Automatisches Beweisen—Vertiefung

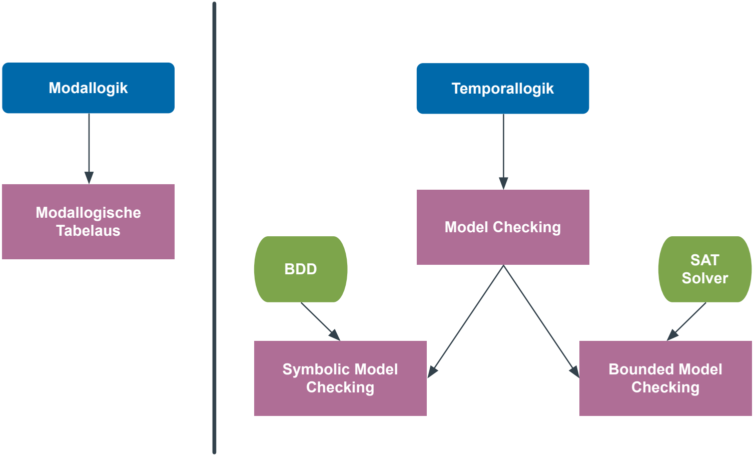
Model Checking

Christoph Zengler

Arbeitsbereich Symbolisches Rechnen
Prof. Dr. Wolfgang Küchlin
Universität Tübingen

10. Januar 2012

Der Plan für die nächsten drei Wochen



Ein Beispiel für Model Checking

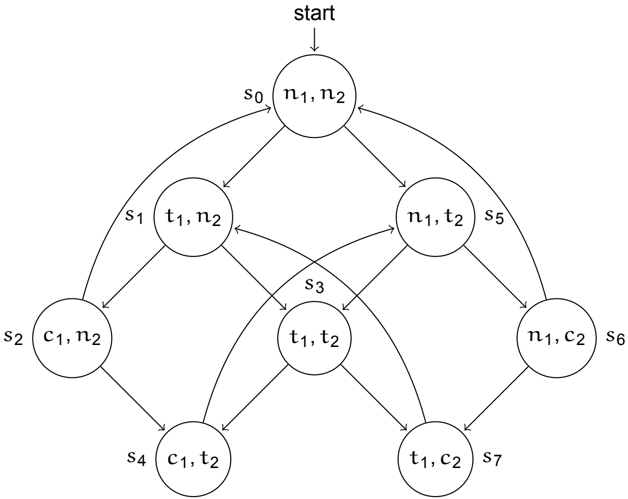
Gegenseitiger Ausschluss von Prozessen

- Gleichzeitiger Zugriff auf Ressourcen muss gekapselt werden
- Prozess darf nur in seinem kritischen Bereich (KB) darauf zugreifen

Eigenschaften

1. Sicherheit/Safety: Nur ein Prozess kann pro Zeitschritt in seinem kritischen Bereich sein.
2. Lebendigkeit/Liveness: Wann immer ein Prozess beantragt, in seinen KB zu kommen, wird ihm dies irgendwann erlaubt.
3. Nicht-Blockierend/non-blocking: Ein Prozess kann jederzeit beantragen in seinen KB zu kommen.

