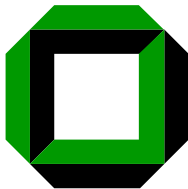


Formale Systeme

Prof. P.H. Schmitt

Fakultät für Informatik
Universität Karlsruhe (TH)

Winter 2007/2008



Aussagenlogische Modellprüfung

Bounded Model Checking



Notationsvariante

Sei $\mathcal{A}$ ein Büchi-Automat, A eine LTL-Formel.

$$\mathcal{A} \models A$$

bedeutet:

für jede akzeptierende Berechnungsfolge s von $\mathcal{A}$ gilt $s \models A$

Dabei bedeutet $s \models A$, daß für die zugeordnete omega-Struktur ξ_s gilt $\xi_s \models A$.

Bisher hatten wir nur $\xi \models A$ für eine beliebige omega-Struktur ξ definiert. In dieser Vorlesung kommen nur omega-Strukturen vor, die von akzeptierenden Berechnungsfolgen eines Büchi-Automaten stammen.



Semantik von LTL für Berechnungsfolgen

Sei s eine akzeptierende Berechnungsfolge eines Büchi-Automaten $\mathcal{A}$.
 $\xi(n)$ bezeichnet, wie gehabt, die Markierung der Kante von s_n nach s_{n+1} .
und s_n ist die Restfolge $(s_m)_{m \geq n}$

$s \models p$	gdw	$p \in \xi(0)$ (p ein AL Atom)
$s \models op(A, B)$		für aussagenlogische Kombinationen $op(A, B)$ von A und B wie üblich
$s \models \Box A$	gdw	für alle $n \in \mathbb{N}$ gilt $s_n \models A$
$s \models \Diamond A$	gdw	es gibt ein $n \in \mathbb{N}$ mit $s_n \models A$
$s \models A \mathbf{U} B$	gdw	es gibt $n \in \mathbb{N}$ mit $s_n \models B$ und für alle m mit $0 \leq m < n$ gilt $s_m \models A$
$s \models X A$	gdw	$s_1 \models A$



Ziel

Zu gegebenem Büchi-Automaten $\mathcal{A}$ und gegebener LTL-Formel A ist eine endliche Menge aussagenlogischer Formeln M gesucht, so daß gilt:

Es gibt eine akzeptierende Berechnungsfolge s von $\mathcal{A}$ mit $s \models A$

genau dann, wenn

M ist erfüllbar

Problem s ist unendlich.

So einfach geht das nicht!



Zyklische Berechnungsfolgen

Definition

- ① Eine endliche Berechnungsfolge $s_0, \dots, s_k$ heißt *i-zyklisch*, falls $0 \leq i < k$ und $s_k = s_i$.
- ② Eine endliche Berechnungsfolge $s_0, \dots, s_k$ heißt *zyklisch*, falls sie *i-zyklisch* ist für ein i mit $0 \leq i < k$.
- ③ Eine *i-zyklische* Berechnungsfolge $s_0, \dots, s_k$ heißt *akzeptierend*, wenn unter den Zuständen $s_i, \dots, s_k$ mindestens ein Endzustand vorkommt.
- ④ Für eine *i-zyklische* Berechnungsfolge $s = s_0, \dots, s_k$ und eine LTL Formel A schreiben wir $s \models A$ anstelle von $s_0, \dots, s_{i-1}, (s_i, \dots, s_{k-1})^\omega \models A$.



Lemma

Sei $\mathcal{A}$ ein Büchi-Automat und A eine LTL-Formel.

Falls es eine akzeptierende Berechnungsfolge t gibt mit

$$t \models A,$$

dann gibt es auch eine endliche zyklische akzeptierende Berechnungsfolge s_e mit

$$s_e \models A.$$



Präzisiertes Ziel

Zu einem gegebenen Büchi-Automaten $\mathcal{A}$

und eine LTL-Formel F

gibt es für jedes $k \in \mathbb{N}$ eine aussagenlogische Formel M_k , so daß gilt

M_k ist erfüllbar gdw es gibt eine zyklische Berechnungsfolge s
der Länge k mit $s \models F$

Falls erforderlich schreiben wir anstelle von M_k genauer $M_k(\mathcal{A}, F)$.



Die AL-Variablen in M_k

Die Zustände von $\mathcal{A}$ seien $1, 2, \dots, n$

Die aussagenlogischen Variablen von A seien $\Sigma = \{p_1, \dots, p_r\}$.

