



Informatik III - Tutorium IX & X (SR -107)

Tut Nr. 8 – Üb7, Entscheid- und Aufzählbarkeit

David Münch

Universität Karlsruhe (TH)
Institut für Informatik
IAKS Beth

17. Dezember 2008



Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825



Inhaltsverzeichnis

1 Auftakt



Inhaltsverzeichnis

- 1 Auftakt
- 2 Lernziele



Inhaltsverzeichnis

- 1 Auftakt
- 2 Lernziele
- 3 Themen
 - Übungsblatt 7
 - Entscheidbarkeit
 - Komplexitätstheorie



Inhaltsverzeichnis

- 1 Auftakt
- 2 Lernziele
- 3 Themen
 - Übungsblatt 7
 - Entscheidbarkeit
 - Komplexitätstheorie
- 4 Abspann



Organisatorisches

Email: muenchdavid@gmail.com

<https://www.stud.uni-karlsruhe.de/~uhbro/>

Tutorium 09: Mittwochs 8:00 Uhr - Raum -107

Tutorium 10: Mittwochs 9:45 Uhr - Raum -107

Übungsblattabgabe Donnerstag.



Was wollen wir heute erreichen?



Was wollen wir heute erreichen?

- Kapitel **Berechenbarkeitstheorie** abschließen.



Was wollen wir heute erreichen?

- Kapitel **Berechenbarkeitstheorie** abschließen.
- Einführung in die Komplexitätstheorie.



Aufgabe 1

Zeigen oder widerlegen Sie, dass es einen Index $n \in \mathbb{N}$ gibt mit

$$\varphi_n(i, x) = \begin{cases} \varphi_i^{(n)}(x), & \text{falls } n \text{ Index eines **while**-Programms ist} \\ \perp, & \text{sonst} \end{cases}$$

wobei $\varphi_i^{(n)}(x)$ bedeutet, dass die Funktion φ_i n -mal auf die Eingabe x angewandt wird.



Lösung:

Es sei $\Psi(k, i, x) = \text{„if korrekt}(i)=0 \text{ then } \perp \text{ else } \phi_i^k(x)\text{“}$. Diese Funktion ist berechenbar. Daraus folgt es existiert ein Index $e \in \mathbb{N}_0$ mit $\phi_e = \Psi$. Nach dem $s - m - n$ -Theorem gibt es eine totale berechenbare Funktion s mit $\phi_e(k, i, x) = \phi_{s(e,k)}(i, x)$.

Setze $g(k) := s(e, k)$.

Da g auch total und berechenbar ist, gibt es nach dem Rekursionssatz ein $n \in \mathbb{N}_0$ mit $\phi_n = \phi_{g(n)}$.

Insgesamt gilt also: Es existiert ein Index $n \in \mathbb{N}_0$ mit

$$\phi_n(i, x) = \phi_{g(n)}(i, x) = \phi_{s(e,k)}(i, x) = \phi_e(k, i, x)$$

$$= \Psi(k, i, x) = \begin{cases} \phi_i^{(n)}(x), & \text{falls korrekt}(n) = 1 \\ \perp, & \text{sonst} \end{cases}$$



Aufgabe 2

Gegeben sei das folgende rekursive While-Programm:

M_{McC} :

if $x_1 > 100$ **then**

$x_1 := x_1 - 10;$

else begin

$x_1 := x_1 + 11;$

$M_{McC};$

$M_{McC};$

end;

- 1 Welche Funktionstransformation $T : \mathcal{F} \rightarrow \mathcal{F}$, wobei $\mathcal{F} = \{f : \mathbb{N}_0 \rightarrow \mathbb{N}_0 \mid f \text{ partiell oder total}\}$, wird durch das rekursive While-Programm M_{McC} definiert?
- 2 Berechnen sie den minimalen Fixpunkt von T (und damit die von M_{McC} realisierte Funktion)

**Hinweis:**

$$f(f(x + 11)) = (f \circ f)(x + 11) = f^2(x + 11) \neq (f(x + 11))^2$$

**Hinweis:**

$$f(f(x + 11)) = (f \circ f)(x + 11) = f^2(x + 11) \neq (f(x + 11))^2$$

Lösung:

i.) Man sieht dass M_{McC} die Funktionstransformation

$$(Tf)(x) = \begin{cases} x - 10 & , \text{ falls } x > 100 \\ f(f(x + 11)) & , \text{ falls } x \leq 100 \end{cases}$$

definiert.