1. Modellierungsversuch



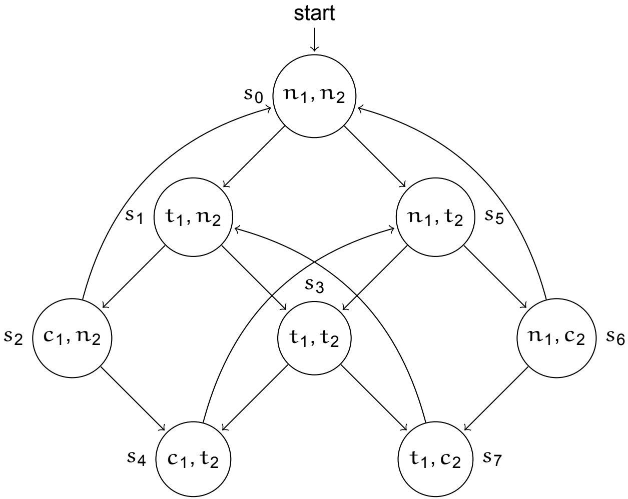
Zwei Prozesse, die entweder

(n) in ihrem nichtkritischen Bereich sind,

(t) beantragen in ihren KB zu kommen, oder

(c) in ihrem KB sind

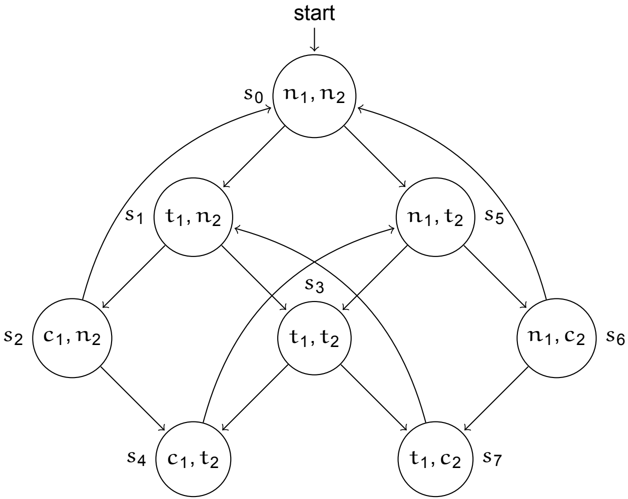
1. Modellierungsversuch



Safety in LTL

$$\mathbf{G}\,\neg(c_1 \wedge c_2) \qquad \checkmark$$

1. Modellierungsversuch

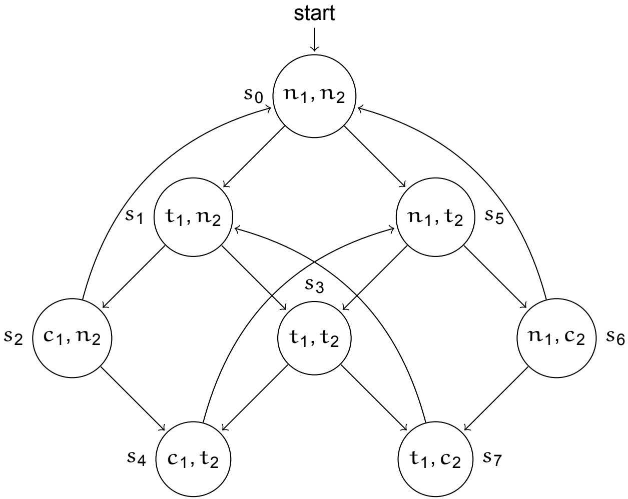


Liveness in LTL

$$\mathbf{G}(t_1 \to \mathbf{F}\, c_1) \land \mathbf{G}(t_2 \to \mathbf{F}\, c_2) \qquad ↯$$

Problem: Nicht-Determinismus in s_3

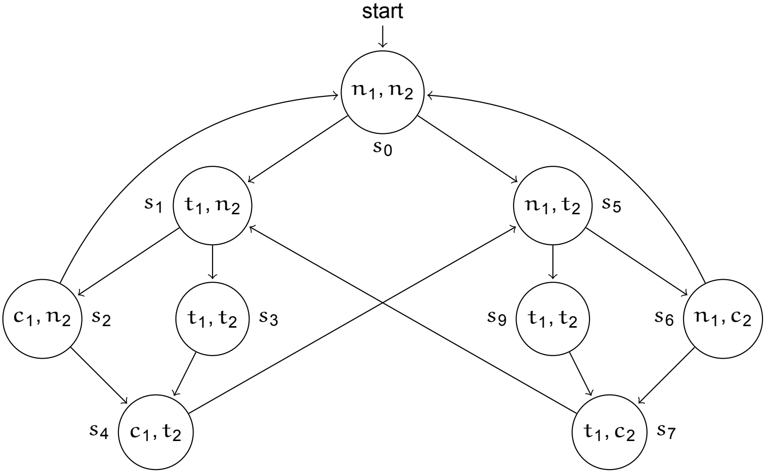
1. Modellierungsversuch



Non-blocking in LTL

Kann nicht formuliert werden *(Für jeden Zustand mit n_1* ***gibt es einen Nachfolgezustand,*** *der t_1 erfüllt.)*

2. Modellierungsversuch



Model Checking

Definition (Model Checking)

Gegeben sei eine Kripke-Struktur $\mathcal{M}$, ein Zustand $s \in S$ und eine temporallogische Formel φ. Die Frage, ob $\mathtt{eval}(\mathcal{M}, s, \varphi)$ wahr ist, wird als Model Checking bezeichnet.

- Wir können keine unendlichen Bäume aus der Kripke Struktur abwickeln, d.h. Check muss auf der Struktur selber stattfinden
- Im Falle, dass φ nicht gilt, kann Model Checking einen Pfad in $\mathcal{M}$ angeben, der φ verletzt (Gegenbeispiel)

Alternative Definition (Model Checking)

Gegeben sei eine Kripke Struktur $\mathcal{M}$ und eine temporallogische Formel φ. Model Checking beantwortet die Frage, in welchen Zuständen $s \in S$ die Formel φ gilt.

