Wieviel Platz brauchen Algorithmen wirklich?

Vergleich von Problemen mittels Reduktionen

Dmitriy Shorin

August 7, 2010

Abstract

Vergleichen ist oft leichter als messen. Es mag Ihnen schwer fallen, die Größe eines Menschen auf einen Blick auf den Zentimeter genau anzugeben – Sie werden aber kein Problem haben, für zwei nebeneinander stehende Menschen den größeren zu bestimmen; auch wenn einer nur einen Zentimeter größer ist. Wenn Sie nicht ein absolutes Gehör haben, werden Sie kaum eine Tonhöhe direkt bestimmen können – Sie werden aber kein Problem haben, wenn Sie eine Stimmgabel zur Hand haben.

In der Theorie ist auch leichter, die Schwierigkeit von Problemen zu vergleichen als sie direkt zu bestimmen. Für solche "Schwierigkeitsvergleiche" benutzt man Reduktionen.

Contents

1 Einführung

Reduktion ist einer der grundlegenden Begriffe der Komplexitätstheorie.

2 Reduktion

Definition. "Das Problem A ist auf das Problem B **reduzierbar**" bedeutet, dass man aus einem beliebigen wirksamen Algorithmus für das Problem A den wirksamen Algorithmus für das Problem B machen kann. Reduktion ist tatsächlich eine Umwandlung eines Problems ins anderes.

Wichtige Bemerkungen:
Was muss eine Reduktion von A auf B alles leisten?

- Die Reduktion löst das Problem B nicht.

- Die Reduktion löst das Problem A auch nicht.

- Es reicht nicht, wenn "jede Lösung für A" (also jedes $x \in A$) in "eine Lösung für B" (also ein $y \in B$) umgewandelt wird. Vielmehr muss auch im Falle von $x \notin A$ gelten $y \notin B$ [Tan09].

3 Logspace-Many-One-Reduktion

Definition. Eine **Logspace-Turing-Maschine mit Ausgabe** ist eine Deterministische Turing-Maschine mit

- einem Read-Only-Eingabeband,

- Arbeitsbänder, auf denen der Platzverbrauch maximal $O(\log n)$ ist bei Eingaben der Länge n, und

- einem Write-Only-Ausgabeband [Tan09].

Definition. Eine Funktion $f : \Sigma* \to \Sigma*$ heißt **logspace-berechenbar**, wenn eine Logspace-Turing-Maschine M mit Ausgabe existiert, die

- bei jeder Eingabe x irgendwann anhält und

- dann auf dem Ausgabeband gerade $f(x)$ steht [Tan09].

Lemma. Sei f logspace-berechenbar via einer Maschine M. Dann benötigt f auch nur polynomielle Zeit.
(Genauer: Für jedes $x \in \Sigma*$ endet M bei Eingabe x nach $O(n^k)$ Schritten für ein festes k.) [Tan09]

Definition. Seien A und B Sprachen. Man sagt, "A ist auf B **logspace-many-one-reduzierbar** ($A \leq_m^{\log} B$), falls es eine logspace-berechenbare Funktion f gibt, so dass

für alle $x \in \Sigma*$ gilt $x \in A \Leftrightarrow f(x) \in B$ [Tan09].

4 Transivität

"Die Logspace-Many-One-Reduktion ist **transitiv**" bedeutet, dass man A auf B und B auf C reduzieren kann, so auch A auf C.

5 Schwere

Definition. Seien

- C eine Klasse von Sprachen und

- X ein Problem (nicht unbedingt in C).

Dann heißt X **schwer für** C **unter** $\leq_m^{\log}$**-Reduktionen**, falls

- für alle Probleme $A \in C$ gilt $A \leq_m^{\log} X$.

Bemerkung: Kann man ein für C schweres Problem effizient lösen, dann kann man alle Probleme in C effizient lösen [Tan09].

6 Vollständigkeit

Definition. Seien

- C eine Klasse von Sprachen und

- X ein Problem.

Dann heißt X **vollständig für** C **unter** $\leq_m^{\log}$**-Reduktionen**, falls

- $X \in C$

- X ist schwer für C.

Bemerkung: Vollständige Probleme sind die schwierigsten Probleme einer Klasse [Tan09].

References

[Tan09] T. Tantau. Theoretische Informatik. *Vorlesungsskript*, 2009.