Technische Universität Braunschweig

TCS

Prof. Dr. Roland Meyer, Dr. Jürgen Koslowski

Einführung in die Logik

Aufgabenblatt 1, 2018-04-16

Präsenzaufgabe 1

Zur Erinnerung: Für eine Formelmenge $\Sigma \subseteq \mathcal{F}$ setzt man

$$\boldsymbol{Folg}(\Sigma) := \{\, A \in \mathcal{F} \,:\, \Sigma \models A \,\}$$

Damit ist ***Folg*** eine Funktion auf der Potenzmenge von $\boldsymbol{F}$, geschrieben $\boldsymbol{P}(\mathcal{F}) \xrightarrow{\boldsymbol{Folg}} \boldsymbol{P}(\mathcal{F})$.

Beweisen Sie mit den Methoden der zweiten VL (und so effizient wie möglich), dass die ***Folg***-Funktion *idempotent* ist, d.h., $\boldsymbol{Folg}(\boldsymbol{Folg}(\Sigma)) = \boldsymbol{Folg}(\Sigma)$.

Lösungsvorschlag:

Die Gleichheit zweier Mengen X und Y weist man meistens nach, indem man beide Inklusionen $X \subseteq Y$ und $Y \subseteq X$ zeigt.

$\supseteq$: In der zweiten VL wurde die *Extensivität* der ***Folg***-Funktion beweisen, d.h., $\Sigma \subseteq \boldsymbol{Folg}(\Sigma)$ für jede Menge $\Sigma \subseteq \mathcal{F}$. Ersetzt man in dieser Aussage die Menge Σ durch $\boldsymbol{Folg}(\Sigma)$, so ergibt sich $\boldsymbol{Folg}(\Sigma) \subseteq \boldsymbol{Folg}(\boldsymbol{Folg}(\Sigma))$, oder anders ausgedrückt $\boldsymbol{Folg}(\boldsymbol{Folg}(\Sigma)) \supseteq \boldsymbol{Folg}(\Sigma)$.

$\subseteq$: Wähle $A \in \boldsymbol{Folg}(\boldsymbol{Folg}(\Sigma))$ und eine für $\Sigma \cup \{A\}$ passende Belegung, die Σ erfüllt. Zu zeigen: φ erfüllt auch A.

Wir wissen á priori nicht, ob alle atomaren Aussagen in A auch in einer der Formeln von Σ vorkommen. Aber falls $B \in \boldsymbol{Folg}(\Sigma)$, dann gilt für jede atomare Aussage p auch $B \wedge \{p \vee \neg p) \in \boldsymbol{Folg}(\Sigma)$. Damit kommt jede atomare Aussage in einer der Formeln von $\boldsymbol{Folg}(\Sigma)$ vor.

Falls also p in keiner Formel von Σ vorkommt, können wir jede Σ erfüllende Belegung ψ, die auf p nicht definiert ist, auf zwei Weisen zu einer auch für p passenden Belegung $\bar{\psi}$ erweitern: indem wir $\bar{\psi}(p) = 0$ bzw. $\bar{\psi}(p) = 1$ setzen. Nach Voraussetzung erfüllt jede dieser Erweiterungen weiterhin Σ. Mit anderen Worten: der Wert einer Σ erfüllenden Belegung auf atomaren Aussagen, die nirgends in Σ vorkommen, ist für Aussagen in $\boldsymbol{Folg}(\Sigma)$ irrelevant, aber damit auch für Aussagen in $\boldsymbol{Folg}(\boldsymbol{Folg}(\Sigma))$.

Insbesondere können wir die gegebene Σ erfüllende Belegung φ beliebig zu einer totalen $\boldsymbol{Folg}(\Sigma)$ erfüllenden Belegung $\bar{\varphi}$ erweitern. Diese erfüllt nach Voraussetzung A, also gilt das auch für die ursprüngliche Belegung φ, da diese für A paßte.

Präsenzaufgabe 2

[BMC für While-Programme]

Wir betrachten eine imperative Programmiersprache mit folgenden Eigenschaften:

- Es gibt nur Boole'sche Variablen.
- While-Blöcke der Form **While** A **do**...**end while** können gebildet werden, wobei A eine aussagenlogische Formel ist.
- Anweisungen der Form **assert** A sind erlaubt.