- Die zweite Fragestellung schließt die erste ein
- Die alternative Definition ist algorithmisch leichter lösbar

Der Labelling Algorithmus für CTL

Idee!

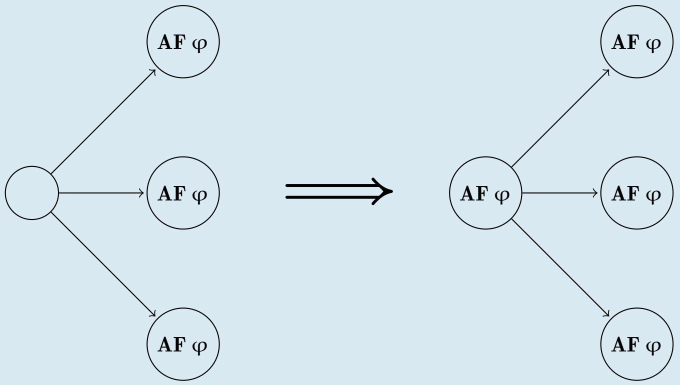
Beschrifte alle Zustände mit Teilformeln der relevanten Formeln (von den kleinsten Formeln nach außen), die an ihnen gelten. Schaue am Ende, in welchen Zuständen die relevante Formel gilt.

Wir brauchen nicht alle Operatoren zu betrachten

- Bei den aussagenlogischen Operatoren reichen $\bot$, $\neg$ und $\wedge$
- Bei den temporallogischen Operatoren reichen $\mathbf{AF}$, $\mathbf{EU}$ und $\mathbf{EX}$
  - $\mathbf{AG}\,\varphi \equiv \neg\,\mathbf{EF}\,\neg\varphi$
  - $\mathbf{A}[\varphi\,\mathbf{U}\,\psi] \equiv \neg(\mathbf{E}[\neg\psi\,\mathbf{U}(\neg\varphi \wedge \neg\psi)] \vee \mathbf{EG}\,\neg\psi)$
  - $\mathbf{AX}\,\varphi \equiv \neg\,\mathbf{EX}\,\neg\varphi$
  - $\mathbf{EF}\,\varphi \equiv \mathbf{E}[\top\,\mathbf{U}\,\varphi]$
  - $\mathbf{EG}\,\varphi \equiv \neg\,\mathbf{AF}\,\neg\varphi$

Grundidee des Labelling Algorithmus

Beispiel (Beschriftung von Zuständen mit $\mathbf{AF}$)



Implementierung

- keine explizite Annotation der Zustände

Der Algorithmus

Hauptalgorithmus

Algorithmus: $\texttt{modelCheckCTL}(\mathcal{M}, \psi)$

Eingabe: Kripke-Struktur $\mathcal{M}$ und CTL Formel ψ
Ausgabe: Menge an Zuständen $\subseteq S$, an denen ψ gilt

1. Reduziere ψ auf die Operatoren $\bot$, $\neg$, $\wedge$ $\mathbf{AF}$, $\mathbf{EU}$ und $\mathbf{EX}$
2. $\texttt{modelCheckCTL}(\mathcal{M}, \psi) = \psi\ \textit{match}$

$$
\begin{array}{lll}
 & \bot & \rightsquigarrow \emptyset \\
| & v \in \mathcal{V} & \rightsquigarrow \{s \in S \mid v \in L(s)\} \\
| & \neg\varphi & \rightsquigarrow S - \texttt{modelCheckCTL}(\mathcal{M}, \varphi) \\
| & \varphi_1 \wedge \varphi_2 & \rightsquigarrow \texttt{modelCheckCTL}(\mathcal{M}, \varphi_1) \cap \texttt{modelCheckCTL}(\mathcal{M}, \varphi_2) \\
| & \mathbf{EX}\,\varphi & \rightsquigarrow \texttt{mc}_{\mathbf{EX}}(\mathcal{M}, \varphi) \\
| & \mathbf{AF}\,\varphi & \rightsquigarrow \texttt{mc}_{\mathbf{AF}}(\mathcal{M}, \varphi) \\
| & \mathbf{E}[\varphi_1\,\mathbf{U}\,\varphi_2] & \rightsquigarrow \texttt{mc}_{\mathbf{EU}}(\mathcal{M}, \varphi_1, \varphi_2)
\end{array}
$$

Existentielle und Universelle Urbilder

Zwei Funktionen zum Berechnen von Urbildern:

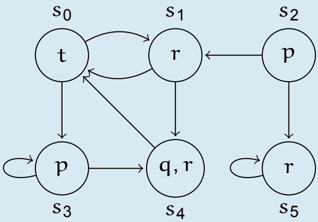
- $\mathtt{pre}_{\exists}(Y)$: Berechnet für eine Menge von Zuständen Y alle Zustände, die einen Übergang in Y machen *können*

$$\mathtt{pre}_{\exists}(Y) = \{s \in S \mid \text{exists } s' \text{ with } s \longrightarrow s' \text{ and } s' \in Y\}$$

