Definition NP-Vollständig

Definition 1:

Eine Sprache $L \subseteq \Sigma^*$ heißt NP-Vollständig, falls L

1. NP-hart ist und
2. selbst ein Element von NP.

Alle NP-Vollständigen Probleme bilden die Klasse NPC

NP-Hart

Definition 2:

Eine Sprache L' ist NP-Hart, falls für alle $L \in NP$ gilt:

$$L \leq_p L'$$

Polynomielle Reduktion

Definition 3:

Gegeben zweier Sprachen $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$ ist ein Problem dann polynomiell reduzierbar, falls es eine Funktion $f(x): \Sigma_1^* \rightarrow \Sigma_2^*, \quad x \in L_1$ gibt, welche:

1. in polynomieller Zeit berechnet werden kann und
2. $x \in L_1 \Leftrightarrow f(x) \in L_2$ gilt.

Existiert diese Funktion, so schreibt man: $L_1 \leq_p L_2$

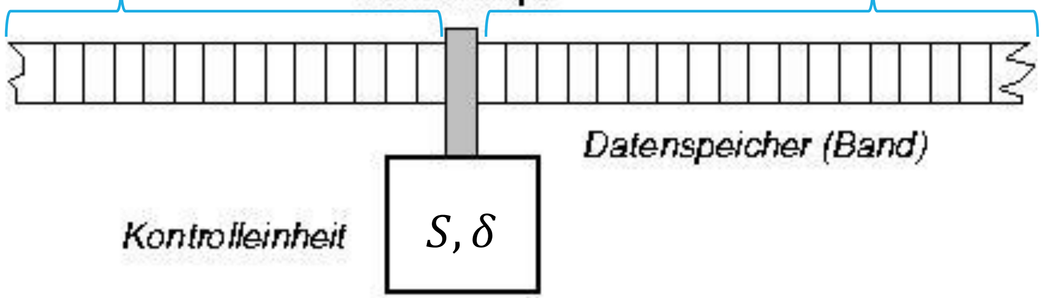
Nichtdeterminismus und NDTM

Ein Nichtdeterministischer polynomieller Algorithmus besteht aus zwei Phasen:

1. Raten
2. Verifizieren

Auf diese Weise als TM darstellbar

-p(n) Leerzeichen: Raum für **rate z** **Eingabe x** und Leerzeichen bis p(n)



SAT

Kurzform von „Satisfiability“ (Erfüllbarkeit)

Auch bekannt als „Erfüllbarkeitsproblem der Aussagenlogik“

Klausel: Disjunktion von Literalen -> $(x_1 \lor y_1)$

Formel: Konjunktion von Klauseln ->$(x_1 \lor y_1) \land (x_2 \lor y_2)$

Problemformulierung:

„Gibt es eine Belegung für alle Literale bei denen die Formel zu WAHR evaluiert wird?“

Cook-Levin Theorem

Satz:

SAT ist NP-Vollständig

Beweis C-L - Variablen

- Bandinhalt B:

  $B_{i,\,k,\sigma} = 1 \rightarrow \text{Nach i Schritten}$ steht Symbol σ in Feld k

- Zustand S:

  $S_{i,\,s} = 1 \;\rightarrow$ Nach i Schritten ist TM in Zustand s

  $$0 \le i \le p(n)$$
  $$-p(n) \le k \le p(n)$$

- Kopfposition P:

  $P_{i,\,k} = 1 \rightarrow$ Nach i Schritten ist Kopf an Position k

Beweis C-L - Startbedingung

Zu Beginn gilt die Startbedingung S:

- TM ist im Startzustand $s_0 \rightarrow S_{0,s_0}$
- Kopf ist an Position 0 -> $P_{0,0}$
- Auf dem Band findet sich nur das Eingabewort ω

- Darstellung:

$$S = S_{0,s_0} \wedge P_{0,0} \left(\bigwedge_{k=-p(n)}^{-1} B_{0,k,\blacksquare}\right) \left(\bigwedge_{k=0}^{n-1} B_{0,k,x_k}\right) \left(\bigwedge_{k=n}^{p(n)} B_{0,k,\blacksquare}\right)$$

