6.0/4.0 VU Formale Methoden der Informatik

6 May 2016

Kennzahl (study id)	Matrikelnummer (student id)	Familienname (family name)	Vorname (first name)	Gruppe (version)

1.) Consider the following problems:

> **GROUPS**
>
> INSTANCE: A pair (P, D), where
>
> - $P = \{p_1, \ldots, p_n\}$ is a set of elements (intuitively, P is a set of persons), and
> - $D \subseteq P \times P$ is a binary relation over P (intuitively, $(p, p') \in D$ means that p dislikes p').
>
> QUESTION: Is it possible to divide the person of P into 5 groups such that none of the groups contains persons p, p' such that p dislikes p'? That is, does there exist a partitioning of P into sets $G_1, \ldots G_5$ such that $(p, p') \notin D$ holds for all $1 \leq i \leq 5$ and all pairs $p, p' \in D_i$?

> **GENERALIZED-COLORING (GC)**
>
> INSTANCE: A pair (G, k), where $G = (V, E)$ is a directed graph and k is an integer.
>
> QUESTION: Does there exist a k-coloring of G? That is, does there exist a coloring function $\mu : V \to \{1, \ldots, k\}$ such that $\mu(v) \neq \mu(v')$ for all edges $(v, v') \in E$?

Provide a polynomial time reduction from **GROUPS** to **GENERALIZED-COLORING**, and prove the correctness of your reduction.

(15 points)

2.) (a) Let $\varphi(x_1, \ldots, x_n)$ be a first-order formula with free variables $x_1, \ldots, x_n$ (all free variables of φ are in $x_1, \ldots, x_n$). Let $c_1, \ldots, c_n$ be new constants. Prove that $\varphi(x_1, \ldots, x_n)$ is valid if and only if $\varphi(c_1, \ldots, c_n)$ is valid. What can you say about the translation of quantifier-free first-order formulas to propositional formulas. **(9 points)**

(b) Consider the clauses $C_1, \ldots, C_5$ in `dimacs` format (in this order, shown in the box) which are given as input to a SAT solver. Apply CDCL to solve the CNF using the convention that if a variable is assigned as a decision, then it is assigned 'false'. Further, select variable 2 as the first decision variable that is assigned.

- Each time when a conflict occurs and after backtracking, draw the implication graph and indicate all UIPs and mark the first UIP. For each UIP, indicate the cut (i.e., a set of edges) and its asserting conflict clause. Learn the asserting conflict clause that corresponds to the first UIP.
- Is the given CNF satisfiable, unsatisfiable, or valid? Can the empty clause be derived from the given CNF during CDCL? Justify your answers to the above questions.

```
1 0
-1 10 0
-1 2 3 0
-3 -4 -10 0
-3 4 -10 0
```

(6 points)

3.) Let π be the program while $x \neq y$ do $x := x + 1;\ y := y + 2$ od. Show that the correctness assertion

$$\{ x = 2z \wedge y = z \wedge z > 0 \}\, x := x + y; \text{if } x > 0 \text{ then } \pi \text{ else abort fi}\, \{ x = y \}$$

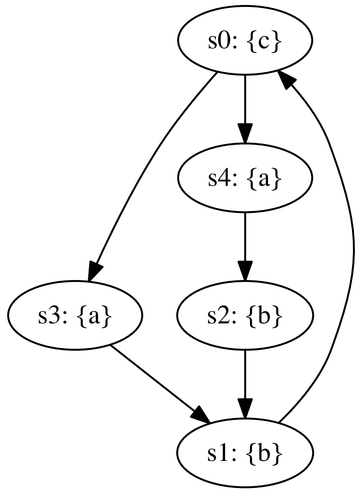
is true with respect to total correctness by using weakest preconditions.

The weakest precondition for some types of statements:
$\text{wp}(\text{if } e \text{ then } p \text{ else } q \text{ fi}, G) = (e \wedge \text{wp}(p, G)) \vee (\neg e \wedge \text{wp}(q, G))$
$\text{wp}(\text{while } e \text{ do } p \text{ od}, G) = \exists i\, (i \geq 0 \wedge F_i)$ where $F_0 = \neg e \wedge G$ and $F_{i+1} = e \wedge \text{wp}(p, F_i)$.

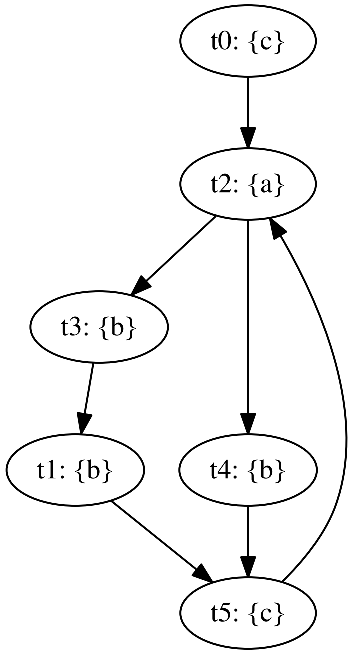
(15 points)

4.) (a) Provide a non-empty simulation relation H that witnesses $M_1 \leq M_2$, where M_1 and M_2 are shown below. The initial state of M_1 is s_0, the initial state of M_2 is t_0:

Kripke structure M_1:



Kripke structure M_2:



(5 points)

(b) For each of the following three equivalences, determine whether the equivalence holds. If so, prove correctness. Otherwise, disprove by giving a Kripke structure M and a path π such that (M, π) satisfies one side of the equivalence but not the other.

i. $\mathit{true}\, \mathbf{U}\, p \equiv \mathbf{F}\, p \wedge \neg \mathbf{G}\, \neg p$
ii. $p\, \mathbf{U}\, q \equiv p \wedge \mathbf{F}\, q$
iii. $\mathbf{GFGF}\, q \equiv \mathbf{GF}\, q$

(5 points)

(c) In the lecture we saw that we can use CBMC to solve NP-complete decision problems. In this exercise, you are to illustrate this approach for the *partition problem.*

The *partition problem* is the task of deciding whether a given set of N positive integers $S = \{i_1, i_2, \ldots, i_N\}$ can be partitioned into two sets S_1, S_2 such that the sum of the numbers in S_1 equals the sum of numbers in S_2. I.e., decide whether there are sets S_1, S_2 such that

- $S_1 \cup S_2 = S$,
- $S_1 \cap S_2 = \emptyset$, and
- $\sum_{i \in S_1} i = \sum_{j \in S_2} j$.

Write a C program that implements a *guess and check* routine for the partition problem and instrument the program with an appropriate CBMC assertion.

You may assume the following template:

```
int nondet_bool(); // nondeterministacally returns 0 or 1

// Fixed sample input:
int N = 8;                                        // size of the set
int values[] = { 4, 8, 15, 16, 23, 42, 11, 13 }; // elements in the set

int main() {
  // add code here:
  // 1. guess a solution
  // 2. put an assertion such that CBMC reports if there is a solution
}
```

(5 points)