- $\mathtt{pre}_{\forall}(Y)$: Berechnet für eine Menge von Zuständen Y alle Zustände, die *nur* Übergange in Y machen

$$\mathtt{pre}_{\forall}(Y) = \{s \in S \mid \text{for all } s' : s \longrightarrow s' \text{ implies } s' \in Y\}$$

Beispiel ($\mathtt{pre}_{\exists}$ und $\mathtt{pre}_{\forall}$)



- $\mathtt{pre}_{\exists}(\{s_3, s_4\}) = \{s_0, s_1, s_3\}$
- $\mathtt{pre}_{\forall}(\{s_3, s_4\}) = \{s_3\}$
- $\mathtt{pre}_{\exists}(\{s_5\}) = \{s_2, s_5\}$
- $\mathtt{pre}_{\forall}(\{s_5\}) = \{s_5\}$

Der Unteralgorithmus $\mathtt{mc}_{\mathbf{EX}}$

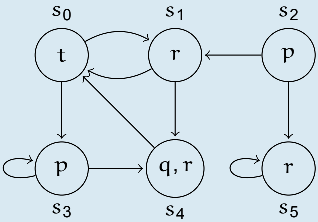
Algorithmus: $\mathtt{mc}_{\mathbf{EX}}(\mathcal{M}, \varphi)$

Eingabe: Kripke-Struktur $\mathcal{M}$ und CTL Formel φ
Ausgabe: Menge an Zuständen $\subseteq S$, an denen $\mathbf{EX}\,\varphi$ gilt

---

$X = \mathtt{modelCheckCTL}(\mathcal{M}, \varphi)$
$Y = \mathtt{pre}_{\exists}(X)$
return Y

Beispiel ($\mathtt{mc}_{\mathbf{EX}}$)



$\mathtt{mc}_{\mathbf{EX}}(\mathcal{M}, r)$:

1. $X = \mathtt{modelCheckCTL}(\mathcal{M}, r) = \{s_1, s_4, s_5\}$
2. $Y = \mathtt{pre}_{\exists}(X) = \{s_0, s_1, s_2, s_3, s_5\}$
3. **return** Y

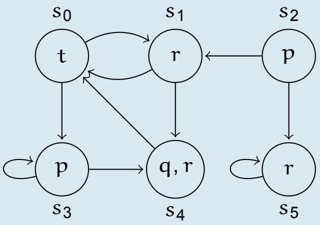
Der Unteralgorithmus $\mathtt{mc}_{\mathbf{AF}}$

Algorithmus: $\mathtt{mc}_{\mathbf{AF}}(\mathcal{M}, \varphi)$

Eingabe: Kripke-Struktur $\mathcal{M}$ und CTL Formel φ
Ausgabe: Menge an Zuständen $\subseteq S$, an denen $\mathbf{AF}\,\varphi$ gilt

$X = S$
$Y = \mathtt{modelCheckCTL}(\mathcal{M}, \varphi)$
while $X \neq Y$ **do**
  $X = Y$
  $Y = Y \cup \mathtt{pre}_\forall(Y)$
return Y

Beispiel ($\mathtt{mc}_{\mathbf{AF}}$)



$\mathtt{mc}_{\mathbf{AF}}(\mathcal{M}, p)$:

1. $X = S = \{s_0, s_1, s_2, s_3, s_4, s_5\}$
2. $Y = \{s_2, s_3\}$
3. $X = \{s_2, s_3\}$, $Y = \{s_2, s_3\}$
4. **return** Y

Der Unteralgorithmus $\texttt{mc}_{\mathbf{EU}}$

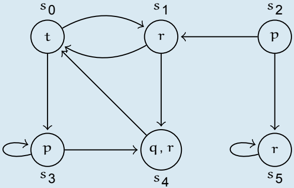
Algorithmus: $\texttt{mc}_{\mathbf{EU}}(\mathcal{M}, \varphi_1, \varphi_2)$

Eingabe: Kripke-Struktur $\mathcal{M}$ und CTL Formeln φ_1 und φ_2
Ausgabe: Menge an Zuständen $\subseteq S$, an denen $\mathbf{E}[\varphi_1 \,\mathbf{U}\, \varphi_2]$ gilt

---

$W = \texttt{modelCheckCTL}(\mathcal{M}, \varphi_1)$
$X = S$
$Y = \texttt{modelCheckCTL}(\mathcal{M}, \varphi_2)$
while $X \neq Y$ **do**
  $X = Y$
  $Y = Y \cup (W \cap \texttt{pre}_{\exists}(Y))$
return Y

