marc-oliver pahl

Informatik II – Kapitel 10

„Theorie der Algorithmenkonstruktion“

Zusammenfassung des Kapitel 10
Küchlin, Weber, Einführung in die Informatik, 2.Auflage

10.6.2004

Problem

- Algorithmus
  - **Spezifikation**
    - Ein-/ Ausgabeparameter, **Vor-/ Nachbedingung**…
  - **Durchführbarkeit**
    - Mechanisch („Programm läuft“)
  - **Korrektheit**
    - Induktion/ Floyd (in Kap. 10 noch mal durchformalisiert)

o**Partielle Korrektheit**:
Ein- und Ausgabeparameter genügen **im Falle einer Terminierung** immer der **Spezifikation**.
o**Totale Korrektheit**:
Partiell korrekt und **terminiert** für jede Eingabe innerhalb der Eingabespezifikation.

Implementierung

Schrittfolge(möglichkeit) einer Methode:

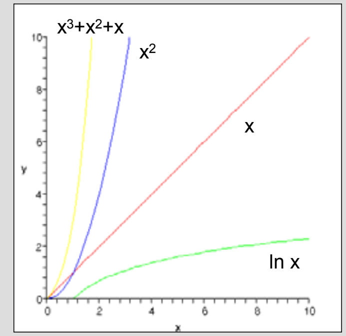
- Initialisierung
  - Initialisiere Hilfsgrößen
- Ausnahmefall
  - Behandle Eingaben, die nicht der Spezifikation entspr.
- Trivialfall
  - Behandle Fälle (oft 0 oder null), für die Erg. Trivial
- Einfacher Fall
  - Fälle bei denen Problemlsg. Einfach (oft Indukt.basis)
- Problemreduktion
  - Behandle schweren Fall
    -> Reduktion:
    - **Rekursion**
    - Iteration

- Pseudo-Code
  - Vermischung von natürlichsprachlichen, mathematischen und programmiersprachlichen Elementen:
    - „r=a“, „a=min{x | x ϵ a[7..14]}“
- Rekursionsformeln
  - „ABS(a) ≡ if a > 0 then a else –a fi“

- Bottom-up
  - o Aus vorhandenen „kleinen“ Elementen etwas größeres Konstruieren
    - Vektoraddition ausgehend von der Addition natürlicher Zahlen
    - „Was haben wir schon, wie können wir daraus die Lösung bauen?“
- Top-down
  - o Das gesamte Problem in Teile zerlegen
    - Matrizenmultiplikation
    - „Nehmen wir an, wir könnten schon…, dann ginge es so und so. Jetzt müssen wir nur noch … implementieren.“

- Greedy (gierig) $\boxed{x^y = x * x^{y-1}}$
  - o Trivialfälle
  - o Reduktion in eine Richtung
  - o Rekursion
- Divide-and-Conquer $\boxed{x^y = x^{y/2} * x^{y/2}}$
  - o Trivialfälle
  - o Teile
  - o Herrsche auf deinem Teil
  - o Kombiniere die fertigen Teile zur Gesamtlösung
  - o Stichwort „dynamisches Programmierung“
    - o Betrachtung mehrerer möglicher Teilungen simultan

Der Graph Laufzeit/ Eingabe interessiert uns…
- o Logarithmisch
- o Linear
- o Quadratisch
- o Polynomial
- o Exponentiell
- o …



Konstante Faktoren (graphisch: Verschiebung der Kurve) sind dabei unerheblich.

In der Regel wird betrachtet…
- o Zeitkomplexität
- o Platzkomplexität

Sinnvolle Sichtwinkel sind…
- o Worst case
- o Average case
- o Best case

Notation:

- Obere Schranke
  - $O(g(n)) = \{f(n) \mid \exists\, c, n_0>0 \ \forall\, n>n_0\ 0 <= f(n) <= c*g(n)\}$
- Exakte Ordnung
  - $\theta(g(n)) = \{f(n) \mid \exists\, c_1, c_2, n_0>0 \ \forall\, n>n_0\ 0 <= c_1*g(n) <= f(n) <= c_2*g(n)\}$
- Untere Schranke
  - $\Omega(g(n)) = \{f(n) \mid \exists\, c, n_0>0 \ \forall\, n>n_0\ 0 <= c*g(n) <= f(n)\}$

Zum Rechnen auf „Komplexitäten“ sind die Sätze und Lemmata aus Buch (2. Aufl. S.316f) und Vorlesung (l‘Hopital) relevant!

Deren (zumeist leichte) Beweise führen evtl. zu weiterführender Einsicht… also nachvollziehen…