Die **assert**-Anweisungen haben keinen Effekt, schlagen aber fehl, wenn die Formel A nicht

erfüllt ist. Beispiel:

```
1  x := 1
2  y := 1
3  z := 0
4  while x ∨ y do
5      x := ¬y
6      z := x ∧ z
7  end while
8  assert z
```

Beschreiben Sie, analog zum BMC für Schaltwerke, ein Verfahren, das prüft, ob in einem gegebenen Programm eine bestimmte ***assert***-Anweisung fehlschlägt.

Lösungsvorschlag:

Idee: Entfaltung der While-Schleifen mit Hilfe von If-Konstrukten; die Schranke für die Tiefe der Entfaltung wird realisiert, indem man nach k vielen If-Konstrukten explizit die zu prüfende Bedingung auf „falsch“ setzt.

```
1  x := 1
2  y := 1
3  z := 0
4  if x ∨ y then
5      x := ¬y
6      y := x ∧ z
7      while x ∨ y do
8          x := ¬y
9          z := x ∧ z
10     end while
11 end if
12 assert z
```

...

```
1  x := 1
2  y := 1
3  z := 0
4  if x ∨ y then
5      x := ¬y
6      z := x ∧ z
7      if x ∨ y then
8          x := ¬y
9          z := x ∧ z
10         if x ∨ y then
11             ...
12         end if
13     end if
14 end if
15 z := ¬z
16 assert z
```

Beim Aufstellen der zugehörigen aussagenlogischen Formel ist zu beachten, dass die If-Konstruktion der Implikation $\rightarrow$ entspricht, die in den Schaltwerken nicht vorkam. Andererseits sind die Zuweisungen $:=$ durch die Äquivalenz $\leftrightarrow$ auszudrücken.

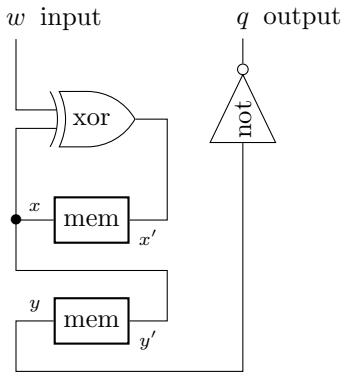
Verschiedene While-Schleifen können mit unterschiedlichen Schranken entfaltet werden (etwa bei geschachtelten While-Schleifen).

Hausaufgabe 3 [15 PUNKTE]

[Bounded Model Checking für Schaltwerke]

Betrachten Sie das durch das folgende Diagramm gegebene Schaltwerk. Hier sollen beide

Speicherzellen mit 0 initialisiert sein.



Prüfen Sie mittels iterativem BMC, ob die Bedingung $q = 1$ in diesem System immer gegeben ist. Falls nicht, geben Sie einen Ablauf und Eingaben an, die die Bedingung verletzen. [Hinweis: Mehr als vier Iterationen sollten Sie nicht benötigen.]

Hausaufgabe 4 [10 PUNKTE]

Behauptung: es gibt unendlich viele Primzahlen

(1) Führen Sie zunächst die Annahme, es gäbe nur endlich viele Primzahlen zu einem Widerspruch.

(2) In der 2. VL haben Sie das allgemeine Induktionsprinzip für rekursiv aufgebaute Mengen kennengelernt. Wenden Sie dieses Prinzip an um zu zeigen, dass es unendlich viele Primzahlen gibt. Formen Sie zunächst die Behauptung geeignet um, so dass das Induktionsprinzip anwendbar wird.

Hausaufgabe 5 [5 PUNKTE]

Finden Sie den Fehler in folgendem Induktionsargument:

Behauptung: Alle Katzen sind gleich gefärbt.

(Da wir z.B. eine weiße Katze haben mit dunkel geringeltem Schwanz und dunklen Flecken im Gesicht, müßten alle Katzen so aussehen, was natürlich nicht stimmen kann.)

„Beweis“: Der Fall $n = 1$ ist klar, jede einzelne Katze ist so gefärbt wie sie selbst.

Wir nehmen an, dass jeweils $k \geq 1$ Katzen gleich gefärbt sind (Induktionsvoraussetzung).

Um dasselbe für $k + 1$ Katzen zu zeigen, reihen wir sie von links nach rechts auf. Die linken k Katzen sind nach Induktionsvoraussetzung gleich gefärbt, ebenso die rechten k Katzen. Aber damit ist die linke Katze genauso gefärbt wie Ihre Nachbarin, und diese wie die rechte Katze. Damit sind haben alle $k + 1$ Katzen gleich gefärbt.

Abgabe bis Montag, 2018-04-23, 13:15, im Kasten neben IZ 343