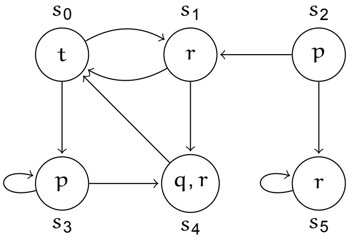
Beispiel ($\texttt{mc}_{\mathbf{EU}}$)



$\texttt{mc}_{\mathbf{EU}}(\mathcal{M}, t, r)$:

1. $W = \{s_0\}$, $X = S$, $Y = \{s_1, s_4, s_5\}$
2. $X = \{s_1, s_4, s_5\}$, $Y = \{s_0, s_1, s_4, s_5\}$
3. $X = \{s_0, s_1, s_4, s_5\}$, $Y = \{s_0, s_1, s_4, s_5\}$
4. **return** Y

Beispiel



Beispiel ($\mathbf{AG}(\mathbf{AF}(p \vee t))$)

1. Reduktion: $\varphi = \neg\,\mathbf{E}[\neg\bot\,\mathbf{U}\,\neg(\mathbf{AF}(p \vee t))]$
2. $\texttt{modelCheckCTL}(\mathcal{M}, \varphi)$

$$= S - \texttt{modelCheckCTL}(\mathcal{M}, \mathbf{E}[\neg\bot\,\mathbf{U}\,\neg(\mathbf{AF}(p \vee t))])$$
$$= S - \texttt{mc}_{\mathbf{EU}}(\mathcal{M}, \neg\bot, \neg(\mathbf{AF}(p \vee t)))$$

3. $\texttt{mc}_{\mathbf{EU}}(\mathcal{M}, \neg\bot, \neg(\mathbf{AF}(p \vee t)))$
   - $W = \texttt{modelCheckCTL}(\mathcal{M}, \neg\bot) = \{s_0, s_1, s_2, s_3, s_4, s_5\}$
   - $X = \{s_0, s_1, s_2, s_3, s_4, s_5\}$
   - $Y = \texttt{modelCheckCTL}(\mathcal{M}, \neg(\mathbf{AF}(p \vee t))) = S - \texttt{mc}_{\mathbf{AF}}(\mathcal{M}, p \vee t)$
   - ... *(an der Tafel)*

Komplexität

Komplexität

$$O(f \cdot V \cdot (V + E))$$

mit

- f Anzahl der Konnektive in der Formel
- V Anzahl der Zustände
- E Anzahl der Zustandsübergänge

d.h. linear in der Formel, quadratisch in der Kripke-Struktur

Verbesserung

Durch spezielle Behandlung von $\mathbf{EG}$ kann der Algorithmus auch linear in der Struktur sein, d.h. $O(f \cdot (V + E))$

Zustandsexplosion

- Modelle sind oft exponentiell in der Anzahl der Variablen (d.h. eine neue Variable verdoppelt das Modell)
- Umgehen durch z.B. Symbolic Model Checking (nächste VL)

LTL Model Checking

- CTL Model Checking: Labelling Algorithmus annotiert Zustände mit Formeln, die an ihnen gelten
- LTL Model Checking: Evaluation nicht an Zuständen sondern auf Pfaden

Prinzipielles Vorgehen

Wir wollen testen, ob $\mathcal{M}, s \models \varphi$, d.h. ob alle von s ausgehenden Pfade φ erfüllen.

1. Konstruktion eines Automaten $A_{\neg\varphi}$ (Tableau) für die Formel $\neg\varphi$.
2. Kombination des Automaten mit der Kripke Struktur $\mathcal{M}$.
3. Suche einen von s ausgehenden Pfad in dem kombinierten Übergangssystem. Wird ein solcher Pfad gefunden, ist er ein Gegenbeispiel für die Verifikationsbedingung.

Konstruktion des Tableau

Konstruktion des Automaten $A_{\neg\varphi}$ für $\neg\varphi$

- Automat kann einen Trace akzeptieren
- Trace: Folge von Belegungen der aussagenlogischen Variablen
- Von einem Pfad kann sein Trace abstrahiert werden
- Für alle Pfade π gilt: $\pi \models \psi$ gdw. der Trace von π von A_ψ akzeptiert wird.

$\Rightarrow$ A_ψ kodiert genau die Traces, die ψ erfüllen.

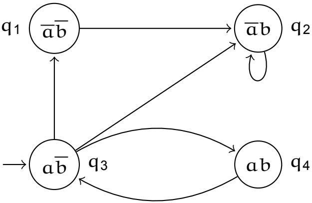
$\Rightarrow$ $A_{\neg\varphi}$ kodiert genau die Traces, die φ nicht erfüllen.

