Prof. Martin Hofmann
Dr. Ulrich Schöpp
Stephan Barth

Ludwig-Maximilians-Universität München
Institut für Informatik
SS 2013

Übungen zur Vorlesung
Formale Spezifikation und Verifikation

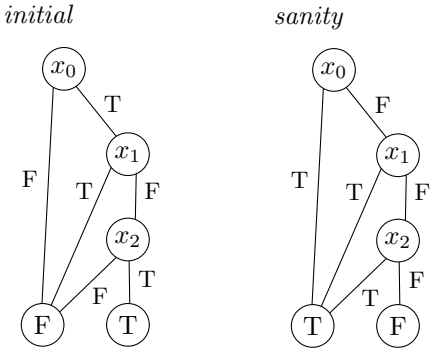
Blatt 4

Aufgabe 4-1 (3 Punkte) In dieser Aufgabe wollen wir die Berechnung der Menge der erreichbaren Zustände eines Systems mittels BDDs anhand eines Beispiels durchführen.

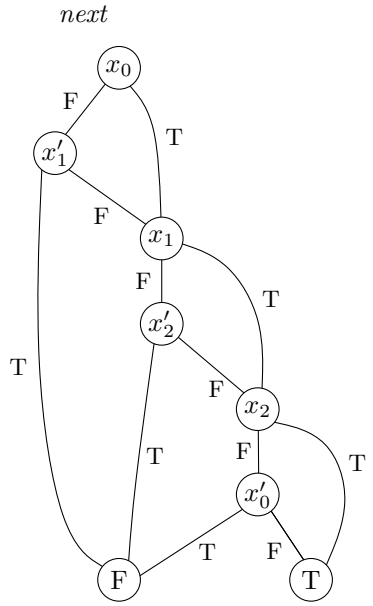
Gegeben sei ein System, in dem jeder Zustand durch drei Bits $b_0, b_1, b_2 \in \{0, 1\}$ repräsentiert wird. Wir schreiben $b_0b_1b_2$ für einen solchen Zustand und definieren das System wie folgt:

- Zulässige Systemzustände sind alle Zustände $b_0b_1b_2$, in denen mindestens ein Bit gesetzt ist, d.h. für die $b_0 + b_1 + b_2 > 0$ gilt.
- Der Anfangszustand ist 101.
- Das System kann von einem Zustand $b_0b_1b_2$ in einen beliebigen Zustand $b'_0b'_1b'_2$ übergehen, für den $b'_1 \leq b_0$ und $b'_2 \leq b_1$ und $b'_0 \leq b_2$ gilt.

Repräsentiert man die Bits b_0, b_1 und b_2 eines Zustands durch drei Boolesche Variablen x_0, x_1 und x_2 (der Zustand 110 wird also zum Beispiel durch $x_0 = x_1 = \mathit{true}$ und $x_2 = \mathit{false}$ repräsentiert), so kann man die Menge der Anfangszustände sowie die Menge der zulässigen Zustände durch folgende BDDs *initial* und *sanity* repräsentieren:



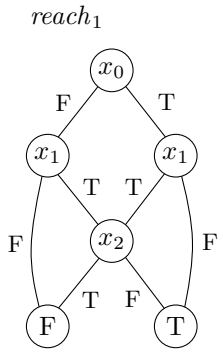
Die möglichen Übergänge von einem Zustand, der durch die Variablen $x_0x_1x_2$ repräsentiert wird, in einen, der durch die Variablen $x'_0x'_1x'_2$ repräsentiert wird, ist durch folgendes BDD *next* gegeben.



