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.

Notationsvariante

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

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

Notationsvariante

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

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

bedeutet:

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

Notationsvariante

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

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

bedeutet:

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

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

Notationsvariante

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

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

bedeutet:

für jede akzepierende 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 Berechungsfolgen

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 $(s_m)_{m \geq n}$

Semantik von LTL für Berechungsfolgen

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:

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

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 Berechungsfolgen

Definition

1. Eine endliche Berechnungsfolge $s_0, \ldots, s_k$ heißt *i-zyklisch*, falls $0 \leq i < k$ und $s_k = s_i$.

Zyklische Berechungsfolgen

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 $0 \leq i < k$.

Zyklische Berechungsfolgen

Definition

1. Eine endliche Berechnungsfolge $s_0, \ldots, s_k$ heißt *i-zyklisch*, falls $0 \leq i < k$ und $s_k = s_i$.
2. Eine endliche Berechnungsfolge $s_0, \ldots, s_k$ heißt *zyklisch*, falls sie *i-zyklisch* ist für ein i mit $0 \leq i < k$.
3. Eine i-zyklische Berechnungsfolge $s_0, \ldots, s_k$ heißt akzeptierend, wenn unter den Zuständen $s_i, \ldots, s_k$ mindestens ein Endzustand vorkommt.

Zyklische Berechungsfolgen

Definition

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

Lemma

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

Lemma

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

Falls es eine akzeptierende Berechungsfolge 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 $s \models F$

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

$$\begin{array}{lll} M_k \text{ ist erfüllbar} & \text{gdw} & \text{es gibt eine zyklische Berechnungsfolge } s \\ & & \text{der Länge } k \text{ mit } s \models F \end{array}$$

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

Die AL-Variablen in M_k

Die Zustände von $\mathcal{A}$ n seien $1, 2, \ldots n$
Die aussagenlogischen Variablen von A seien $\Sigma = \{p_1, \ldots, p_r\}$.

Die AL-Variablen in M_k

Die Zustände von $\mathcal{A}$ n seien $1, 2, \ldots n$
Die aussagenlogischen Variablen von A seien $\Sigma = \{p_1, \ldots, p_r\}$.

AL-Variablen: (zunächst,später mehr)

c^i_j für $1 \leq i \leq k$, $1 \leq j \leq m$,
p^i_j für $1 \leq i < k$, $1 \leq j \leq r$

Die AL-Variablen in M_k

Die Zustände von $\mathcal{A}$ n seien $1, 2, \dots n$
Die aussagenlogischen Variablen von A seien $\Sigma = \{p_1, \dots, p_r\}$.

AL-Variablen: (zunächst,später mehr)

c^i_j für $1 \leq i \leq k$, $1 \leq j \leq m$,
p^i_j für $1 \leq i < k$, $1 \leq j \leq r$

Für eine Interpretation I fassen wir $I(c^i_1), \dots, I(c^i_m)$ als Binärkodierung einer Zahl n_i, $1 \leq n_i \leq n$, auf.
Ingesamt erhalten wir eine Folge von Zuständen $\pi = n_1, \dots, n_k$.

$I(p^i_1), \dots, I(p^i_r)$ 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 eine Wort $b_1, \dots, b_{k-1}$ der Länge $k - 1$.

Aufgabenteilung

Die Formel M_k wird in mehreren Teilen konstruiert

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

Aufgabenteilung

Die Formel M_k wird in mehreren Teilen konstruiert

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

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

Aufgabenteilung

Die Formel M_k wird in mehreren Teilen konstruiert

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

Init und *Trans* werden dafür sorgen, daß daß $n_1, \dots, n_k$ eine Berechungsfolge 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 $d_1, \ldots, d_m$ setzen wir

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

Die Formel Init

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

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

Die Formel **Trans**

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

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

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

$$Trans \equiv \bigwedge_{1 \leq i < k} \bigvee_{1 \leq j \leq K} (S^i_{d_a^j} \wedge S^{i+1}_{d_e^j} \wedge B^i_{b^j})$$

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)