Definition (Akzeptanz eines Trace)

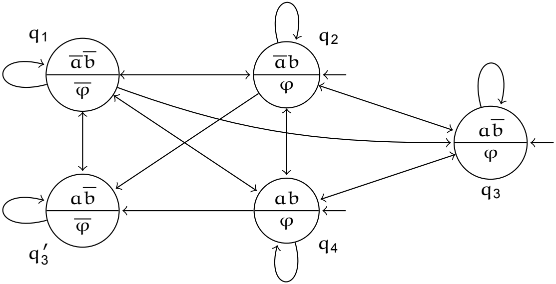
Ein Automat akzeptiert einen Trace t, wenn es einen Pfad π durch den Automaten gibt mit

- π startet in einem Startzustand
- π gehorcht der Übergangsrelation des Automaten
- t ist der Trace von π
- π gehorcht einer bestimmten *Akzeptanzbedingung*

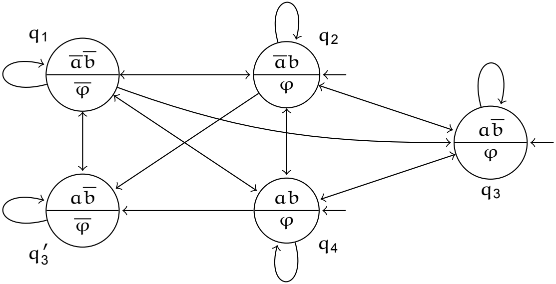
Konstruktion des Tableau—Beispiel



Wir betrachten die Formel $\neg(a\,\mathbf{U}\,b)$, d.h. konstruieren den Automaten für $\varphi = a\,\mathbf{U}\,b$



Konstruktion des Tableau—Beispiel

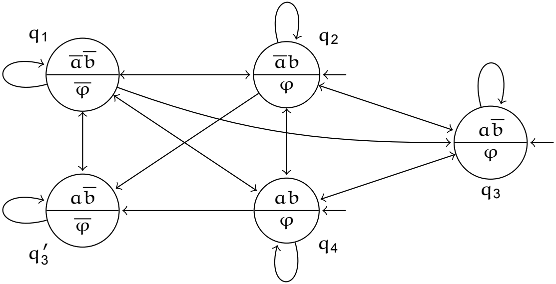


Beispiel

- Akzeptanzbedinung: q_3 darf nicht unendlich oft auf dem Pfad vorkommen
- Betrachte den Trace: $a\overline{b}, a\overline{b}, a\overline{b}, ab, ab, \overline{a}\overline{b}, a\overline{b}, a\overline{b}, \ldots$
- Wähle den Pfad: $q_3, q_3, q_3, q_4, q_4, q_1, q_3', q_3', \ldots$

$\Rightarrow$ Trace wird vom Automat akzeptiert

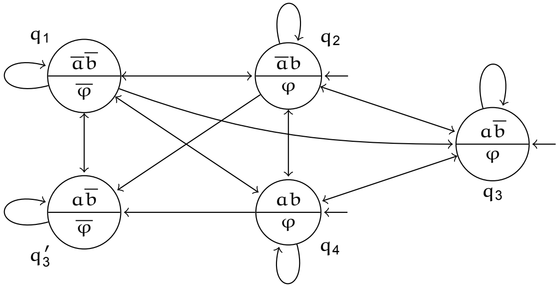
Konstruktion des Tableau—Beispiel



Intuition hinter dem Automat

- Ist ein Zustand mit φ annotiert: entweder wir erwarten, dass φ noch wahr wird, oder es ist gerade wahr geworden
- Ist ein Zustand mit $\overline{\varphi}$ annotiert: entweder wir erwarten nicht mehr, dass φ noch wahr wird und es ist nicht gerade wahr geworden
- Es gibt mind. einen Zustand für jede Belegung $\{ab, \overline{a}b, a\overline{b}, \overline{a}\overline{b}\}$

Konstruktion des Tableau—Beispiel



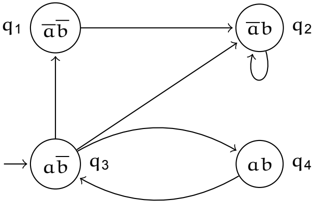
Intuition hinter dem Automat

- $\{ab, \overline{a}b, \overline{a}\,\overline{b}\}$ bestimmen bereits über den Wahrheitswert von $a \,\mathbf{U}\, b$
- In $a\overline{b}$ kann $a \,\mathbf{U}\, b$ sowohl wahr als auch falsch werden

