

FGI-2 – Formale Grundlagen der Informatik II

Modellierung und Analyse von Informatiksystemen

Präsenzlösung 5: CTL und CTL-Model-Checking

Präsenzteil am 11./12.11. – Abgabe am 18./19.11.2013

Präsenzaufgabe 5.1:

1. Betrachten Sie die Kripke-Strukturen M_1 und M_2 im Skript, Seite 41. Gibt es LTL-Formeln, die die beiden Strukturen unterscheiden? Falls ja, geben Sie welche an!

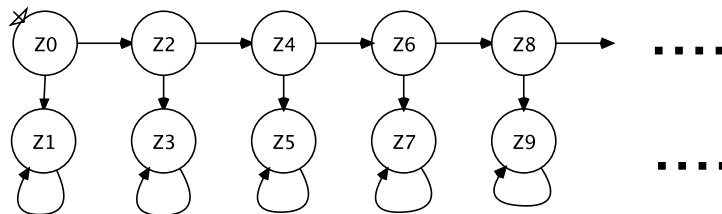
Lösung: Nein, da jede LTL-Formel in allen Pfaden ab Startzustand gelten muss, um erfüllt zu sein. Der Zeitpunkt der Verzweigung ist in den Pfaden nicht sichtbar, beide Strukturen haben dieselbe Menge möglicher Pfade.

2. M_1 und M_2 wie zuvor: Gibt es CTL-Formeln, die die beiden Strukturen unterscheiden?

Lösung: Ja, z.B. ist die Formel $\mathbf{AXEX}q$ im Startzustand der Struktur M_1 erfüllt, da auf allen möglichen Pfaden im Folgezustand rechtsseitig ein Pfad beginnt, in dessen nächsten Zustand q gilt.

In M_2 ist die Formel aber nicht erfüllt, da im linksseitigen Pfad im Folgezustand kein solcher Pfad beginnt.

3. Betrachte die folgende Kripkestruktur mit unendlicher Zustandsmenge S , wobei die Zustandsetikettenfunktion für alle $n \in \mathbb{N}$ durch $E_S(z_{2n}) = \emptyset$ und $E_S(z_{2n+1}) = \{p\}$ definiert sei.



- (a) Gilt $f_1 = \mathbf{EF}p$?
- (b) Gilt $f_2 = \mathbf{AGEF}p$?
- (c) Gilt $f_3 = \mathbf{AF}p$?

Lösung:

- (a) $f_1 = \mathbf{EF}p$ gilt, denn $\pi = z_0 z_1 z_1 \dots$ ist eine Abwicklung, die $\mathbf{F}p$ erfüllt.

- (b) $f_2 = \mathbf{AGEF}p$ gilt.

Wir zeigen zunächst, dass in jedem Zustand $\mathbf{EF}p$ gilt.

- i. Zu jedem Zustand z_{2n} existiert der hier startende Pfad $z_{2n} z_{2n+1} z_{2n+1} \dots$, für den irgendwann (nämlich im 2. Zustand) p gilt.
- ii. Zu jedem Zustand z_{2n+1} , existiert der hier startende Pfad $z_{2n+1} z_{2n+1} \dots$, für den irgendwann (nämlich sofort) p gilt.

Also $M, z \models \mathbf{EF}p$ für alle Zustände z .

Da es für alle Zustände gilt, folgt dass für jeden aus dem Startzustand startenden Pfad π auch $\mathbf{GEF}p$ gilt.

Da es für jeden Pfad gilt, folgt dass auch $f_2 = \mathbf{AGEF}p$ gilt.

(c) $f_3 = \mathbf{AF}p$ gilt nicht, denn p gilt nirgends auf dem Pfad $z_0 z_2 z_4 z_6 \dots$

Präsenzaufgabe 5.2: Äquivalenzen in CTL.

- Formulieren Sie die folgenden Äquivalenzen in natürlicher Sprache und begründen Sie deren Gültigkeit: (i) $\neg \mathbf{G}f \equiv \mathbf{F}(\neg f)$, (ii) $\mathbf{F}f \equiv \mathbf{TrueU}f$, (iii) $\mathbf{A}f \equiv \neg(\mathbf{E}\neg f)$ und (iv) $\neg \mathbf{X}f \equiv \mathbf{X}\neg f$.