Es handelt sich hier um die McCarthy-91-Funktion.

http://en.wikipedia.org/wiki/McCarthy_91_function



Übungsblatt 7

ii.) Wie in der Vorlesung gezeigt verwendet man das Verfahren aus dem Kleene'schen Fixpunktsatz um den minimalen Fixpunkt von T zu finden. Dazu beginnen wir mit der leeren Funktion $f_0 \equiv \perp$ und berechnen

$$f_1(x) = (Tf_0)(x) = \begin{cases} x - 10 & , \text{ falls } x > 100 \\ \perp & , \text{ falls } x \leq 100 \end{cases} .$$

Es gilt nun für $f_2 = Tf_1$:

$$f_2(100) = (Tf_1)(100) = f_1(f_1(100 + 11)) = f_1(101) = 91,$$

also

$$f_2(x) = (Tf_1)(x) = \begin{cases} x - 10 & , \text{ falls } x > 100 \\ 91 & , \text{ falls } x = 100 \\ \perp & , \text{ falls } x \leq 99 \end{cases} .$$



Genauso erhalten wir

$$f_3(x) = (Tf_2)(x) = \begin{cases} x - 10 & , \text{ falls } x > 100 \\ 91 & , \text{ falls } 99 \leq x \leq 100 \\ \perp & , \text{ sonst} \end{cases}$$



Übungsblatt 7

Genauso erhalten wir

$$f_3(x) = (Tf_2)(x) = \begin{cases} x - 10 & , \text{ falls } x > 100 \\ 91 & , \text{ falls } 99 \leq x \leq 100 \\ \perp & , \text{ sonst} \end{cases}$$

und

$$f_4(x) = (Tf_3)(x) = \begin{cases} x - 10 & , \text{ falls } x > 100 \\ 91 & , \text{ falls } 98 \leq x \leq 100 \\ \perp & , \text{ sonst} \end{cases} .$$



Genauso erhalten wir

$$f_3(x) = (Tf_2)(x) = \begin{cases} x - 10 & , \text{ falls } x > 100 \\ 91 & , \text{ falls } 99 \leq x \leq 100 \\ \perp & , \text{ sonst} \end{cases}$$

und

$$f_4(x) = (Tf_3)(x) = \begin{cases} x - 10 & , \text{ falls } x > 100 \\ 91 & , \text{ falls } 98 \leq x \leq 100 \\ \perp & , \text{ sonst} \end{cases} .$$

Das legt die Vermutung nahe dass der minimale Fixpunkt von T

$$f_*(x) = \begin{cases} x - 10 & , \text{ falls } x > 100 \\ 91 & , \text{ falls } x \leq 100 \end{cases}$$

ist.



Übungsblatt 7

f_* ist in der Tat Fixpunkt, da

$$(Tf_*)(x) = \begin{cases} x - 10 & , \text{ falls } x > 100 \\ f_*(f_*(x + 11)) & , \text{ falls } x \leq 100 \end{cases}$$

da für $x = 100$

$$(Tf_*)(x) = f_*(f_*(100 + 11)) = f_*(101) = 91$$

und für $90 \leq x < 100$

$$(Tf_*)(x) = f_*(f_*(x + 11)) = f_*(\underbrace{x + 1}_{\leq 100}) = 91$$

und für $x < 90$

$$(Tf_*)(x) = f_*(f_*(\underbrace{x + 11}_{\leq 100})) = f_*(91) = 91.$$

Also gilt

$$Tf_* = f_*$$



Entscheidbarkeit



Definition:

- Eine Menge $A \subseteq \mathbb{N}^k$ ist **entscheidbar**, **berechenbar**, **lösbar**, **rekursiv** \Leftrightarrow ihre charakteristische Funktion χ_A berechenbar ist, wobei gilt:

$$\chi_A : A^* \rightarrow \{0, 1\} \text{ mit } \chi_A(w) = \begin{cases} 1, & \text{falls } w \in A \\ 0, & \text{sonst.} \end{cases}$$



Definition:

- Eine Menge $A \subseteq \mathbb{N}^k$ ist **entscheidbar, berechenbar, lösbar, rekursiv** \Leftrightarrow ihre charakteristische Funktion χ_A berechenbar ist, wobei gilt:

$$\chi_A : A^* \rightarrow \{0, 1\} \text{ mit } \chi_A(w) = \begin{cases} 1, & \text{falls } w \in A \\ 0, & \text{sonst.} \end{cases}$$

- Eine Menge A ist **semi-entscheidbar, rekursiv aufzählbar** \Leftrightarrow Funktion χ_A^* berechenbar ist, wobei gilt:

$$\chi_A^*(w) = \begin{cases} 1, & \text{falls } w \in A \\ \text{undefiniert,} & \text{sonst.} \end{cases}$$



Aufgabe

Sei $A \subseteq \mathbb{N}_0$ eine entscheidbare Menge. Zeigen Sie, dass

$$B = \{x + 2y^2 + 17 + 11^x \mid x, y \in A\}$$

entscheidbar ist.

**Lösung:**

Es gilt:

$$z \in B \Leftrightarrow \exists x, y \in A : 0 \leq x, y \leq z \wedge z = x + 2y^2 + 17 + 11^x.$$

Da A entscheidbar, ist ihre charakteristische Funktion χ_A berechenbar. Betrachte folgendes while-Programm:

```

z := x1  x1 = 0;
for      x = 0 to z do
    for    y = 0 to z do
        if (χA(x) = 1) ∧ (χA(y) = 1) ∧ (z = x + 2y2 + 17 + 11x)
            then x1 := 1
        end
    end
end

```

Das while-Programm berechnet χ_B , d.h. B ist entscheidbar.



Satz

Jede rekursive Menge A ist aufzählbar.



Satz

Jede rekursive Menge A ist aufzählbar.

Satz

Eine Menge A ist rekursiv genau dann, wenn sie und ihr Komplement rekursiv aufzählbar sind.



Definition:

Sei $\varphi : \mathbb{N}^k \rightarrow \mathbb{N}^m$. Dann ist

$$D(\varphi) = \{x \in \mathbb{N}^k \mid \varphi(x) \downarrow\} = \varphi^{-1}(\mathbb{N}^m)$$

der **Definitionsbereich** und

$$\text{Bild}(\varphi) = \varphi(\mathbb{N}^k)$$

der **Bildbereich** von φ .



Satz

A rekursiv aufzählbar $\Leftrightarrow A = D(\varphi) \Leftrightarrow A = \text{Bild}(\rho)$
wobei φ und ρ beliebig, aber berechenbar.



Definition: respektiert Funktionen

Eine Teilmenge $M \subset \mathbb{N}_0$ respektiert Funktionen, falls sie für jede berechenbare Funktion alle Indizes oder keinen enthält, das heißt falls

$$\forall i \in M \forall j \in \mathbb{N}_0 : \varphi_i = \varphi_j \Rightarrow j \in M.$$



Definition: respektiert Funktionen

Eine Teilmenge $M \subset \mathbb{N}_0$ repektiert Funktionen, falls sie für jede berechenbare Funktion alle Indizes oder keinen enthält, das heißt falls

$$\forall i \in M \forall j \in \mathbb{N}_0 : \varphi_i = \varphi_j \Rightarrow j \in M.$$

Satz von Rice

Die einzigen rekursiven Teilmengen von \mathbb{N}_0 , die Funktionen respektieren sind \emptyset und \mathbb{N}_0 .