AL-Variablen: (zunächst, später mehr)

$$\begin{aligned} c_j^i & \text{ für } 1 \leq i \leq k, 1 \leq j \leq m, \\ p_j^i & \text{ für } 1 \leq i < k, 1 \leq j \leq r \end{aligned}$$

Für eine Interpretation I fassen wir $I(c_1^i), \dots, I(c_m^i)$ als Binärkodierung einer Zahl n_i , $1 \leq n_i \leq n$, auf.

Insgesamt erhalten wir eine Folge von Zuständen $\pi = n_1, \dots, n_k$.

$I(p_1^i), \dots, I(p_r^i)$ fassen wir als Kodierung einer Teilmenge b_i der Variablen $p_1, \dots, p_r$ auf. Die b_i sind Buchstaben aus dem Vokabular V von $\mathcal{A}$.
Insgesamt erhalten wir ein Wort $b_1, \dots, b_{k-1}$ der Länge $k - 1$.



Aufgabenteilung

Die Formel M_k wird in mehreren Teilen konstruiert

$$M_k \equiv \textit{Init} \wedge \textit{Trans} \wedge \bigvee_{1 \leq i < k} L_i$$

$\textit{Init}$ und $\textit{Trans}$ werden dafür sorgen, daß $n_1, \dots, n_k$ eine Berechnungsfolge von $\mathcal{A}$ ist mit der Folge $b_1 \dots b_{k-1}$ von Kantenmarkierungen.

Zu den L_i kommen wir später.



Die Formel *Init*

Für einen Zustand d von $\mathcal{A}$ mit Binärkode $d_1, \dots, d_m$ setzen wir

$$S_d^i = \bigwedge_{d_j=1} c_j^i \wedge \bigwedge_{d_j=0} \neg c_j^i.$$

Ist d_0 der Anfangszustand von $\mathcal{A}$ dann ist

$$Init \equiv S_{d_0}^1$$



Die Formel Trans

Für einen Buchstaben $b \subseteq \Sigma$ setzen wir

$$B_b^i = \bigwedge_{p_j \in b} p_j^i \wedge \bigwedge_{p_j \notin b} \neg p_j^i.$$

Seien $(d_a^1, d_e^1, b^1), \dots, (d_a^K, d_e^K, b^K)$ alle Kanten des Automaten $\mathcal{A}$, dargestellt durch Ausgangsknoten d_a^j , Endknoten d_e^j und Kantenmarkierung b^j .

$$Trans \equiv \bigwedge_{1 \leq i < k} \bigvee_{1 \leq j \leq K} (S_{d_a^j}^i \wedge S_{d_e^j}^{i+1} \wedge B_{b^j}^i)$$



Die Formeln L_i

$$L_i = Z_i \wedge Akz_i \wedge erfF_i$$

Die Formel L_i sorgt dafür, daß die aus einer erfüllenden Interpretation I extrahierte endliche Berechnungsfolge π

- i -zyklisch ist (via Z_i)
- akzeptierend ist (via Akz_i)
- die Formel F erfüllt (via $erfF_i$)

$$\begin{aligned} Z_i &\equiv \bigwedge_{1 \leq j \leq m} c_j^k \leftrightarrow \bigwedge_{1 \leq j \leq m} c_j^i \\ Akz_i &\equiv Fin_i \vee Fin_{i+1} \dots \vee Fin_{k-1} \end{aligned}$$

wobei

$$F_i \equiv S_{d_1^f}^i \vee \dots \vee S_{d_R^f}^i$$

wenn $d_1^f, \dots, d_R^f$ alle Finalzustände von $\mathcal{A}$ sind.



Neue AL-Variablen

Wir brauchen

für jede Teilformel C von F ,

für jedes i , $1 \leq i < k$ und

jedes j , $1 \leq j \leq k$

eine neue aussagenlogische Variable, die wir mit

$$[C]_i^j$$

bezeichnen.



Zyklische Semantik für LTL Formeln

$$\begin{array}{lll} [C]_i^j & \leftrightarrow & p_l^j \quad \text{falls } C = p_l \in \Sigma \\ [C]_i^j & \leftrightarrow & \neg p_l^j \quad \text{falls } C = \neg p_l \text{ mit } p_l \in \Sigma \\ [C_1 \wedge C_2]_i^j & \leftrightarrow & [C_1]_i^j \wedge [C_2]_i^j \\ [C_1 \vee C_2]_i^j & \leftrightarrow & [C_1]_i^j \vee [C_2]_i^j \\ [\Box C]_i^j & \leftrightarrow & \bigwedge_{j \leq l < k} [C]_i^l \quad \text{falls } j \leq i \\ [\Box C]_i^j & \leftrightarrow & \bigwedge_{j \leq l < k} [C]_i^l \wedge \bigwedge_{i \leq l < j} [C]_i^l \quad \text{falls } i < j \\ [\Diamond C]_i^j & \leftrightarrow & \bigvee_{j \leq l < k} [C]_i^l \quad \text{falls } j \leq i \\ [\Diamond C]_i^j & \leftrightarrow & \bigvee_{j \leq l < k} [C]_i^l \vee \bigvee_{i \leq l < j} [C]_i^l \quad \text{falls } i < j \end{array}$$