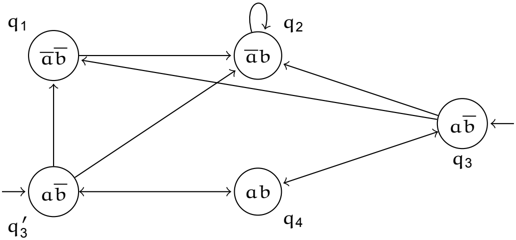
$\Rightarrow$ Teile Zustand q_3 auf in q_3 und q_3'

- Jeder Pfad aus q_3 führt durch einen Zustand, an dem b gilt
- Alle anderen Übergänge sind möglich
- Gilt die Akzeptanzbedingung, erfüllen alle Pfade die Formel $a \,\mathbf{U}\, b$

Kombination des Automaten mit dem Modell

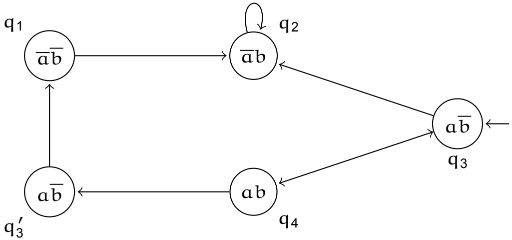


Teile den Zustand q_3 wie im Automaten auf



Kombination des Automaten mit dem Modell

Schneide den Automaten $A_{a\,\mathbf{U}\,b}$ mit der Kripke-Struktur mit geteiltem q_3:



Suche einen Pfad im kombinierten Automaten

- Muss bei q_3 beginnen *(einziger Startzustand)*
- Darf nicht unendlich oft in q_3 kommen *(Akzeptanzbedingung)*
- 2 mögliche Pfade
  - $q_3, (q_4, q_3)^* q_2, q_2, \ldots$
  - $q_3, q_4, (q_3, q_4)^* q_3', q_1, q_2, q_2, \ldots$

$\Rightarrow$ Es gibt ein Gegenbeispiel zu der Bedingung $\neg(a\,\mathbf{U}\,b)$

Konstruktion des Automaten—Algorithmus

Schritt 1: Reduziere die Formel auf bestimmte Operatoren

Reduziere die Formel auf die Operatoren $\perp, \neg, \vee, \mathbf{X}$ und $\mathbf{U}$

- $\varphi \, \mathbf{R} \, \psi \equiv \neg(\neg\varphi \, \mathbf{U} \, \neg\psi)$
- $\varphi \, \mathbf{W} \, \psi \equiv \psi \, \mathbf{R}(\varphi \vee \psi)$
- $\mathbf{F} \, \varphi \equiv \top \, \mathbf{U} \, \varphi$
- $\mathbf{G} \, \varphi \equiv \neg \, \mathbf{F} \, \neg\varphi$

Schritt 2: Konstruiere die Hülle der Formel

Die Hülle $\mathcal{C}(\varphi)$ einer Formel ist die Menge aller Teilformeln von φ inklusive ihrer Negationen.

Beispiel ($\mathcal{C}(a \, \mathbf{U} \, b)$)

$$\{a, b, \neg a, \neg b, a \, \mathbf{U} \, b, \neg(a \, \mathbf{U} \, b)\}$$

Konstruktion des Automaten—Algorithmus

Schritt 3: Konstruiere die Zustände des Automaten

Die Zustände $q, q', \ldots$ von A_φ sind maximale Teilmengen von $\mathcal{C}(\varphi)$, die die folgenden Bedingungen erfüllen:

- Für alle nicht-negierten $\psi \in \mathcal{C}(\varphi)$ ist entweder $\psi \in q$ oder $\neg\psi \in q$
- $\psi_1 \vee \psi_2 \in q$ gilt gdw. $\psi_1 \in q$ oder $\psi_2 \in q$ wann immer $\psi_1 \vee \psi_2 \in \mathcal{C}(\varphi)$
- Wenn $\psi_1 \mathbf{U} \psi_2 \in q$, dann $\psi_1 \in q$ oder $\psi_2 \in q$
- Wenn $\neg(\psi_1 \mathbf{U} \psi_2) \in q$, dann $\neg\psi_2 \in q$

Die Anfangszustände sind genau die Zustände q mit $\varphi \in q$

Beispiel (Zustände für $a \mathbf{U} b$)

$$\mathcal{C}(a \mathbf{U} b) = \{a, b, \neg a, \neg b, a \mathbf{U} b, \neg(a \mathbf{U} b)\}$$

Maximale Teilmengen:

- $\{a, b, a \mathbf{U} b\}$ *Anfangszustand*
- $\{\neg a, b, a \mathbf{U} b\}$ *Anfangszustand*
- $\{a, \neg b, a \mathbf{U} b\}$ *Anfangszustand*
- $\{a, \neg b, \neg(a \mathbf{U} b)\}$
- $\{\neg a, \neg b, \neg(a \mathbf{U} b)\}$

