6.0/4.0 VU Formale Methoden der Informatik (WS2010) **March 25, 2011**				
Kennzahl (study id)	Matrikelnummer (student id)	Familienname (family name)	Vorname (first name)	Gruppe (version)

1.) We provide next a reduction from **2-COLORABILITY** to **2-SAT**. Let $G = (V, E)$ be an arbitrary undirected graph (i.e. an arbitrary instance of **2-COLORABILITY**), where $V = \{v_1, \ldots, v_n\}$. For the reduction we use propositional variables $x_1, \ldots, x_n$. Then the instance φ_G of **2-SAT** resulting from G is defined as follows:

$$\varphi_G = \bigwedge_{[v_i,v_j] \in E} (x_i \vee x_j) \wedge (\neg x_i \vee \neg x_j).$$

Task: Prove the "⇒" direction in the proof of correctness of the reduction, i.e. prove the following statement: if G is a positive instance of **2-COLORABILITY**, then φ_G is a positive instance of **2-SAT**.

Note: For any property that you use in your proof, make it perfectly clear why this property holds (e.g., "by the problem reduction", "by the assumption X", "by the definition X", etc.)

(15 points)

2.) (a) Prove the formula $\varphi\colon ((x \to y) \to z) \to (x \to (y \to z))$ in the following steps:

(i) Compute $\hat{\delta}(\varphi)$. (Hint: It is allowed to avoid the labels for atoms; use the atoms instead.)

ii) Show the validity of

$$\bigwedge_{D \in \hat{\delta}(\varphi)} D \to \ell_\varphi$$

by resolution. If the formula is not valid, then provide a counter-example. **(4 points)**

(b) Given the following clauses, draw an implication graph starting with $\neg x_6$@1 followed by x_2@2 (if necessary).

$C_1\colon\ \neg x_1 \vee x_2$	$C_2\colon\ \neg x_1 \vee x_3 \vee x_5$	$C_3\colon\ \neg x_2 \vee x_4$	$C_4\colon\ \neg x_3 \vee \neg x_4$
$C_5\colon\ x_1 \vee x_5 \vee \neg x_2$	$C_6\colon\ x_2 \vee x_3$	$C_7\colon\ x_2 \vee \neg x_3$	$C_8\colon\ x_6 \vee \neg x_5$

Is the clause set unsatisfiable? If yes, then give a proof; if not, then provide a model. **(4 points)**

(c) Let $\varphi\colon \forall x \exists y \big[(s(x) = y) \wedge (y = s(x))\big]$, where $=/2$ is the equality predicate written in infix notation. Let T be a theory which contains the first-order equality axioms reflexivity, symmetry and transitivity. Show by purely semantical means that $T \models \varphi$ holds. (Hint: Show that $Mod(T) \subseteq Mod(\varphi)$.) **(7 points)**

3.) (a) We use $[x]$ to denote the function associated with the syntactic entity x, where x may be a program, an expression, or one of the pre-defined operators. Investigate for each of the three cases, whether $[x] = [y]$ implies $x = y$ for arbitrary programs/expressions/operators x and y. If yes, give an argument for it, if not, give a counterexample. Note that this are three separate questions. What about the converse: Does $x = y$ necessarily imply $[x] = [y]$? **(4 points)**

(b) Prove the total correctness of the assertion below. Describe the function computed by the program when considering n as the input and i as the output.

$\{\, \mathbf{1\colon n \geq 0} \,\}$
$i \leftarrow 0;$
$j \leftarrow 1;$
$k \leftarrow 1;$
$\{\, \boldsymbol{Inv\colon n \geq i^3 \wedge j = (i+1)^2 \wedge k = (i+1)^3} \,\}$
while $n \geq k$ do
    $i \leftarrow i + 1;$
    $k \leftarrow k + 3j + 3i + 1;$
    $j \leftarrow j + 2i + 1$
od
$\{\, \mathbf{2\colon i^3 \leq n < (i+1)^3} \,\}$

(11 points)

4.) (a) Find a Kripke structure K with initial state s_0 that has the properties $\mathbf{AGEF}p$ and $\mathbf{A}\,(\mathbf{GF}p \Rightarrow \mathbf{GF}q)$ at state s_0, but not $\mathbf{AG}(p \Rightarrow \mathbf{AF}q)$. Justify your choice. **(5 points)**

(b) Consider the following program, where the semantics of the statement $(x, y) := (a, b)$ is that the values a and b are simultaneously assigned to the variables x and y.

```
int x, y;

void foo() {
  (x, y) := (50, 49);

  while (x > y) {
   if (y == 0) {
     (x, y) := (x - 1, y);
   }
   else {
     (x, y) := (x - 1, y - 1);
   }
  }

  assert(y == 0);
}
```

i. Provide a labeled transition system for the given program.
ii. Provide an abstraction for the labeled transition system that uses the predicate $x > y$.
iii. Give an error trace in the abstraction.
iv. Introduce a new predicate to refine the abstraction to get rid of the error state. Give the new abstraction.

(4 points)

(c) Given a graph, write a C program such that CBMC can determine whether the given graph is 3-colorable. Augment the following given code corresponding to the following subtasks.

```
#define TRUE  1
#define FALSE 0

#define RED   0
#define GREEN 1
#define BLUE  2

#define N     4 // Number of nodes in the graph

int graph[N][N] = { { 0, 1, 0, 1 }, { 1, 0, 0, 0 }, ... };
int coloring[N];

int nondet_int();
```

i. Write a loop that nondeterministically guesses a coloring for the graph. A coloring assigns to every node of the given graph either the color red, green, or blue.

ii. Write a loop that checks whether the coloring assigns to every node in the graph a color that is different to the colors of its neighbors. Furthermore, ensure that CBMC reports a 3-coloring of the graph in case there exists one.

(6 points)