Zyklische Semantik für LTL Formeln

Forts.

$$[C_1 \mathbf{U} C_2]_i^j \leftrightarrow \bigvee_{j \leq l < k} ([C_2]_i^l \wedge \bigwedge_{j \leq n < l} [C_1]_i^n) \quad j \leq i$$

$$[C_1 \mathbf{U} C_2]_i^j \leftrightarrow \bigvee_{j \leq l < k} ([C_2]_i^l \wedge \bigwedge_{j \leq n < l} [C_1]_i^n) \vee \bigvee_{i \leq l < j} ([C_2]_i^l \wedge \bigwedge_{j \leq n < k} [C_1]_i^n \wedge \bigwedge_{i \leq n < l} [C_1]_i^n) \quad i < j$$

$$[C_1 \mathbf{V} C_2]_i^j \leftrightarrow [C_2]_i^j \wedge \bigwedge_{i \leq l < k} ([\neg C_2]_i^l \rightarrow \bigvee_{i \leq n < l} [C_1]_i^n) \quad j \leq i$$

$$[C_1 \mathbf{V} C_2]_i^j \leftrightarrow [C_2]_i^j \wedge \bigwedge_{j \leq l < k} ([\neg C_2]_i^l \rightarrow \bigvee_{j \leq n < l} [C_1]_i^n) \wedge \bigwedge_{i \leq l < j} ([\neg C_2]_i^l \rightarrow \bigvee_{i \leq n < l} [C_1]_i^n \vee \bigvee_{j \leq n < k} [C_1]_i^n) \quad i < j$$

$$[X C]_i^j \leftrightarrow [C]_i^{j+1} \quad \text{falls } j < (k-1)$$

$$[X C]_i^{k-1} \leftrightarrow [C]_i^i$$



Die Formeln $\text{erf}F_i$

$$\text{erf}F_i \leftrightarrow [F]_i^1$$



Zusammenfassung

Es gibt eine Berechnungsfolge eines Büchi-Automaten $\mathcal{A}$,
welche die LTL-Formel F erfüllt

genau dann, wenn

es ein k gibt, so daß $M_k(\mathcal{A}, F)$ aussagenlogisch erfüllbar ist.

Die Anzahl der Zustände des Automaten $\mathcal{A}$ ist offensichtlich eine obere Schranke für k .

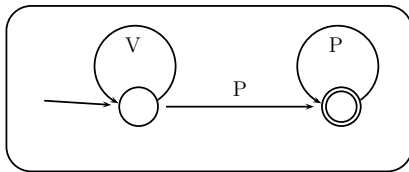
Die kleinste obere Schranke nennt man die Vollständigkeitsschranke (*completeness threshold*) des Problems.



Beispiel

Automat und Formel

$\mathcal{A}_{dbp}$:



$$V = \{\{\}, \{p\}, \{q\}, \{p, q\}\}, \quad P = \{\{p\}, \{p, q\}\}$$

$$A = \Diamond \Box p$$

$$k = 3$$



Beispiel

AL-Variablen (für den Anfang)

$c^1, c^2, c^3, p^1, p^2, q^1, q^2.$

$I(c^i) = \mathbf{0}$ steht für den Anfangszustand

$I(c^i) = \mathbf{1}$ steht für den eindeutigen Endzustand.

