PRIFYSGOL CYMRU; UNIVERSITY OF WALES

DEGREE EXAMINATIONS MAY/JUNE 2002

SWANSEA

Computer Science

CS 342 Constraint Satisfaction Problems and Applications

Attempt 2 questions out of 3

Time allowed: 2 hours

Students are permitted to use the dictionaries provided by the University

Students are NOT permitted to use calculators

CS_342
Constraint Satisfaction Problems and Applications

(Attempt 2 questions out of 3)

Question 1 Autarkies and 2-CNF

(a) Autarkies and their basic properties

(i) Explain in words what an *autarky* is.

(ii) Define formally $\mathbf{Auk}(\boldsymbol{F})$, the set of autarkies for a clause-set F, using a different formulation for autarkies than in part (i).

(iii) State the three main properties of autarkies (their relation to satisfying assignments, their "harmlessness" and about composition).

(iv) Prove two of the three main properties of autarkies from part (iii).

[9 marks]

(b) Unit-clause propagation

(i) Explain in words how $\boldsymbol{r_1} : \mathcal{CLS} \to \mathcal{CLS}$ is defined.

(ii) What are (precisely) the three essential properties of r_1, which guarantee that using r_1 is not too expensive, is "harmless", and can be explained by partial assignments?

[6 marks]

(c) Binary clause-sets

(i) How is $2\text{–}\mathcal{CLS}$ defined?

(ii) Consider $F \in 2\text{–}\mathcal{CLS}$, a variable v and $\varepsilon \in \{0, 1\}$, and assume $r_1(\langle v \to \varepsilon \rangle * F) \neq \{\bot\}$. Now the clauses of $r_1(\langle v \to \varepsilon \rangle * F)$ all have the same length — which ?! Why ?

(iii) Show that under the assumptions of part (ii) it holds that the clause-set $r_1(\langle v \to \varepsilon \rangle * F)$ is satisfiability equivalent to F (use parts (b)(ii) and (a)(iii)).

(iv) Give an algorithm in pseudo-code for deciding satisfiability of clause-set $F \in 2\text{–}\mathcal{CLS}$ in polynomial time. Explain, how it works, and why the running time is polynomially bounded.

[10 marks]

Question 2 Random formulas and phase transitions

(a) Applications of random formulas

(i) When testing a SAT solver, one needs benchmark formulas. Discuss the three categories of benchmark formulas, and their advantages and disadvantages.

(ii) Explain (briefly) the difference between "random" and "pseudo-random" formulas (or clause-sets). **[6 marks]**

(b) The variable clause-length model and the constant clause-length model for random clause-sets

(i) What parameters are used in the variable clause-length model, and what are their meanings?

(ii) What parameters are used in the constant clause-length model, and what are their meanings?

(iii) Define $\mathcal{CLS}(n, p, c)$ and the probabilities $P_0(n, p, c)$, $P_1(n, p, c)$ for unsatisfiability (P_0) resp. satisfiability (P_1). **[4 marks]**

(c) The notion of density

(i) Given a clause-set F, how are the relative densities $\rho_k(F)$ defined?

(ii) What is the density of formulas in $\mathcal{CLS}(n, p, c)$?

(iii) What can be said about P_0 and P_1 for clause-sets in $\mathcal{CLS}(n, p, c)$ with "low" resp. "high" density, and what is the (intuitive) reason for this behaviour? What do we know about the meaning of "low density" and "high density" here? (Do we know more for example for $p = 3$?) **[5 marks]**

(d) The Threshold Phenomenon

(i) Show how a typical graph for P_0, when fixing n and choosing $p = 3$, looks like when varying the density. (Do not forget to label the x-axis and the y-axis.)

(ii) Explain the Threshold conjecture in words.

(iii) State the Threshold Conjecture in a formal way.

(iv) Show a typical graph for running times of DLL-like algorithms for $p = 3$ in the constant clause-length model, when varying the density, and try to explain (intuitively) this graph. **[10 marks]**

Question 3 Clause-sets and DLL-like algorithms

(a) Clause-sets and partial assignments

(i) Define (within a theoretical framework) variables ($\mathcal{VA}$), literals ($\mathcal{LIT}$), clauses ($\mathcal{CL}$) and clause-sets ($\mathcal{CLS}$), and explain the meaning of these concepts.

(ii) Define partial assignments ($\mathcal{PASS}$), and explain the operation $* : \mathcal{PASS} \times \mathcal{CLS} \to \mathcal{CLS}$ of partial assignments on clause-sets; that is, explain how $\varphi * F$ is computed from F.

(iii) Define the set $\mathcal{SAT}$ of satisfiable clause-sets and the set $\mathcal{USAT}$ of unsatisfiable clause-sets.

(iv) Explain the composition $\varphi \circ \psi$ of partial assignments.

(v) State the basic properties of the composition of partial assignments and the operation of partial assignments on clause-sets.

[9 marks]

(b) DLL-like SAT algorithms

(i) Give pseudo-code for the basic DLL-like SAT algorithm.

(ii) Explain the different choices made in this algorithm scheme, and what would be a "good choice".

[6 marks]

(c) Datastructures and efficiency

(i) How could variables, literals, clauses, clause-sets and partial assignments be implemented? What are more efficient and what are less efficient choices?

(ii) Explain the difference between "value semantics" and "reference semantics" and their advantages and disadvantages w.r.t. possible implementations of clause-sets.

(iii) Explain how we can implement a DLL-solver using only linear space.

[10 marks]