Lehr- & Lernziel

- Ermittlung des Zeitaufwands bei der Auswertung von Operationen*) auf Zeichenketten

Hilfsmittel : CE-S

- Algorithmenmodellierung durch bedingte Gleichungen mit Auswertungsschritten und Beweistechnik

---

*) synonym für Algorithmen

Syntaxschema

Name - spec

Schlüsselwörter:

ops: ..., $f: D_1 \times \cdots \times D_k \to D$, ..., $c: \to D'$, ...

vars: ..., $x \in D''$, ...

eqns: ..., $L = R$ [falls b], ...

L, R: Terme

b: Boolescher Term

f, c, x Namen

$D_1, \ldots, D_k, D, D', D''$ Typen $(A, A^*, \mathrm{BOOL}, \mathbb{N}, \mathbb{Z})$

[...] optional

Auswertungsschritt (Gleichungsanwendung)

(1) Zuweisung aktueller Werte $a(x) \in T_D$ an die Variablen $x \in D$ und Substitution in Gleichung $L = R$:

$$L[a] \rightarrow R[a]$$

(2) dasselbe, aber in Argumentterm:

$$op(t_1, \ldots, t_k) \rightarrow op(t_1, \ldots, t_{i-1}, t_i', t_{i+1}, \ldots, t_k)$$

falls $t_i \rightarrow t_i'$

Berechnung und Gleichwertigkeit

- Berechnung als Schrittfolge

  $t = t_0 \to t_1 \to \cdots \to t_n = t'$ $(n=0 : t=t')$

  dafür kurz: $t \overset{*}{\to} t'$

- Gleichwertigkeit wie Berechnung mit Rückwärtsschritt $(R=L$ statt $L=R)$

  $t \leftrightarrow t'$ falls $t \to t'$ oder $t \leftarrow t'$

  $\overset{*}{\leftrightarrow}$ analog $\overset{*}{\to}$

- für $t \overset{*}{\leftrightarrow} t'$ auch $t = t'$

  beachte: $\to \subseteq \overset{*}{\to} \subseteq \overset{*}{\leftrightarrow} \supseteq \leftrightarrow$

Auswertungsschritt (Gleichungsanwendung)

(1) Zuweisung aktueller Werte $a(x) \in T_D$ an die Variablen $x \in D$ und Substitution in Gleichung $L = R$ falls b:

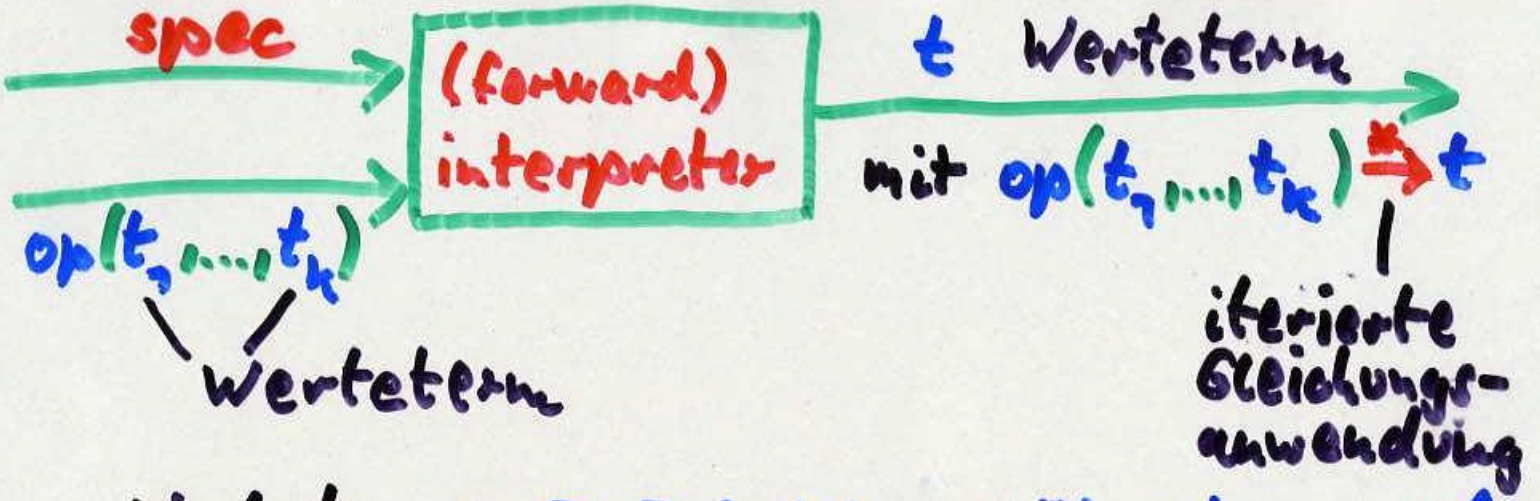
$$L[a] \rightarrow R[a] \quad \text{falls } b(a) = T$$

(2) dasselbe, aber in Argumentterm:

