Yves Radunz 17.11.2006

Nichtdeterministische Turingmaschinen und NP

1 Nichtdeterministische Algorithmen

1.1 Problembeispiele

Gegeben sind eine Instanz des Problems und eine Vermutung:

1. TSP
   - Instanz: die Städte und die Abstände zwischen ihnen
   - Vermutung: ein Weg, der (vermutlich) nicht länger als $b \in \mathbb{R}_+$ ist
2. Teilgraphenisomorphie
   - Instanz: zwei Graphen G_1 und G_2
   - Vermutung: Teilgraph $G' \subseteq G_1$ und Abbildung $v : G' \to G_2$

$\Rightarrow$ Überprüfung der Korrektheit einer Vermutung ist leicht, d.h. in Polynomialzeit ausführbar

1.2 Arbeitsweise eines Nichtdeterministischen Algorithmus

1. Raten
   - Eingabe: Instanz I eines Problems Π
   - Rückgabe: Vermutung S zu dieser Instanz
2. Überprüfen
   - Eingabe: I und S
   - Rückgabe: Korrektheit der Vermutung = Wird durch S die Instanz I mit „ja“ beantwortet?

Achtung: Die Vermutung muss (wie die Instanz des Problems) sinnvoll kodiert sein.

1.3 Vorläufige Beschreibung von NP

NP ist die Klasse aller Probleme, für die bei gegebenen I und S ein polynomieller Algorithmus zum Überprüfen der Vermutung existiert.

1.4 Vergleich mit P

- Der Algorithmus ist eher eine Beschreibung eines Prüfungsverfahrens für eine Vermutung, als eine Beschreibung der Berechnung.
- Die Symmetrie zwischen einem Problem und dem Komplementärproblem fehlt.
  Beispiel: Existiert bei dem TSP kein Weg, der nicht länger als b ist?

2 Nichtdeterministische Turingmaschinen

2.1 Zweiteilige NDTM

Besteht aus:
- unendlich langes wiederbeschreibbares Band (s. DTM)
- Vermutungseinheit, welche die Vermutung auf das Band schreibt
- Schreibkopf, der die Vermutung schreibt
- endliche Steuereinheit (s. DTM)
- Schreib-/Lesekopf

Programm: $M = (\Gamma, \Sigma, b, Q, \delta, q_0, E)$, wie bei einer DTM

Nachdem die Vermutung auf das Band geschrieben wurde, arbeitet die NDTM wie eine DTM, wird jedoch in ihrem Verhalten ggf. von der Vermutung beeinflusst.

2.2 Einteilige NDTM

Besteht aus:
- unendlich langes wiederbeschreibbares Band (s. DTM)
- endliche Steuereinheit (s. DTM)
- Schreib-/Lesekopf (s. DTM)

Programm: $M = (\Gamma, \Sigma, b, Q, \delta, q_0, E)$, dabei: $\Gamma, \Sigma, b, Q, q_0, E$ wie bei einer DTM
$\delta : Q \times \Gamma \to \mathcal{P}(Q) \times \Gamma \times \{\pm 1\}$
$\Rightarrow$ Jedem Paar $(q, s) \in Q \times \Gamma$ werden ggf. mehrere Folgezustände statt genau einem zugewiesen.

2.3 Definitionen

NDTM akzeptiert ein Wort $x \Leftrightarrow \exists$ Vermutung S : DTM-Bestandteil erkennt Sx
$L_M = \{x \in \Sigma^* | M \text{ akzeptiert } x\}$ = Sprache, die von M erkannt wird
$m_M(x) = \min\{n \in \mathbb{N}_+ | \exists$ Vermutung S :Die Turingmaschine erreicht nach n Schritten den Zustand q_Y.$\}$.
$T_M(n) = \max(\{1\} \cup \{m \in \mathbb{N}_+ | \exists x \in L_M : m_M(x) = m\})$

2.4 Formale Definition von NP

NP : Klasse aller Probleme, die mit nichtdeterministischen Turingmaschinen mit polynomieller Zeitkomplexität gelöst werden können.
Hierfür:
- Gegeben: Instanz I eines Problems Π, Kodierung e
- $x = e(I)$ Wort aus $L[\Pi, e]$, der Sprache, die alle Instanzen des Problems für die NDTM kodiert
- erzeugt alle Vermutungen S (gleichzeitig!)
- DTM erkennt die Wörter Sx in Polynomialzeit, also:
  $\exists$ Polynom $p : \mathbb{N}_+ \to \mathbb{N}_+ : \forall n \in \mathbb{N}_+ : p(n) \geq T_M(n)$

Anmerkung:
Auf meiner Homepage gibt es eine ausführlichere Skript-Version dieses Vortrags:
http://www.mathematik.hu-berlin.de/~radunz/studium/PSPvNP.pdf