Übungen zur Vorlesung
Entscheidungsverfahren mit
Anwendungen in der Softwareverifikation

Wintersemester 2018/19

Aufgabenblatt 4

Institut für Theoretische Informatik, Karlsruher Institut für Technologie (KIT)
Prof. Dr. Carsten Sinz

20.12.2018
Abgabe: 08.01.2019

Aufgabe 11 (Bit-Vektor-Logik, Arithmetik in $\mathbb{Z}/2^k\mathbb{Z}$) [2+3 Punkte]

Sei $0 < k \in \mathbb{N}$, $U = \mathbb{Z}/2^k\mathbb{Z}$ und $x \in U$.

a) Für welche k und $a, b \in U$ besitzt die lineare Gleichung $a \cdot x = b$ eine Lösung $x \in U$?

b) Sei nun $k = 4$. Bestimmen Sie für alle $a \in U$ die Lösungen der quadratischen Gleichung $x \cdot x = a$. Gibt es einen Zusammenhang zu den Lösungen für $k = 3$?

Aufgabe 12 (Bit-Vektor-Logik, Multiplikation) [10 Punkte + 5 Punkte optional]

Abgabe der Programmieraufgabe bis 15.01.2019 möglich.

a) Implementieren Sie einen Bit-Blaster für die Multiplikationsoperation $\texttt{mul} : \mathrm{BV}(n) \times \mathrm{BV}(n) \to \mathrm{BV}(2 \cdot n)$, wobei $\mathrm{BV}(n)$ die Menge der Bit-Vektoren der Länge n bezeichnet. Die Eingabe Ihres Programms sollen zwei Zahlen a, b mit $0 \leq a, b < 2^n$ sein, deren Produkt bestimmt werden soll. Die Ausgabe bestehe aus einer aussagenlogischen Formel im DIMACS-Format. Prüfen Sie Ihr Programm mit Hilfe eines SAT-Solvers (z.B. Minisat), indem Sie sich das Modell ausgeben lassen.

b) [optional] Erweitern Sie Ihren Bit-Blaster aus Teil a) so, dass er für eine Eingabe a mit $1 \leq a < 2^{2n}$ eine aussagenlogische Formel generiert, die genau dann erfüllbar ist, wenn a keine Primzahl ist.

Aufgabe 13 (Array-Logik mit λ-Termen) [5+5 Punkte, optional]

Die Logik $T_{\lambda A}$ erweitert das quantorenfreie Fragment T_A^{QFF} der Theorie der Arrays um einen Array-Konstruktor, $\lambda i \,.\, t(i)$, wobei i eine Variable der Indextheorie und $t(i)$ ein Term der Elementtheorie ist. Für das Array $a = \lambda i \,.\, t(i)$ gilt $\texttt{read}(a, i) = t(i)$ für alle i.

a) Unter der Annahme, dass auf der Indextheorie nur die Gleichheit definiert ist, geben Sie ein Entscheidungsverfahren für die Logik $T_{\lambda A}$ an.

b) Wie lässt sich in $T_{\lambda A}$ die Semantik der C Standard-Bibliotheksfunktionen `memcpy` und `memset` definieren? (Der Hauptspeicher soll dabei als Byte-Array aufgefasst werden.)

Aufgabe 14 (Array-Logik) [5 Punkte]

Vereinfachen Sie den Array-Logik-Ausdruck

$$\begin{aligned} y &= 5 \\ m_1 &= write(m_0, p_1, 7) \\ m_2 &= write(m_1, p_2 + 4, 5) \\ x &= read(m_2, p_1) \\ m_3 &= write(m_2, p_2, x) \\ m_4 &= write(m_2, p_2, y) \\ m_5 &= (c = 0 \,?\, m_3 : m_4) \\ 5 &\neq read(m_5, p_2) \end{aligned}$$

(zu verstehen als Konjunktion der einzelnen Zeilen) so weit wie möglich, indem Sie das McCarthy-Axiom

$$read(write(m, p, x), q) \;\rightarrow\; (p = q \,?\, x : read(m, q))$$

als Ersetzungsregel anwenden und die entstehenden Ausdrücke vereinfachen (Gleichheit, lineare Arithmetik). Ist die Gleichungsmenge erfüllbar? Falls ja, geben Sie eine erfüllende Belegung an.

Der ternäre ?-Operator ist wie in den Programmiersprachen C/C++/Java zu verstehen, das heißt $a = (c \,?\, x : y)$ steht für $(c \Rightarrow a = x) \wedge (\neg c \Rightarrow a = y)$.