Leçon 916 – Formules du calcul propositionnel : Représentation, formes normales, satisifiabilité. Applications.

2 décembre 2019

1 Extraits du Rapport

Rapport de jury 2017

Le jury attend des candidats qu'ils abordent les questions de la complexité de la satisfiabilité. Pour autant, les applications ne sauraient se réduire à la réduction de problèmes NP-complets à SAT. Une partie significative du plan doit être consacrée à la représentation des formules et à leurs formes normales.

2 Cœur de la leçon

— Syntaxe et la sémantique des propositions, tables de vérité.

— Formes normales (CNF, DNF, BDD),

— Les méthodes syntaxiques de preuve. (déduction naturelle, calcul des séquents, résolution)

— Validité, satisfiabilité.

3 À savoir

— Complétude du calcul propositionnel.

— Les méthodes sémantiques de preuve (DPLL, tableau, table de vérité, BDD).

— Complexité ave SAT, 3SAT, HORNSAT, et 2SAT.

— Système de connecteurs, complétude (le XOR est complet). pour chacune la complexité des problèmes de décision, et du passage d'une forme à l'autre. (pour la validité ? pour la satisfiabilité ?)

4 Ouvertures possibles

— Le théorème de compacité.

— L'étude du fragment propositionnel via Herbrand.

— La résolution au premier ordre.

— Applications des SAT solvers (HAMPATH, Sudoku, coloration de graphes, crack de MD5). Attention les réductions sont *à l'envers* !

— Génération de circuits.

— Application de HORNSAT aux gestions de dépendances (analyse syntaxique, transformation d'automates et grammaires)

5 Conseils au candidat

— Ne pas centrer toute la leçon sur NP.

— Faire une grosse partie sur les formes normales, leurs avantages et leurs inconvénients (vitesse, conversion).

— Bien comparer les méthodes de résolution, leurs avantages et leurs inconvénients (vitesse, certificat).

— Différents livres peuvent avoir des notations différentes ($\neg T, \overline{T}$), rester cohérent.

— Ne pas oublier les applications.

6 Questions classiques

— Pourquoi fait-on tout en CNF alors que la DNF possède de meilleures propriétés ?

— En utilisant un SAT solveur, on doit écrire une traduction, quelle est la condition nécessaire et suffisante sur le problème pour que ce soit possible ?

— Quel est l'algorithme pour résoudre HORNSAT en temps linéaire ? Quelles applications ?

— Donner un exemple de formule pour laquelle le BDD est de taille exponentielle.

— Comparer la résolution, DPLL et les autres méthodes ?

— Combien de résolutions pour prouver $\bot$?

— Comment utiliser la résolution pour résoudre $2SAT$ en temps polynomial ?

7 Références

— [*Go*] Proof Theory and Automated Deduction - Jean Goubault-Larrecq - à la BU/LSV
Le must pour la logique

— [*Cori*] Logique mathématique Tome 1/2 - René Cori, Daniel Lascar - pas en BU
Typo assez ancienne, pas toujours agréable à lire.

— [*Da*] Introduction à la logique - R. David, K. Nour, C. Raffalli - à la BU/LSV
L'autre must pour la logique

— [*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.

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.

++ $2SAT$ est NL-Complet (ou algo en temps linéaire selon les leçons) - ([Car], [Pap], [Cor] (pour les algo de graphe)) - 915,916,925

++ Complétude de la résolution propositionnelle - ([Gou] ?) - 916

++ Formule CNF équisatisfiable - ([Per],Prop 3-Z p.77,[Gou]) - 915,916
À partir d'une formule, on produit une formule CNF équisatisfiable en temps linéaire. Attention, la mise en CNF est exponentielle (avoir un exemple).