Beweis C-L - Randbedingungen

Zusätzlich gelten Randbedingungen $R = R_1 \wedge R_2 \wedge R_3$

- R_1: zu jedem Zeitpunkt i ist die TM in **genau einem** Zustand
- R_2: zu jedem Zeitpunkt i ist der Kopf an **genau einer** Position
- R_3: zu jedem Zeitpunkt i ist **genau ein** Symbol in jedem Feld

$$R_1 = \bigwedge_{i=0}^{p(n)} one\ (S_{i,s_0}, S_{i,s_1}, \dots)$$

$$R_2 = \bigwedge_{i=0}^{p(n)} one\ (P_{i,-p(n)}, \dots, P_{i,p(n)})$$

$$R_3 = \bigwedge_{i=0}^{p(n)} \bigwedge_{k=-p(n)}^{p(n)} One\ (B_{i,k,\sigma_1}, B_{i,k,\sigma_2}, \dots)$$

One $(y_1,\dots,y_m) \Leftrightarrow \sum_{i=1}^{n} y_i = 1$

Beweis C-L - Transition

Es gilt $T = (T_1 \land T_2 \land T_3)$

T_1 sichert ab, dass die TM beim Schritt von i nach $i+1$ in eine Folgekonfiguration übergeht

T_2 sorgt dafür, dass eine terminierte TM nicht weiter arbeitet

T_3 bildet die Tatsache ab, dass nur der Kopf den Bandinhalt ändern kann

$$T_2 = \bigwedge_{\substack{i,k,s,\sigma \\ \delta(s,\sigma)=\emptyset}} (S_{i,s} \land P_{i,k} \land B_{i,k,\sigma}) \rightarrow (S_{i+1,s} \land P_{i+1,k} \land B_{i+1,k,\sigma})$$

$$T_3 = \bigwedge_{i=0}^{p(n)} \bigwedge_{k=-p(n)}^{p(n)} \bigwedge_{\sigma\in\Pi} (\overline{P_{i,k}} \land B_{i,k,\sigma}) \rightarrow B_{i+1,k,\sigma}$$

Beweis Cook-Levin - Bedingungen

$$T_1 = \bigwedge_{\substack{i,k,s,\sigma \\ \delta(s,\sigma)\neq\emptyset}} (S_{i,s} \wedge P_{i,k} \wedge B_{i,k,\sigma}) \rightarrow \left[\bigvee_{\substack{(s',\sigma',\leftarrow) \\ \in\delta(s,\sigma)}} (S_{i+1,s'} \wedge P_{i+1,k-1} \wedge B_{i+1,k,\sigma'}) \right.$$

$$\left. \bigvee_{\substack{(s',\sigma',\rightarrow) \\ \in\delta(s,\sigma)}} (S_{i+1,s'} \wedge P_{i+1,k+1} \wedge B_{i+1,k,\sigma'}) \bigvee_{\substack{(s',\sigma',\circlearrowleft) \\ \in\delta(s,\sigma)}} (S_{i+1,s'} \wedge P_{i+1,k} \wedge B_{i+1,k,\sigma'}) \right]$$

Beweis Cook-Levin - Bedingungen

Zusätzlich gilt die Akzeptanzbedingung:

$$A = \bigvee_{z \in E} S_{p(n),z}$$

Beweis Cook-Levin - Klauselmenge

Klauselmenge bleibt polynomiell:

- $|S| = \mathcal{O}(p(n))$
- $|R| = \mathcal{O}(p(n)^3)$
- $|T| = \mathcal{O}(p(n)^2)$
- $|A| = \mathcal{O}(1)$

Fazit

- NP-Vollständigkeit eines Problems P bedeutet:
  - P ist NP-Hart und
  - P liegt in NP

- Ein Problem liegt dann in NP, wenn ein ND-Algorithmus existiert der das Problem löst
  - ND-Polynomielle Algorithmen bestehen aus Guess & Check Phasen

- NP-Härte wird dadurch gezeigt, dass sich alle Probleme in NP auf ein bestimmtes Problem polynomiell reduzieren lassen

Quellen nach Folie

D.W. Hoffmann - Theoretische Informatik, Kapitel 7

M.R. Garey, D.S. Johnson – Computers and Intractability, Kapitel 2