Aufgabe

Sei g ein Funktion mit $D(g) = \{1, 2, \dots, n\} \neq \emptyset$.

Zeigen Sie:

$B = \{i \in \mathbb{N}_0 \mid g \leq \varphi_i\}$ ist rekursiv aufzählbar, aber nicht entscheidbar.



Lösung:

Betrachte folgendes **while**-Programm:

```
 $P_j :=$     begin  $x_2 = x_1; x_1 := 1;$   
              if  $\Phi(x_2, 1) \neq g_1$  then  $x_1 := 0;$   
              :  
              if  $\Phi(x_2, n) \neq g_n$  then  $x_1 := 0;$   
              while  $x_1 = 0$  do begin end  
              end
```

Es ist dann $D(\varphi_j) = B$, weswegen B rekursiv aufzählbar ist. Da für $n > 0$ aber $g \neq \perp$, ist B eine echte, nichtleere Teilmenge von \mathbb{N}_0 , die Funktionen respektiert.

Deshalb ist B nach dem Satz von Rice nicht entscheidbar.



Aufgabe

Beweisen oder widerlegen Sie, dass die folgende Mengen rekursiv aufzählbar sind:

- 1 $A := \{i \in \mathbb{N}_0 \mid \varphi_i \text{ hat einen Fixpunkt}\}$
- 2 $B := \overline{A}$



Lösung:

1) φ_i hat einen Fixpunkt $\Leftrightarrow \exists x \in \mathbb{N}_0 : \varphi_i(x) = x$.

Sei $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0^2$ eine berechenbare Bijektion.

Betrachte folgendes while-Programm mit Index e :

```

 $x_2 := 0;$      $(y, z) := f(x_2);$ 
while       $(\Phi(x_1, y) \uparrow \text{ in } z \text{ Schritten}) \vee (\Phi(x_1, y) \neq y)$ 
do         $x_2 := x_2 + 1; (y, z) := f(x_2);$ 
end

```

Dann ist $D(\varphi_e) = A$, d.h. A ist rekursiv aufzählbar.



Lösung:

1) φ_i hat einen Fixpunkt $\Leftrightarrow \exists x \in \mathbb{N}_0 : \varphi_i(x) = x$.

Sei $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0^2$ eine berechenbare Bijektion.

Betrachte folgendes while-Programm mit Index e :

```

 $x_2 := 0;$      $(y, z) := f(x_2);$ 
while       $(\Phi(x_1, y) \uparrow \text{ in } z \text{ Schritten}) \vee (\Phi(x_1, y) \neq y)$ 
do         $x_2 := x_2 + 1; (y, z) := f(x_2);$ 
end

```

Dann ist $D(\varphi_e) = A$, d.h. A ist rekursiv aufzählbar.

2) Wäre B rekursiv aufzählbar, dann wären A und B sogar rekursiv. Da A aber offensichtlich Funktionen respektiert (Definition von Satz RICE, im Skript Seite 127) und sowohl $A \neq \emptyset$ (id besitzt Fixpunkt) als auch $A \neq \mathbb{N}_0$ (\perp besitzt keinen Fixpunkt), ist dies nach dem Satz von Rice nicht der Fall.



Definition: reduzierbar

Eine Menge A heißt auf eine Menge B reduzierbar, kurz $A \leq B$, falls es eine berechenbare totale Funktion f gibt, mit

$$f(A) \subset B \text{ und } f(\overline{A}) \subset \overline{B},$$

das heißt mit $A = f^{-1}(B)$.



Lemma

Falls $A \leq B$, dann gilt:



Lemma

Falls $A \leq B$, dann gilt:

- 1 Ist B rekursiv, dann auch A .