Lösung:

- $\neg \mathbf{G}f \equiv \mathbf{F}(\neg f)$: Gilt auf einem Pfad nicht immer f , so gibt es einen Zustand, in dem $\neg f$ gilt (und umgekehrt).
- $\mathbf{F}f \equiv \mathbf{TrueU}f$: Gilt auf einem Pfad in mindestens einem Zustand f , so gilt auch auf allen Zuständen vor diesem Zustand (trivialerweise) \mathbf{True} , also gilt auf diesem Pfad \mathbf{True} bis f gilt. Umgekehrt: Wenn \mathbf{True} immer gilt, bis mindestens einmal f gilt, so muss es einen Zustand geben, in dem f gilt.
- $\mathbf{A}f \equiv \neg(\mathbf{E}\neg f)$: Gilt f in allen vom aktuellen Zustand startenden Pfaden, dann gibt es keinen Pfad, auf dem $\neg f$ gelten würde (und umgekehrt).
- $\neg \mathbf{X}f \equiv \mathbf{X}\neg f$: Wenn es *nicht* stimmt, dass im nächsten Zustand eines Pfades f erfüllt ist, so muss in genau diesem nächsten Zustand zwingend $\neg f$ erfüllt sein. Umgekehrt: Ist im nächsten Zustand $\neg f$ erfüllt, so kann in genau diesem Zustand zwingend f nicht gelten, also gilt *nicht* im nächsten Zustand f . (Das kursive *nicht* steht für das Negationszeichen auf der linken Seite der Äquivalenz).

- Beweisen Sie die Äquivalenzen:

$$\begin{aligned} \mathbf{AX}g &\equiv \neg \mathbf{EX}(\neg g) \\ \mathbf{EF}g &\equiv \mathbf{E}[\mathbf{TrueU}g] \\ \mathbf{AG}g &\equiv \neg \mathbf{EF}(\neg g) \\ \mathbf{AF}g &\equiv \neg \mathbf{EG}(\neg g) \end{aligned}$$

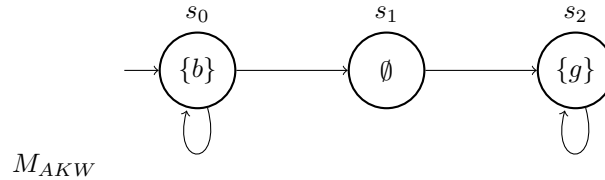
Tipp: Nutzen Sie in der Argumentation die einfacheren Äquivalenzen der ersten Teilaufgabe.

Lösung:

- Mit (iii) und (iv) gilt $\mathbf{AX}g \equiv \neg \mathbf{E}(\neg \mathbf{X}g) \equiv \neg \mathbf{EX}(\neg g)$.
- Mit (ii) gilt $\mathbf{EF}g \equiv \mathbf{E}[\mathbf{TrueU}g]$.
- Mit (iii) und (i) gilt $\mathbf{AG}g \equiv \neg \mathbf{E}(\neg \mathbf{G}g) \equiv \neg \mathbf{EF}(\neg g)$.
- Mit (iii), (i) und $\neg \neg f \equiv f$ gilt $\mathbf{AF}g \equiv \neg \mathbf{E}\neg \mathbf{F}g \equiv \neg \mathbf{E}\neg \mathbf{F}\neg \neg g \equiv \neg \mathbf{E}\neg \neg \mathbf{G}\neg g \equiv \neg \mathbf{EG}(\neg g)$.

Übungsaufgabe 5.3: Betrachten Sie das Kripke-Modell M_{AKW} eines Atomkraftwerkes. In dem Normalbetrieb (Zustand s_0 : „Betrieb“ b) kann eine Störung (Zustand s_1) auftreten, wonach der Störbetrieb (Zustand s_2 : „gestört“ g) aufgenommen wird.

