Leçon 915 – Classes de complexité. Exemples

9 février 2019

1 Extraits du Rapport

Rapport de jury 2017

Le jury attend que le candidat aborde à la fois la complexité en temps et en espace. Il faut naturellement exhiber des exemples de problèmes appartenant aux classes de complexité introduites, et montrer les relations d'inclusion existantes entre ces classes, en abordant le caractère strict ou non de ces inclusions. Le jury s'attend à ce que les notions de réduction polynomiale, de problème complet pour une classe, de robustesse d'une classe vis à vis des modèles de calcul soient abordées. Parler de décidabilité dans cette leçon serait hors sujet.

2 Coeur de la leçons

- La notion de complexité en espace et en temps avec les inclusions évidentes entre elles
- Le problème "classiquement complet"

$$L = \{(M, x, 1^t) \mid \langle M, x \rangle \text{ termine en temps (resp. espace)} \leq t\} \quad (1)$$

- Les classes usuelles (P, NP, PSPACE)
- Notion de réduction, de problème complet

3 À savoir

- La robustesse vis-à-vis du modèle de calcul (déterministe, non-déterministe, une bande, plusieurs bandes, écriture sur entrée)
- Savitch et les classes en espace
- Exemples de problèmes complets sur des domaines différents (graphes, automates, grammaires, logique, circuits)
- Les théorèmes de hiérarchie stricte
- Un petit diagramme qui permet de s'y retrouver en annexe avec les inclusions strictes et les égalités

4 Ouvertures possibles

- Les théorèmes de simulation pour les machines universelles avec des bornes fines de complexité
- Les classes alternantes, le lien avec EXSPACE et la hiérarchie polynomiale (Carton)
- Les classes NL , $co - NL$ et Immerman-Szelepcsényi
- Algorithmes randomisés comme autre modèle de calcul (tests de primalité, algorithme de Berlekamp, polynomial identity testing, RP, coRP)

- Des théorèmes de borne inférieure de simulation (une bande vs deux bandes en $O(n)$ vs $O(n^2)$ pour le langage des palindromes).
- Une machine qui calcule trop rapidement est un automate (Carton)
- Développer le côté logique (2SAT, HORNSAT, SAT, QBF, CTLSAT).
- Les arguments de padding peuvent faire de jolis résultats ($NSPACE = SPACE$ via Savitch, si $P = NP$ alors $EXP = NEXP$)
- S'il existe un problème NP -complet sur un langage unaire, alors $P = NP$.

5 Conseils au candidat

- Ne pas centrer toute la leçon sur P et NP
- Bien expliquer la notion de réduction polynomiale, et logarithmique si on parle de NL
- Si on parle de NL , bien formaliser le type de machines considérées
- Subtilités avec les classes complémentaires, problèmes à promesses

6 Questions classiques

- Quels sont les impacts de la variation du modèle sur la complexité?
- Pourquoi utiliser le formalisme des machines de Turing plutôt qu'un autre (λ -calcul, fonctions récursives)?
- Citer une classe qui n'est pas P , NP ou $PSPACE$
- Montrer que tel problème X est C -complet.
- Comment varie la complexité en fonction de la taille de l'alphabet

7 Références

- **TMP** Carton **TMP**
- **TMP** Arora Barak **TMP**

8 Dev

- **TMP** Hiérarchie en temps et en espace **TMP**
- **TMP** Immerman Szelepcsényi **TMP**
- **TMP** NP -complétude d'un problème **TMP**
- **TMP** 2SAT est NL -complet et en temps linéaire sur une machine RAM **TMP**
- **TMP** Universalité d'un langage rationnel est $PSPACE$ complet **TMP**