Lemma

Falls $A \leq B$, dann gilt:

- 1 Ist B rekursiv, dann auch A .
- 2 Ist A nicht rekursiv, dann ist auch B nicht rekursiv.



Lemma

Falls $A \leq B$, dann gilt:

- 1 Ist B rekursiv, dann auch A .
- 2 Ist A nicht rekursiv, dann ist auch B nicht rekursiv.
- 3 Ist B rekursiv aufzählbar, dann auch A .



Lemma

Falls $A \leq B$, dann gilt:

- 1 Ist B rekursiv, dann auch A .
- 2 Ist A nicht rekursiv, dann ist auch B nicht rekursiv.
- 3 Ist B rekursiv aufzählbar, dann auch A .
- 4 Ist A nicht rekursiv aufzählbar, dann ist auch B nicht rekursiv aufzählbar.



Aufgabe

Gegeben seien die Sprachen $\mathcal{L}_Q := \{(a, b) \mid b = a^2, a \in \mathbb{N}\}$ und $\mathcal{L}_M := \{(a_1, a_2, b) \in \mathbb{N}^3 \mid b = a_1 * a_2\}$. Reduzieren Sie die Sprache \mathcal{L}_Q auf \mathcal{L}_M .



Lösung:

Die Sprache der Quadratzahlen \mathcal{L}_Q lässt sich auf die Sprache aller Multiplikationen mit 2 Faktoren \mathcal{L}_M reduzieren, da über die Abbildung $f : \mathbb{N}^2 \rightarrow \mathbb{N}^3$, $(a, b) \mapsto (a, a, b)$ jedes Wort aus \mathcal{L}_Q in ein Wort aus \mathcal{L}_M abgebildet werden kann und jedes Bild dieser Funktion f wiederum ein Urbild in \mathcal{L}_Q besitzt.

Durch die Reduktion zeigt man, dass die charakteristische Funktion von \mathcal{L}_Q mit Hilfe der charakteristischen Funktion von \mathcal{L}_M berechnet werden kann: nämlich man schaut, ob ein Tupel (a, b) in \mathcal{L}_Q liegt, indem man es zuerst mit f transformiert, also auf das Tupel (a, a, b) abbildet, und dann prüft, ob dieses Tupel (a, a, b) in \mathcal{L}_M liegt. Die charakteristische Funktion $\chi_{\mathcal{L}_Q}$ berechnet man somit: $\chi_{\mathcal{L}_Q}(a, b) = \chi_{\mathcal{L}_M}(f(a, b)) = \chi_{\mathcal{L}_M}(a, a, b)$. D.h. man hat eigentlich gezeigt, dass, wenn man entscheiden kann, ob b das Produkt von 2 Zahlen a_1 und a_2 ist, dann kann man auch entscheiden, ob b die Quadratzahl a^2 ist.



Komplexitätsklassen



Sei M eine TM mit Eingabealphabet A . Die Rechenzeit von M wird durch die Funktion $t_M : A^* \rightarrow \mathbb{N}_0$ beschrieben.

$$t_M(\alpha) = \begin{cases} \text{minimale Zahl von Rechenschritten,} & \text{falls } \alpha \in L(M) \\ \text{die } M \text{ braucht, um } \alpha \text{ zu akzeptieren,} & \\ 0, & \text{falls } \alpha \notin L(M). \end{cases}$$



Mit diesen Rechenzeiten kommen wir nun zu folgenden Komplexitätsklassen:

DTIME $(f(n)) =$
 $\{L \subset A^* \mid L = L(M), \text{ wobei } M \text{ eine deterministische}$
 $\text{TM mit } t_M(\alpha) = \mathcal{O}(f(|\alpha|))\}$



Mit diesen Rechenzeiten kommen wir nun zu folgenden Komplexitätsklassen:

DTIME $(f(n)) =$

$\{L \subset A^* \mid L = L(M), \text{ wobei } M \text{ eine deterministische TM mit } t_M(\alpha) = \mathcal{O}(f(|\alpha|))\}$

