Lehrstuhl für Software and Computational Systems
Prof. Dr. Gidon Ernst, Marian Lingsch-Rosenfeld

Übungen zur Vorlesung
Formale Spezifikation und Verifikation

Wintersemester 2024/25

Übungsblatt 10

Bekanntgabe am 27.01.2025

Auf diesem Übungsblatt sollen Sie Eigenschaften mit Quantoren über Zahlen und Arrays formalisieren.

Auf diesem Blatt sollen Sie *keine Beweise* führen.

1 Eigenschaften über Zahlen

Die Sorte Int bezeichnet die ganzen Zahlen. Sie haben uneingeschränkt die mathematischen Operatoren $+$, $*$, usw zur Verfügung.

Aufgabe

- Für die Funktion
$$even\colon \text{Int} \to \text{Bool}$$
definieren Sie ein Axiom (d.h. eine Allquantifizierte Formel), die besagt, dass das Ergebnis von $even(x)$ immer angibt ob x gerade ist.

  Eine gerade Zahl lässt sich darstellen als das Vielfache von 2 und einer weiteren ganzen Zahl.

  Lösung: $\forall x.\, even(x) \iff (\exists y.\, 2 * y = x)$

  Kommentare und Varianten

  - Sorten (= Typen) der quantifizierten Variablen explizit:
  $$\forall x\colon \text{Int}.\, even(x) \iff (\exists y\colon \text{Int}.\, 2 * y = x)$$
  - Statt $\iff$ können wir hier auch $=$ schreiben
  - Teilbarkeit durch 2: $\exists y.\, y + y = x$
  - Teilbarkeit durch 2: $((x/2) * 2) = x$ wenn $/$ eine Ganzzahldivision mit Abschneiden des Rests bezeichnet (`div` in SMT-LIB).
- Definieren Sie mit Hilfe von *even* eine Funktion
$$odd\colon \text{Int} \to \text{Bool}$$
sodass $odd(x)$ genau für die ungeraden Zahlen gilt (nicht zu kompliziert denken!)

  Lösung: $\forall x.\, odd(x) \iff \neg even(x)$

  Variante: $\forall x.\, odd(x) \iff even(x+1)$ (oder mit -1)

2 Eigenschaften über Arrays

Die (SMT-LIB) Theorie der funktionalen Arrays wie in der Vorlesung besprochen besteht aus

- Sorten: $\text{Array}\langle s, s'\rangle$ für alle Sorten s, s'
- Funktionen: $\_[\_]: \quad \text{Array}\langle s, s'\rangle \times s \to s'$
  $\_[\_ := \_]: \quad \text{Array}\langle s, s'\rangle \times s \times s' \to \text{Array}\langle s, s'\rangle$

Zur Erinnerung, es gelten folgende Eigenschaften. Diese benötigen Sie nicht *direkt* in Ihrer Formalisierung, aber es hilft, die Bedeutung der Funktionen zu erklären.

- Read over Write $\quad \forall\, k, k', v.\ a[k := v][k'] = \begin{cases} v, & \text{if } k = k' \\ a[k'], & \text{if } k \neq k' \end{cases}$
- Extensionalität $\quad (\forall\, k.\ a[k] = b[k]) \implies a = b$

Aufgaben

- Folgende Funktion
  $$const\colon s' \to \text{Array}\langle s, s'\rangle$$
  soll ein Array bezeichnen bei denen für alle Schlüssel ein angegebener Wert herauskommt. Die Funktion initialisiert also ein Array in dem alle Werte gleich sind.
  - Schreiben Sie eine Formel, die bezeichnet, dass wenn a gleich $const(7)$ ist, dann ist $a[9]$ dieser Wert 7. In diesem Fall, was sind dann die konkreten Sorten s und s'?

    Lösung: $a = const(7) \implies a[9] = 7$
  - Definieren Sie eine Eigenschaft die besagt, dass ganz allgemein für $a = const(c)$ bei jedem Index der richtige Wert herauskommt.

    Lösung: $\forall a, k, c.\ a = const(c) \implies a[k] = c$ direkte Formalisierung der Angabe

    Variante: $\forall k, c.\ const(c)[k] = c$ logisch äquivalent
- Formalisieren Sie dass für ein gegebenes Array $a\colon \text{Array}\langle \text{Int}, \text{Int}\rangle$
  - Gerade Schlüssel immer gerade Werte enthalten.