Konstruktion des Automaten—Algorithmus

Schritt 4: Konstruiere die Zustandsübergänge

Es gilt $q \longrightarrow q'$ gdw. alle der folgenden Bedingungen gelten:

- Wenn $\mathbf{X}\,\varphi \in q$, dann $\varphi \in q'$
- Wenn $\neg\mathbf{X}\,\varphi \in q$, dann $\neg\varphi \in q'$
- Wenn $\varphi_1 \,\mathbf{U}\, \varphi_2 \in q$ und $\varphi_2 \notin q$, dann $\varphi_1 \,\mathbf{U}\, \varphi_2 \in q'$
- Wenn $\neg(\varphi_1 \,\mathbf{U}\, \varphi_2) \in q$ und $\varphi_1 \in q$, dann $\neg(\varphi_1 \,\mathbf{U}\, \varphi_2) \in q'$

Beispiel (Zustandsübergänge für $a \,\mathbf{U}\, b$)

Zustände:

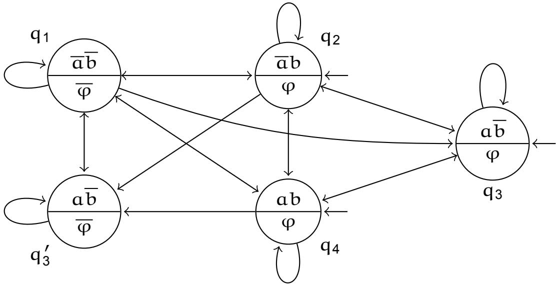
- q_1: $\{\neg a, \neg b, \neg(a \,\mathbf{U}\, b)\}$
- q_2: $\{\neg a, b, a \,\mathbf{U}\, b\}$
- q_3: $\{a, \neg b, a \,\mathbf{U}\, b\}$
- q_3': $\{a, \neg b, \neg(a \,\mathbf{U}\, b)\}$
- q_4: $\{a, b, a \,\mathbf{U}\, b\}$

Übergänge:

- q_1, q_2 und q_4 haben Übergänge zu allen Zuständen (inkl. ihnen selber)
- $q_3 \longrightarrow q_2$, $q_3 \longrightarrow q_3$, $q_3 \longrightarrow q_4$
- $q_3' \longrightarrow q_1$, $q_3' \longrightarrow q_3'$

Konstruktion des Automaten—Algorithmus

Schritt 1–4



Schritt 5: Erzeugen der Akzeptanzbedingung

- Seien $\psi_1 \mathbf{U} \chi_1, \ldots, \psi_k \mathbf{U} \chi_k$ alle Subformeln dieser Form in $\mathcal{C}(\varphi)$
- Dann ist die Akzeptanzbedingung:

Ein Pfad durch den Automaten wird nur akzeptiert, wenn der Pfad für alle $1 \leq i \leq k$ *unendliche viele Zustände passiert, die* $\neg(\psi_i \mathbf{U} \chi_i) \vee \chi_i$ *erfüllen.*

LTL Model Checking—Algorithmus

Algorithmus: `modelCheckLTL`$(\mathcal{M}, s, \varphi)$

Eingabe: Kripke-Struktur $\mathcal{M}$, Zustand $s \in S$ und LTL Formel ψ
Ausgabe: `true`, wenn $\mathcal{M}, s \models \varphi$, `false` sonst

---

1. Konstruiere den Automaten $A_{\neg\varphi}$
   1. Reduktion der Formel
   2. Konstruktion der Hülle
   3. Konstruktion der Zustände
   4. Konstruktion der Zustandsübergänge
   5. Erzeugen der Akzeptanzbedingungen
2. Bilde Schnitt von $\mathcal{M}$ und $A_{\neg\varphi}$
   1. Teile zuerst Zustände von $\mathcal{M}$ entsprechend $A_{\neg\varphi}$ auf
   2. Übernehme nur Zustandsübergänge, die in beiden Strukturen vorkommen
3. Gibt es einen Pfad in dem kombinierten Automaten, der bei s startet und den Akzeptanzbedingungen gehorcht, so gebe `false` zurück, ansonsten `true`.

Literaturtipps

Literaturhinweis

- *M. Huth & M. Ryan.* **Logic in Computer Science** **Chapter 3.** Cambridge University Press, 2004.
- *E. M. Clarke, O. Grumberg & D. A. Peled.* **Model Checking.** The MIT Press, 1999.

Web Links

- `http://nusmv.fbk.eu/` — Model Checker NuSMV