NTIME $(f(n)) =$

$\{L \subset A^* \mid L = L(M), \text{ wobei } M \text{ eine nichtdeterministische TM mit } t_M(\alpha) = \mathcal{O}(f(|\alpha|))\}$



Mit diesen Rechenzeiten kommen wir nun zu folgenden Komplexitätsklassen:

$$\mathbf{DTIME}(f(n)) =$$

$\{L \subset A^* \mid L = L(M), \text{ wobei } M \text{ eine deterministische TM mit } t_M(\alpha) = \mathcal{O}(f(|\alpha|))\}$

$$\mathbf{NTIME}(f(n)) =$$

$\{L \subset A^* \mid L = L(M), \text{ wobei } M \text{ eine nichtdeterministische TM mit } t_M(\alpha) = \mathcal{O}(f(|\alpha|))\}$

$$\mathbf{P} = \bigcup_{k=1}^{\infty} \mathbf{DTIME}(n^k)$$



Mit diesen Rechenzeiten kommen wir nun zu folgenden Komplexitätsklassen:

$$\mathbf{DTIME}(f(n)) =$$

$\{L \subset A^* \mid L = L(M), \text{ wobei } M \text{ eine deterministische TM mit } t_M(\alpha) = \mathcal{O}(f(|\alpha|))\}$

$$\mathbf{NTIME}(f(n)) =$$

$\{L \subset A^* \mid L = L(M), \text{ wobei } M \text{ eine nichtdeterministische TM mit } t_M(\alpha) = \mathcal{O}(f(|\alpha|))\}$

$$\mathbf{P} = \bigcup_{k=1}^{\infty} \mathbf{DTIME}(n^k)$$

$$\mathbf{NP} = \bigcup_{k=1}^{\infty} \mathbf{NTIME}(n^k)$$



Mit diesen Rechenzeiten kommen wir nun zu folgenden Komplexitätsklassen:

$$\mathbf{DTIME}(f(n)) =$$

$\{L \subset A^* \mid L = L(M), \text{ wobei } M \text{ eine deterministische TM mit } t_M(\alpha) = \mathcal{O}(f(|\alpha|))\}$

$$\mathbf{NTIME}(f(n)) =$$

$\{L \subset A^* \mid L = L(M), \text{ wobei } M \text{ eine nichtdeterministische TM mit } t_M(\alpha) = \mathcal{O}(f(|\alpha|))\}$

$$\mathbf{P} = \bigcup_{k=1}^{\infty} \mathbf{DTIME}(n^k)$$

$$\mathbf{NP} = \bigcup_{k=1}^{\infty} \mathbf{NTIME}(n^k)$$

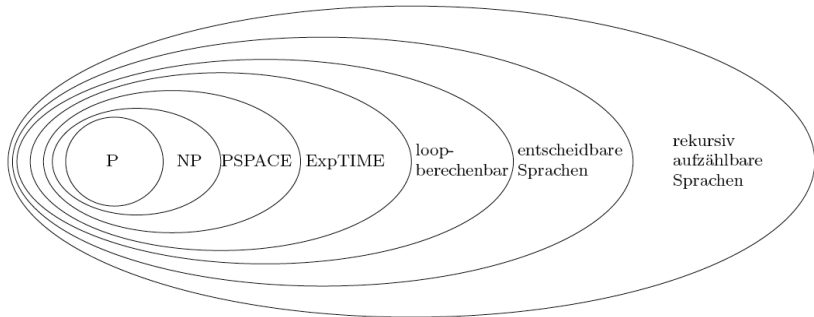
$$\mathbf{EXPTIME} = \bigcup_{k=1}^{\infty} \mathbf{DTIME}(2^n)$$



Zentrale Frage: $\mathcal{P} = \mathcal{NP}$?