    Lösung: $\forall k.\ even(k) \implies even(a[k])$ (wobei k ein Int bezeichnet)
  - Der Wert an allen Schlüsseln ist immer größer als der Schlüssel selbst

    Lösung: $\forall k.\ a[k] > k$
  - Jeder gerade im Array enthaltene Wert ist an einem ungeraden Schlüssel gespeichert

    Hier ist die Aufgabenstellung nicht eindeutig—Anforderungen in natürlicher Sprache sind manchmal unpräzise und lassen unterschiedliche Interpretationen zu.

    Hier ist die Uneindeutigkeit falls ein gerader Wert c mehrfach vorkommt, d.h. ob wir für $a[k_1] = c$ und $a[k_2] = c$ fordern, dass k_1 und k_2 *beide* ungerade sind oder zumindest einer davon.

    Lösung 1: Wenn ein Wert c an einem Schlüssel k gespeichert ist, und dieser Wert tatsächlich gerade ist, dann muss *dieser* Schlüssel k ungerade sein. Anders gesprochen, gerade Werte dürfen nur an ungeraden Indizes vorkommen.

    $$\forall k, c.\ a[k] = c \wedge even(c) \implies odd(k)$$

logisch Äquivalent, da $A \wedge B \implies C$ genau dann wenn $A \implies (B \implies C)$

$$\forall k, c.\ a[k] = c \implies \big(even(c) \implies odd(k)\big)$$

oder auch in anderer Reihenfolge, da $A \wedge B$ genau dann wenn $B \wedge A$

$$\forall k, c.\ even(c) \implies \big(a[k] = c \implies odd(k)\big)$$

Lösung 2: Jeder im Array vorkommende gerade Wert c muss zumindest *irgendwo* auch an einem (möglicherweise anderen) ungeraden Index gespeichert sein.

$$\forall c.\ even(c) \wedge \big(\exists k_1.\ a[k_1] = c\big) \implies \big(\exists k_2.\ odd(k_2) \wedge a[k_2] = c\big)$$

Beide Lösungen sind korrekte Antworten auf die Aufgabenstellung. Um zu entscheiden, welche Variante in einer konkreten Anwendung korrekt ist, muss man die Anforderungen genauer kennen.

Zwei Beispiele, die sich ungefähr an den obigen Mustern orientieren (aber nicht unbedingt genau in die gegebene Formalisierung passen):

Beispiel 1 (Datenbank Index): Wenn ein Tupel in der Datenbank gespeichert ist, dann muss es von seinem Primärschlüssel aus genau dort gefunden werden.

Beispiel 2 (Java Garbage Collection): Wenn ein nicht zur Löschung markiertes Objekt c von einem anderen Objekt k_1 aus direkt referenziert wird, dann muss c auch von einem *Wurzelobjekt* k_2 über möglicherweise mehrere Schritte erreichbar sein.

Nutzen Sie *even* und *odd* wie oben beschrieben.

- Arrays der Form $a\colon \text{Array}\langle \text{Int}, \text{Bool}\rangle$ können zur Charakterisierung von Mengen über Int verwendet werden, dabei entspricht $a[n]$ dem Test ob n in der von a repräsentierten Menge enthalten ist (d.h. statt $n \in a$ schreiben wir $a[n]$).

  - Wie lässt sich die leere Menge mit Hilfe von *const* als ein Term schreiben?

    Lösung: $const(false)$ repräsentiert die leere Menge $\emptyset$, denn

    $const(false)[n]$ genau dann wenn $false$ (per Definition von *const* aus voriger Teilaufgabe) und dass wiederum genau dann wenn $n \in \emptyset$, für alle n.

    "$const(false)[n]$ gilt für kein n" entspricht "die leere Menge enthält keine Elemente".

  - Bestimmen Sie einen Term gegeben a ein Array angibt, welches alle Elemente von a repräsentiert außer die 7. Nutzen Sie dazu das Array Update $a[\_ := \_]$.

    Lösung: $a[7 := false]$,

    denn $a[7 := false][7]$ ist $false$ ("7 ist nicht drin") und $a[7 := false][n]$ für $n \neq 7$ ist $a[n]$, "testen ob es überhaupt drin war").

  - Bestimmen Sie eine Formel: a repräsentiert die Vereinigung der Mengen, die durch b und c repräsentiert werden

    Lösung: $\forall n.\ \big(a[n] \iff (b[n] \vee c[n])\big)$

    "jedes Element n von a ist entweder in b oder in c, und umgekehrt."

Allgemeine Hinweise:

- Die vorgestellte Lösung ist lediglich ein Lösungsvorschlag. Es wird kein Anspruch auf Vollständigkeit oder Korrektheit erhoben.