Info III Tutorium

Thomas Pajor

ITI Sanders

30. Januar 2007

Aufgabe

Aufgabe 1

Beweisen Sie: Es kann keinen (Sprach–)Compiler geben, der bezüglich der erzeugten Maschinencodegröße optimal ist, also immer den kürzest möglichen Code erzeugt.

Wdh.: Probleme, Instanzen, Sprachen...

Definition (Problem)

Ein *Problem* Π ist charakterisiert durch

- Eine allgemeine Beschreibung aller vorkommenden Parameter
- Eine genaue Beschreibung der Eigenschaften, die die Lösung haben soll

Ein *Entscheidungsproblem* hat als Lösungen nur „Ja“ oder „Nein“.

Definition (Instanz)

Eine *Instanz* $I \in \Pi$ (Problembeispiel von Π) erhalten wir, indem wir die Parameter von Π festlegen.

Wdh.: Probleme, Instanzen, Sprachen...

Definition (Problem)

Ein *Problem* Π ist charakterisiert durch

- Eine allgemeine Beschreibung aller vorkommenden Parameter
- Eine genaue Beschreibung der Eigenschaften, die die Lösung haben soll

Ein *Entscheidungsproblem* hat als Lösungen nur „Ja“ oder „Nein“.

Definition (Instanz)

Eine *Instanz* $I \in \Pi$ (Problembeispiel von Π) erhalten wir, indem wir die Parameter von Π festlegen.

Wdh.: Probleme, Instanzen, Sprachen...

Definition (Ja–/Neininstanzen)

Kodiert man die Instanzen zu einem Entscheidungsproblem Π über einem Alphabet Σ, so enthält $J_\Pi \subseteq \Sigma^*$ die Wörter $I \in \Sigma^*$, deren Lösung in Π „Ja“ ist. I bezeichnen wir dann auch als *Ja–Instanz*.
Die Menge N_Π enthält analog alle *Nein–Instanzen*.

Definition (Sprache)

Die *Sprache* zu einem Entscheidungsproblem Π ist

$$L(\Pi) := \{I \in \Sigma^* \mid I \in J_\Pi\}$$

$\rightsquigarrow$ Eine TM $\mathcal{M}$ *löst* $\Pi \Leftrightarrow L(\mathcal{M}) = L(\Pi)$.

Wdh.: Probleme, Instanzen, Sprachen...

Definition (Ja–/Neininstanzen)

Kodiert man die Instanzen zu einem Entscheidungsproblem Π über einem Alphabet Σ, so enthält $J_\Pi \subseteq \Sigma^*$ die Wörter $I \in \Sigma^*$, deren Lösung in Π „Ja“ ist. I bezeichnen wir dann auch als *Ja–Instanz*.
Die Menge N_Π enthält analog alle *Nein–Instanzen*.

Definition (Sprache)

Die *Sprache* zu einem Entscheidungsproblem Π ist

$$L(\Pi) := \{I \in \Sigma^* \mid I \in J_\Pi\}$$

$\rightsquigarrow$ Eine TM $\mathcal{M}$ *löst* $\Pi \Leftrightarrow L(\mathcal{M}) = L(\Pi)$.

Aufgabe

Problem 3SAT

Sei U eine Menge von aussagenlogischen Variablen und C eine Menge von Klauseln über U, wobei jede Klausel aus C genau die Länge 3 habe. Eine Klausel aus C ist ein Ausdruck der Form:

$$y_1 \vee y_2 \vee y_3 \text{ mit } y_i \in \{u_1, \ldots, u_m\} \cup \{\neg u_1, \ldots, \neg u_m\} \cup \{\texttt{true}, \texttt{false}\}$$

Gesucht: Eine Belegung der Variablen $u_i \in U$ mit `true` oder `false`, so dass alle Klauseln erfüllt werden.

Aufgabe 2

Zeigen Sie: 3SAT $\in \mathcal{NP}$.

Aufgabe

Problem 3SAT

Sei U eine Menge von aussagenlogischen Variablen und C eine Menge von Klauseln über U, wobei jede Klausel aus C genau die Länge 3 habe. Eine Klausel aus C ist ein Ausdruck der Form:

$$y_1 \vee y_2 \vee y_3 \text{ mit } y_i \in \{u_1, \ldots, u_m\} \cup \{\neg u_1, \ldots, \neg u_m\} \cup \{\texttt{true}, \texttt{false}\}$$

Gesucht: Eine Belegung der Variablen $u_i \in U$ mit `true` oder `false`, so dass alle Klauseln erfüllt werden.

Aufgabe 2

Zeigen Sie: 3SAT $\in \mathcal{NP}$.

Aufgabe

Definition

Eine Relation $\rho \subseteq M \times M$ heißt *transitiv*, falls für alle $a, b, c \in M$ gilt:

$$a\rho b \wedge b\rho c \quad \Rightarrow \quad a\rho c$$

Aufgabe 3

Sei $\leq_p$ die Relation „polynomiell reduzierbar“.

Zeigen Sie: $\leq_p$ ist transitiv.

Aufgabe

Definition

Eine Relation $\rho \subseteq M \times M$ heißt *transitiv*, falls für alle $a, b, c \in M$ gilt:

$$a\rho b \wedge b\rho c \quad \Rightarrow \quad a\rho c$$

Aufgabe 3

Sei $\leq_p$ die Relation „polynomiell reduzierbar“.

Zeigen Sie: $\leq_p$ ist transitiv.