Universität Koblenz-Landau

FB 4 Informatik

Prof. Dr. Viorica Sofronie-Stokkermans
Dipl. Inf. Markus Bender

December 10, 2012

Exercises for "Decision Procedures for Verification"
Exercise sheet 9

Let $\Sigma = (\Omega, \Pi)$ such that $\Pi = \emptyset$ (i.e. $\approx$ is the unique predicate symbol).

For every ground formula $\phi := s_1 \approx s'_1 \wedge \cdots \wedge s_k \approx s'_k \wedge t_1 \not\approx t'_1 \wedge \cdots \wedge t_m \not\approx t'_m$
we denote by $FLAT(\phi)$ the formula obtained by replacing, in a bottom-up fashion, every term of the form $f(c_1, \ldots, c_n)$ with a new constant c_f.

The set of functional consistency axioms associated with the flattening is the set of clauses:

$FC = \{(c_1 \approx c'_1 \wedge \cdots \wedge c_n \approx c'_n) \rightarrow d \approx d' \mid d$ introduced as abbreviation for $f(c_1, \ldots, c_n)$
d' introduced as abbreviation for $f(c'_1, \ldots, c'_n)\}$

Example: If ϕ is $f(a, b) \approx a \wedge f(f(a, b), b) \not\approx a$
then $FLAT[\phi]$ is computed by introducing new constants renaming terms starting with f and then replacing the terms with the constants:

- $FLAT[\underbrace{f(a,b)}_{a_1} \approx a \wedge \underbrace{f(\underbrace{f(a,b)}_{a_1}, b)}_{a_2} \not\approx a] := a_1 \approx a \wedge a_2 \not\approx a$ Abbreviations : $f(a,b){=}a_1$
  $f(a_1,b){=}a_2$
- $FC := (a \approx a_1 \wedge b \approx b) \rightarrow a_1 \approx a_2$.

Exercise 9.1:
Let ϕ be the following (ground) formula:

$$f(f(c)) \approx f(c) \wedge f(f(c)) \approx f(d) \wedge d \not\approx f(c).$$

- Compute $FLAT(\phi)$ (the formula obtained by recursively replacing, in a bottom-up fashion, any term of the form $f(c')$, where c' is a constant, with a new constant).
- Compute the set FC of functional consistency axioms associated with the flattening in $FLAT(\phi)$:

$$FC = \{c_1 \approx c_2 \rightarrow d_1 \approx d_2 \mid d_i \text{ is introduced as an abbreviation for } f(c_i)\}.$$

Exercise 9.2:
Let $\overline{c} = (c_1, \ldots, c_n)$ be a sequence of constants. We write $s(\overline{c})$ in order to emphasize that the term s contains some of the constants in $\overline{c}$.

Prove that the following are equivalent:

(1) $(\bigwedge_i s_i(\overline{c}) \approx t_i(\overline{c})) \wedge \bigwedge_j s'_j(\overline{c}) \not\approx t'_j(\overline{c})$ is satisfiable
(2) $FC \wedge FLAT[(\bigwedge_i s_i(\overline{c}) \approx t_i(\overline{c})) \wedge \bigwedge_j s'_j(\overline{c}) \not\approx t'_j(\overline{c})]$ is satisfiable

Exercise 9.3:
Consider the fragments of first-order logic studied in the class, namely the Bernays-Schönfinkel class and the Ackermann class. To which of these classes do the following formulae belong (note that they can be in more than one, or in none of the classes above):

(1) $\exists y_1 \exists y_2 \forall x_1 \forall x_2 \ \ ((p(x_1, y_1) \vee r(x_2, y_1)) \wedge q(y_2))$
(2) $\forall x \exists y \forall z \exists u \ \ ((p(x) \vee q(y)) \wedge (q(y) \vee p(u))$
(3) $\exists y \forall x \exists z \ \ ((r(x, y) \vee r(y, z)) \wedge q(z) \wedge r(y, z))$
(4) $\exists z \forall x \forall y \exists z' \ \ ((r(x, y) \vee r(y, z)) \wedge s(z, y, z'))$

Supplementary exercise:

Exercise 9.4:
Let $\Sigma = (\{c_1/0, \dots, c_n/0, f_1/1, \dots, f_n/1\}, \Pi)$. Consider the following classes of Σ-clauses:

- G (denoted in the lecture also $G(c_1, \dots, c_n)$) is the class of all ground clauses in the signature Σ which do not contain any occurrence of a unary function symbol.
- V (denoted in the lecture also $V(x, c_1, \dots, c_n)$) is the class of all clauses with one variable (x) in the signature Σ which do not contain any occurrence of a unary function symbol.

- G_f (denoted in the lecture also $G(c_1, \dots, c_n, f_k(c_j))$) is the class of ground clauses in the signature Σ which contain at least one occurrence of a unary function symbol (having as argument a constant); no nested applications of unary function symbols are allowed.

  Example: Assume $p/3.q/2 \in \Pi$ Then:
  $C_1 : p(c_1, c_2, c_3) \vee \neg q(c_2, c_1)$ is in G but is not in G_f
  $C_2 : q(c_1, c_2) \vee \neg q(f_1(c_3), c_4) \in G_f$
  $C_3 : q(c_1, c_2) \vee \neg q(f_1(c_3), f_2(f_3(c_4))) \notin G_f$.

- V_f (denoted in the lecture also $V(x, c_1, \dots, c_n, f_j(x))$) is the class of all clauses in the signature Σ which contain only one variable (x), at least one occurrence of a unary function symbol (having as argument the variable x), no occurrences of terms of the form $f_k(c_j)$; in addition no nested applications of unary function symbols are allowed.

  Example: Assume $p/3.q/2 \in \Pi$ Then:
  $C'_1 : p(x, c_2, x) \vee \neg q(c_2, c_1)$ is in V but is not in V_f
  $C'_2 : q(x, c_2) \vee \neg q(f_1(c_3), x) \notin V_f$
  $C'_3 : q(c_1, x) \vee \neg q(x, f_2(f_3(x))) \notin V_f$ $C'_4 : p(x, c_2, x) \vee \neg p(c_2, x, f(x)) \in V_f$.

Consider a term ordering $\succ$ in which $f(t) \succ t$ for every term t and terms containing function symbols of arity 1 are larger than those who do not. Consider the general ordered resolution calculus $\mathsf{Res}^{\succ}$. Prove that in this calculus:

(1) The resolvent of a clause in G_f and a clause in V_f is a clause in G_f of G.
(2) The resolvent of two clauses in V_f is a clause in G, G_f, V or V_f.

Please submit your solution until Monday, December 17, 2012 at 9:00. Joint solutions prepared by up to three persons are allowed. Please do not forget to write your name on your solution. Submission possibilities:

- By e-mail to `mbender@uni-koblenz.de` with the keyword "Homework DP" in the subject.
- Put it in the box in front of Room B 222.