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 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 $erfF_i$)

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 $erfF_i$)

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 $erfF_i$)

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

wobei

$$F_i \equiv S^i_{d_1^f} \vee \ldots \vee S^i_{d_R^f}$$

wenn $d_1^f, \ldots, d_R^f$ 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

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

Zyklische Semantik für LTL Formeln

Forts.

$$
\begin{array}{lcll}
[C_1 \ \mathbf{U} \ C_2]_i^j & \leftrightarrow & \bigvee_{j \le l < k}([C_2]_i^l \wedge \bigwedge_{j \le n < l}[C_1]_i^n) & j \le i \\
[C_1 \ \mathbf{U} \ C_2]_i^j & \leftrightarrow & \bigvee_{j \le l < k}([C_2]_i^l \wedge \bigwedge_{j \le n < l}[C_1]_i^n) \vee & \\
 & & \bigvee_{i \le l < j}([C_2]_i^l \wedge \bigwedge_{j \le n < k}[C_1]_i^n \wedge \bigwedge_{i \le n < l}[C_1]_i^n) & i < j
\end{array}
$$

$$
\begin{array}{lcll}
[C_1 \ \mathbf{V} \ C_2]_i^j & \leftrightarrow & [C_2]_i^j \wedge \bigwedge_{i \le l < k}([\neg C_2]_i^l \rightarrow \bigvee_{i \le n < l}[C_1]_i^n) & j \le i \\
[C_1 \ \mathbf{V} \ C_2]_i^j & \leftrightarrow & [C_2]_i^j \wedge \bigwedge_{j \le l < k}([\neg C_2]_i^l \rightarrow \bigvee_{j \le n < l}[C_1]_i^n) \wedge & \\
 & & \bigwedge_{i \le l < j}([\neg C_2]_i^l \rightarrow \bigvee_{i \le n < l}[C_1]_i^n \vee \bigvee_{j \le n < k}[C_1]_i^n) & i < j
\end{array}
$$

$$
\begin{array}{lcll}
[X \ C]_i^j & \leftrightarrow & [C]_i^{j+1} & \text{falls } j < (k-1) \\
[X \ C]_i^{k-1} & \leftrightarrow & [C]_i^i &
\end{array}
$$

Die Formeln $erfF_i$

$$erfF_i \leftrightarrow [F]_i^1$$

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.

Zusammenfassung

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

genau dann, wenn

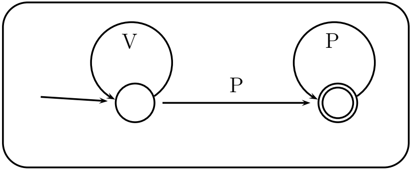
es ein k gibt, so daß $M_k(\mathcal{A}, F)$ aussagenlogisch erfüllbar ist.

Die Anzahl der Zustände des Automaten $\mathcal{A}$ ist offensichtlich eine obere Schranke für k.
Die kleinste obere Schranke nennt man die Vollständigkeitsschranke (*completeness threshhold*) 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.

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

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

Beispiel

Die Formel Trans

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

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

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

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

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

äquivalent ersetzt durch

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

Beispiel

Die Formeln Z_i und Akz_i

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

Beispiel

$ErfF_1$ und $ErfF_2$

$F \equiv \Diamond\Box p$.

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

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

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

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

$$\neg c^1 \wedge$$
$$(\neg c^1 \wedge \neg c^2) \vee (\neg c^1 \wedge c^2 \wedge p^1) \vee (c^1 \wedge c^2 \wedge p^1)$$
$$\wedge$$
$$(\neg c^2 \wedge \neg c^3) \vee (\neg c^2 \wedge c^3 \wedge p^2) \vee (c^2 \wedge c^3 \wedge p^2)$$
$$\wedge$$
$$(((c^1 \leftrightarrow c^3) \wedge (c^2 \vee c^3) \wedge p^1 \wedge p^2)$$
$$\vee$$
$$(c^2 \wedge c^3 \wedge p^2))$$

Beispiel

M_3

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

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.

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.

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.