In der Vorlesung wurde gezeigt, dass sich die Menge der in höchstens k Schritten von einem Startzustand erreichbaren Zustände durch das BDD $reach_k$ repräsentieren lässt, welches iterativ durch folgende Gleichungen berechnet wird.

$$reach_0 = initial$$
$$reach_{i+1} = reach_i \vee (sanity \wedge ((\exists x_2.\, \exists x_1.\, \exists x_0.\, reach_i \wedge next)[x'_0 := x_0][x'_1 := x_1][x'_2 := x_2]))$$

Ein BDD für $reach_1$ wurde bereits berechnet:

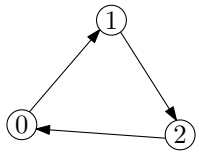


a) Berechnen Sie BDDs für:

   i) $reach_1 \wedge next$
   ii) $\exists x_2.\, \exists x_1.\, \exists x_0.\, reach_1 \wedge next$
   iii) $(\exists x_2.\, \exists x_1.\, \exists x_0.\, reach_1 \wedge next)[x'_0 := x_0][x'_1 := x_1][x'_2 := x_2]$
   iv) $sanity \wedge ((\exists x_2.\, \exists x_1.\, \exists x_0.\, reach_1 \wedge next)[x'_0 := x_0][x'_1 := x_1][x'_2 := x_2])$
   v) $reach_2$

b) Wie muss man die iterativen Gleichungen ändern, um die Menge der Zustände zu errechnen, *von denen aus* ein Startzustand in höchstens k Schritten erreichbar ist?

Aufgabe 4-2 (3 Punkte) Gegeben sei ein System mit Zustandsmenge $\{0, 1, 2\}$, welches folgende Zustandsübergänge erlaubt:



Das System implementiert also einen Zähler mit Überlauf.

Geben Sie BDDs *sanity* und *next* für dieses System an.

Benutzen Sie Boolesche Variablen x_0, x_1 sowie x'_0, x'_1 zur Repräsentierung von Zuständen und Folgezuständen. Die Variablen sollen die Zustände gemäß ihrer Binärkodierung repräsentieren, d.h. Zustand 0 wird repräsentiert durch $x_0 = x_1 = \mathit{false}$, Zustand 1 wird repräsentiert durch $x_0 = \mathit{false} \wedge x_1 = \mathit{true}$ und Zustand 2 wird repräsentiert durch $x_0 = \mathit{true} \wedge x_1 = \mathit{false}$. Folgezustände werden mit x'_0 statt x_0 und x'_1 statt x_1 kodiert.

Verwenden Sie für Ihre BDDs die Variablenordnung $x_0 < x'_0 < x_1 < x'_1$.

Aufgabe 4-3 Gegeben sei die Variablenordnung $x_0 < x_1 < \cdots < x_n$ für ein beliebiges n. Ein BDD B bezüglich dieser Variablenordnung repräsentiert eine Boolesche Funktion $f_B(x_0, x_1, \ldots, x_n)$. In dieser Aufgabe betrachten wir die Menge aller $(n+1)$-Tupel, für die diese Funktion wahr ist:

$$T_B = \{(b_0, b_1, \ldots, b_n) \in \{\mathit{false}, \mathit{true}\}^{n+1} \mid f_B(b_0, b_1, \ldots, b_n) = \mathit{true}\}.$$

Da diese Menge bis zu 2^{n+1} Elemente haben kann, ist es schon für mäßig große n nicht mehr möglich ihre Elemente explizit aufzulisten. Es ist jedoch möglich, die Kardinalität dieser Menge anhand des gegebenen BDDs zu berechnen, ohne die Elemente einzeln aufzulisten.

a) Entwerfen Sie einen Algorithmus, der für ein gegebenes BDD B die Anzahl der Elemente der Menge T_B berechnen? Ihr Algorithmus sollte polynomielle Laufzeit in der Größe des gegebenen BDDs haben.

b) Führen Sie Ihren Algorithmus am Beispiel des BDDs reach_1 von Aufgabe 4-1 aus. (Es gilt $|T_{\mathit{reach}_1}| = 4$.)

Abgabe: Sie können ihre Lösungen bis Donnerstag, den 16.5., um 10 Uhr über UniWorX abgeben.