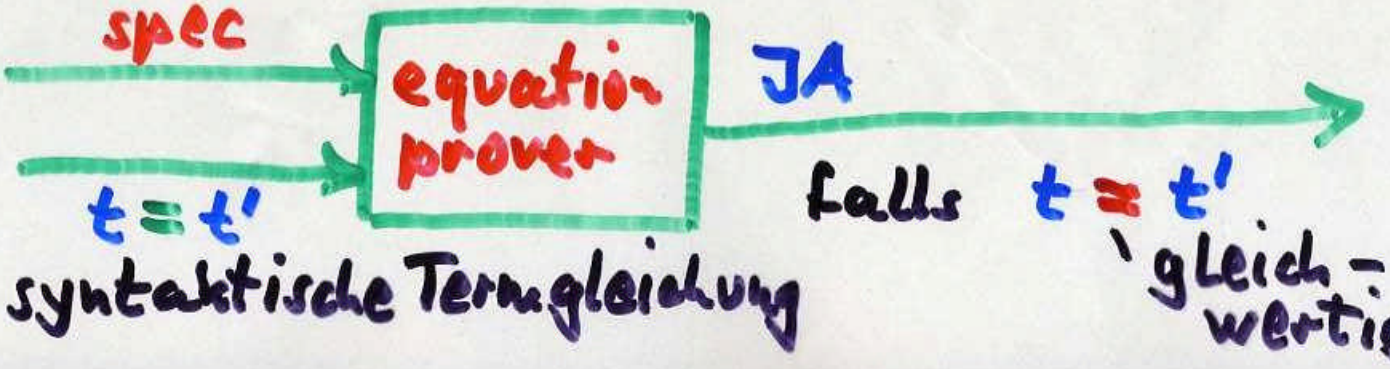
$$op(t_1, \ldots, t_k) \rightarrow op(t_1, \ldots, t_{i-1}, t_i', t_{i+1}, \ldots, t_k)$$

falls $t_i \rightarrow t_i'$

(Vorwärts-) Interpreter & Theorembeweiser



- Wertetermε: $T, F; 0, 1, 2, \ldots \in \mathbb{N}; a, b, c, \ldots \in A;$
  $l, a_1 \ldots a_n \in A^*$ für $a_i \in A, n \geq 1$



Sortiertheitstest

is-sorted

opns: $\text{is-sorted} : A^* \to \text{BOOL}$

vars: $x, y \in A$, $u \in A^*$

eqns: $\text{is-sorted}(\lambda) = T$

$\text{is-sorted}(x) = T$

$\text{is-sorted}(xyu) = (x \equiv y) \wedge \text{is-sorted}(yu)$

Beobachtungen: $T^{\text{is-sorted}}(0) = T^{\text{is-sorted}}(1) = 1$ und

$T^{\text{is-sorted}}(n+2) = 1 + T^{\text{is-sorted}}(n+1)$ für $n \geq 0$

Theorem: $T^{\text{is-sorted}}(k) = k$ für alle $k \geq 1$

Beweis (mit vollständiger Induktion über k):

I.A.: $k = 1$: $T^{\text{is-sorted}}(1) = 1$ (nach Beob.)

I.V.: Beh. gelte für ein $k \geq 1$

I.S.: $k+1$: $T^{\text{is-sorted}}(k+1) \underset{n=k-1}{=} T^{\text{is-sorted}}(n+2) \underset{\text{Beob.}}{=} 1 + T^{\text{is-sorted}}(n+1)$

$\underset{\text{I.V.}}{=} 1 + n + 1 \underset{n=k-1}{=} k + 1$

Aufwand als Zahl der Gleichungsanwendungen

$T^{op}(n)$ = Zahl der Gleichungsanwendungen beim Auswerten von $op(t_1, \dots, t_k)$ mit $Length(t_i) = n$ im schlechtesten Fall

(Berechnungen, die nicht Wörter betreffen, zählen nicht)

- „Time" als Hinweis auf Zeitaufwand
- Operation $op: D_1 \times \cdots \times D_k \to D$ mit $D_i = A^*$
- Länge des Eingabewortes

Sortieren durch Einsortieren

sort

opns: $sort: A^* \to A^*$, $insort: A \times A^* \to A^*$

vars: $x, y \in A$, $u \in A^*$

eqns: $sort(\lambda) = \lambda$

$sort(x\,u) = insort(x, sort(u))$

$insort(x, \lambda) = x$

$insort(x, y\,u) = \text{if } x \leq y \text{ then } x\,y\,u \text{ else } y\,insort(x, u)$

Beobachtungen: $T^{sort}(0) = 1$ und

$T^{sort}(n+1) = 1 + T^{sort}(n) + T^{insort}(m)$

$= n+2 + T^{sort}(n)$

m: Länge von $sort(u)$

Voraussetzungen:

(1) $Length(sort(u)) = Length(u)$

(2) $T^{insort}(n) = n+1$

Theorem: $T^{sort}(n) = \frac{(n+2)(n+1)}{2}$ für alle $n \geq 0$

Beweis (mit vollständiger Induktion über n):

I.A.: $n=0$: $T^{sort}(0) = 1 = \frac{(0+2)(0+1)}{2}$ (nach Beob.)

I.V.: Beh. gelte für ein $n \geq 0$

I.S.: $n+1$: $T^{sort}(n+1) \underset{\text{Beob.}}{=} n+2 + T^{sort}(n)$

$$\underset{\text{I.V.}}{=} n+2 + \frac{(n+2)(n+1)}{2}$$

$$= \left(1 + \frac{n+1}{2}\right)(n+2)$$

$$= \frac{(n+3)(n+2)}{2}$$

wesentliche Elemente zur Ermittlung des Aufwands von Algorithmen

- Modellierung von Algorithmen
- einschließlich ihrer Berechnung (Aufführung)
- quantitative Erfassung als Zahl der Berechnungsschritte
- Nachweisbarkeit dafür erforderlicher Eigenschaften
- geeigneter Ansatz: CE-S