$Init = \neg c^1.$



Beispiel

Die Formel *Trans*

Die Menge aller Kanten von $\mathcal{A}_{dbp}$ ist

$$\begin{array}{cccc} (0, 0, \{\}) & (0, 0, \{p\}) & (0, 0, \{q\}) & (0, 0, \{p, q\}) \\ (0, 1, \{p\}) & (0, 1, \{p, q\}) & (1, 1, \{p\}) & (1, 1, \{p, q\}) \end{array}$$

$$\begin{aligned} Trans = & (\neg c^1 \wedge \neg c^2) \vee (\neg c^1 \wedge c^2 \wedge p^1) \vee (c^1 \wedge c^2 \wedge p^1) \\ & \wedge \\ & (\neg c^2 \wedge \neg c^3) \vee (\neg c^2 \wedge c^3 \wedge p^2) \vee (c^2 \wedge c^3 \wedge p^2) \end{aligned}$$

Dabei haben wir schon einige Vereinfachungen vorgenommen und z.B.

$$\begin{aligned} & (\neg c^1 \wedge \neg c^2 \wedge p^1 \wedge q^1) \vee (\neg c^1 \wedge \neg c^2 \wedge \neg p^1 \wedge q^1) \vee \\ & (\neg c^1 \wedge \neg c^2 \wedge p^1 \wedge \neg q^1) \vee (\neg c^1 \wedge \neg c^2 \wedge \neg p^1 \wedge \neg q^1) \end{aligned}$$

äquivalent ersetzt durch

$$(\neg c^1 \wedge \neg c^2).$$



Beispiel

Die Formeln Z_i und Akz_i

$$\begin{aligned}Z_1 &\equiv c^1 \leftrightarrow c^3 \\Z_2 &\equiv c^2 \leftrightarrow c^3 \\Akz_1 &\equiv c^1 \vee c^2 \vee c^3 \\Akz_2 &\equiv c^2 \vee c^3\end{aligned}$$



Beispiel

$\text{Erf}F_1$ und $\text{Erf}F_2$

$$F \equiv \Diamond \Box p.$$

$$\begin{aligned} [F]_1^1 &\equiv [\Box p]_1^1 \vee [\Box p]_1^2 \\ [\Box p]_1^1 &\equiv [\rho]_1^1 \wedge [\rho]_1^2 \\ &\equiv p^1 \wedge p^2 \\ [\Box p]_1^2 &\equiv [\rho]_1^2 \wedge [\rho]_1^1 \\ &\equiv p^2 \wedge p^1 \end{aligned}$$

Insgesamt also $[F]_1^1 \leftrightarrow p^1 \wedge p^2$

$$\begin{aligned} [F]_2^1 &\equiv [\Box p]_2^1 \vee [\Box p]_2^2 \\ [\Box p]_2^1 &\equiv [\rho]_2^1 \wedge [\rho]_2^2 \\ &\equiv p^1 \wedge p^2 \\ [\Box p]_2^2 &\equiv [\rho]_2^2 \\ &\equiv p^2 \end{aligned}$$

Insgesamt also $[F]_2^1 \leftrightarrow p^2$.



Beispiel

L_1 und L_2

$$\begin{aligned} L_1 &\leftrightarrow (c^1 \leftrightarrow c^3) \wedge (c^1 \vee c^2 \vee c^3) \wedge p^1 \wedge p^2 \\ &\leftrightarrow (c^1 \leftrightarrow c^3) \wedge (c^2 \vee c^3) \wedge p^1 \wedge p^2 \end{aligned}$$

$$\begin{aligned} L_2 &\leftrightarrow (c^2 \leftrightarrow c^3) \wedge (c^2 \vee c^3) \wedge p^2 \\ &\leftrightarrow c^2 \wedge c^3 \wedge p^2 \end{aligned}$$



Beispiel

M_3

$$\begin{aligned} & \neg c^1 \wedge \\ & (\neg c^1 \wedge \neg c^2) \vee (\neg c^1 \wedge c^2 \wedge p^1) \vee (c^1 \wedge c^2 \wedge p^1) \\ & \wedge \\ & (\neg c^2 \wedge \neg c^3) \vee (\neg c^2 \wedge c^3 \wedge p^2) \vee (c^2 \wedge c^3 \wedge p^2) \\ & \wedge \\ & (((c^1 \leftrightarrow c^3) \wedge (c^2 \vee c^3) \wedge p^1 \wedge p^2) \\ & \vee \\ & (c^2 \wedge c^3 \wedge p^2)) \end{aligned}$$

Vereinfacht:

$$M_3 \equiv \neg c^1 \wedge c^2 \wedge c^3 \wedge p^1 \wedge p^2.$$



Beispiel

Diskussion

Die Formel $M_3 \equiv \neg c^1 \wedge c^2 \wedge c^3 \wedge p^1 \wedge p^2$ ist offensichtlich erfüllbar.

Also gibt es eine Berechnungsfolge in $\mathcal{A}_{dbp}$, für die $\Diamond \Box p$ wahr ist.

Wir wissen sogar, daß für jede Berechnungsfolge von $\mathcal{A}_{dbp}$ diese Formel gilt.