von
6



1. Konstruieren Sie den Abwicklungsbaum $Abwicklung_{AKW}$ (siehe Skript Abb. 3.2) bis zur Tiefe 4 und bezeichnen dazu die Zustände und Etikette wie in M_{AKW} .
2. Wieder sei $Sat(\alpha)$ die Menge aller Zustände, die α erfüllen. Bestimmen Sie mit Hilfe von $Abwicklung_{AKW}$ (oder direkt mit M_{AKW}) die Mengen
 - (a) $Sat(\alpha_1)$ mit $\alpha_1 = \mathbf{EX}b$,
 - (b) $Sat(\mathbf{AG}\alpha_1)$,
 - (c) $Sat(\alpha_2)$ mit $\alpha_2 = \mathbf{AG}\neg b$ und
 - (d) $Sat(\mathbf{EX}\alpha_2)$.
3. Interpretieren Sie für M_{AKW} die Formeln $\beta_1 = \mathbf{AGEX}b$ und $\beta_2 = \mathbf{EXAG}\neg b$ und entscheiden Sie (unter zu Hilfenahme von 2.), ob sie für M_{AKW} gelten, d.h. ob
 - (a) $M_{AKW} \models \mathbf{AGEX}b$ und
 - (b) $M_{AKW} \models \mathbf{EXAG}\neg b$
 gelten. Dabei sei (analog zu Def. 3.5) $M \models \alpha \Leftrightarrow \forall s \in S_0 : M, s \models \alpha$.
4. (a) Beweisen Sie für alle Aussagen a : $\mathbf{AXAG}a \equiv \mathbf{AGAX}a$.
 Hinweis: Konstruieren Sie (symbolisch) für beide Seiten den Abwicklungsbaum.
 (Anmerkung: Die Äquivalenz \equiv ist definiert als: $f \equiv g$ gilt genau dann, wenn für jedes Modell M gilt: $M \models f$ gdw. $M \models g$. f ist also in jedem Modell wahr, in dem auch g wahr ist und andersherum. (Siehe auch die Definition zu Beginn von Abschnitt 3.4 im Skript auf Seite 40.))
- (b) Beweisen Sie, dass folgende Äquivalenz **nicht** gilt: $\mathbf{EXEG}(\neg b \wedge \neg g) \equiv \mathbf{EGEX}(\neg b \wedge \neg g)$.
 Hinweis: Benutzen Sie M_{AKW} zur Konstruktion eines Gegenbeispiels.
5. Indem man alle Symbole **A** und **E** streicht, erhält man aus einer CTL-Formel eine LTL-Formel. Ist die so erhaltene Formel äquivalent zur ursprünglichen (im Sinne der Definition vor Satz 3.14 auf Seite 40)?
 - (a) Beweisen Sie als positives Beispiel: $\mathbf{AGAX}b \equiv \mathbf{GX}b$.
 (Hinweis: Die Äquivalenz ist hier so zu verstehen, dass jedes Modell, dass die CTL-Formel $\mathbf{AGAG}x$ auch die LTL-Formel $\mathbf{GX}b$ erfüllt und umgekehrt.)
 - (b) Beweisen Sie als negatives Beispiel: $\mathbf{EG}b \equiv \mathbf{Gb}$.
 Hinweis: Benutzen Sie M_{AKW} zur Konstruktion eines Gegenbeispiels.

Übungsaufgabe 5.4: Wenden Sie den CTL-Model-Checking-Algorithmus (Skript Abschnitt 4.2) auf die beiden CTL-Formeln $\beta_1 = \mathbf{AGEX}b$ und $\beta_2 = \mathbf{EXAG}\neg b$ von Aufgabe 5.3.3 und die Kripke-Struktur M_{AKW} an. Gehen Sie dabei folgendermaßen vor:

von
6

1. Bringen Sie β_1 und β_2 in eine Form β'_1 und β'_2 , die nur **EX**, **EG** und **EU** verwendet. Auch **EF** kann sinnvollerweise als spezielle Form von **EU** benutzt werden.
2. Wenden Sie den CTL-Algorithmus nicht auf den Graphen von M_{AKW} an, sondern in Form folgender Tabelle:

Teilformel	Zustand s_0	Zustand s_1	Zustand s_2
b	+	−	−
$\mathbf{EX}b$
...

In der linken Spalte steht die zu prüfende Formel und aufsteigend alle Teilformeln, beginnend mit der kleinsten Teilformel b . Unter „Zustand s_i “ steht ein +, wenn die Teilformel der Zeile im entsprechenden Schritt im Graphen an diesen Zustand zu schreiben ist. Im anderen Fall steht ein −.

3. Entscheiden Sie, ob $M_{AKW} \models \beta_1$ und $M_{AKW} \models \beta_2$ gilt. Vergleichen Sie die letzten Zeilen der Tabellen mit Ihrem Ergebnis zu $Sat(\beta_1)$ und $Sat(\beta_2)$ aus Aufgabe 5.3.2. und erklären Sie Übereinstimmungen.
4. Anmerkung: Zu dem Verfahren der Umwandlung einer CTL-Formel in eine LTL-Formel gibt es ein interessantes Theorem. Wenn die durch Streichen der Quantoren **A** und **E** erzeugte LTL-Formel nicht äquivalent zur ursprünglichen ist, dann gibt es überhaupt keine LTL-Formel, die das leistet!

Bisher erreichbare Punktzahl: 60