Inklusion der Komplexitätsklassen





Definition: polynomial reduzierbar

Seien L_a und L_b Sprachen über A . Dann heißt L_a auf L_b polynomial reduzierbar, kurz $L_a \leq_p L_b$, falls es eine in polynomialer Laufzeit berechenbare totale Funktion $f : A^* \rightarrow A^*$ gibt mit $f^{-1}L_b = L_a$ bzw. $\forall a \in A^* : a \in L_a \Leftrightarrow f(a) \in L_b$.



Definition: polynomial reduzierbar

Seien L_a und L_b Sprachen über A . Dann heißt L_a auf L_b polynomial reduzierbar, kurz $L_a \leq_p L_b$, falls es eine in polynomialer Laufzeit berechenbare totale Funktion $f : A^* \rightarrow A^*$ gibt mit $f^{-1}L_b = L_a$ bzw. $\forall a \in A^* : a \in L_a \Leftrightarrow f(a) \in L_b$.

Lemma

Ist $L_a \leq_p L_b$ und L_b in \mathcal{P} (oder \mathcal{NP}), ist auch L_a in \mathcal{P} (oder \mathcal{NP}).



Definition: \mathcal{NP} -vollständig

Eine Sprache L heißt **\mathcal{NP} -vollständig**, falls gilt:

- 1 $L \in \mathcal{NP}$ und
- 2 für alle $L' \in \mathcal{NP}$ gilt: $L' \leq_p L$



SAT

Das Erfüllbarkeitsproblem **SAT** (satisfiability) ist das “erste” \mathcal{NP} -vollständige Problem (Steven Cook hat es 1971 bewiesen).

Gegeben: Menge U von Variablen, Menge C von Klauseln über U

Frage: Existiert eine Wahrheitsbelegung von U , so dass C erfüllt wird, d.h. dass alle Klausel aus C den Wahrheitswert *wahr* annehmen?



Aufgabe

Geben Sie eine nicht-triviale Instanz von SAT an und überprüfen Sie, ob es eine Wahrheitsbelegung gibt.



Aufgabe

Geben Sie eine nicht-triviale Instanz von SAT an und überprüfen Sie, ob es eine Wahrheitsbelegung gibt.

Wer keine nicht-triviale Instanz im Kopf hat:

$$U = \{a, b, c, d, e, f\}$$

$$C = \{\bar{a} \vee \bar{b} \vee c \vee d \vee \bar{e} \vee f, \bar{a} \vee b \vee \bar{c} \vee \bar{d} \vee e \vee \bar{f}, \bar{a} \vee b \vee \bar{c} \vee d \vee \bar{e} \vee f, \\ a \vee b \vee \bar{c} \vee \bar{d} \vee e \vee \bar{f}, a \vee \bar{b} \vee c \vee d \vee \bar{e} \vee f, \bar{a} \vee b \vee \bar{c} \vee d \vee e \vee f, \\ a \vee b \vee c \vee d \vee e \vee f\}$$



Aufgabe

Wie könnte eine NTM, die das Entscheidungsproblem von SAT löst, arbeiten? Schätzen Sie grob den Arbeitsaufwand ab.



Aufgabe

Überlegen Sie, wie eine DTM, die das Entscheidungsproblem von SAT löst, arbeiten könnte. Schätzen Sie grob den Arbeitsaufwand ab.



Reflexion

Was haben wir heute gelernt?



Reflexion

Was haben wir heute gelernt?

- Kapitel **Entscheidbarkeit** abgeschlossen



Reflexion

Was haben wir heute gelernt?

- Kapitel **Entscheidbarkeit** abgeschlossen
- Unterschied zwischen \mathcal{P} und \mathcal{NP}



Reflexion

Was haben wir heute gelernt?

- Kapitel **Entscheidbarkeit** abgeschlossen
- Unterschied zwischen \mathcal{P} und \mathcal{NP}
- Polynomiale Transformierbarkeit kennen gelernt



Noch Fragen?



Vorschau



Vorschau

- Nächster Termin am **14.01.2008!**

Bis zum nächsten Mal

