Theoretische Informatik 2

Sabine Kuske
Linzer Str. 9a, OAS 3005
Tel.: 2335, 8794
kuske@informatik.uni-bremen.de
www.informatik.uni-bremen.de/theorie

7. April 2008

Beispiel für eine Spezifikation in CE-S

insort

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

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

eqns: $insort(x, \lambda) = x$

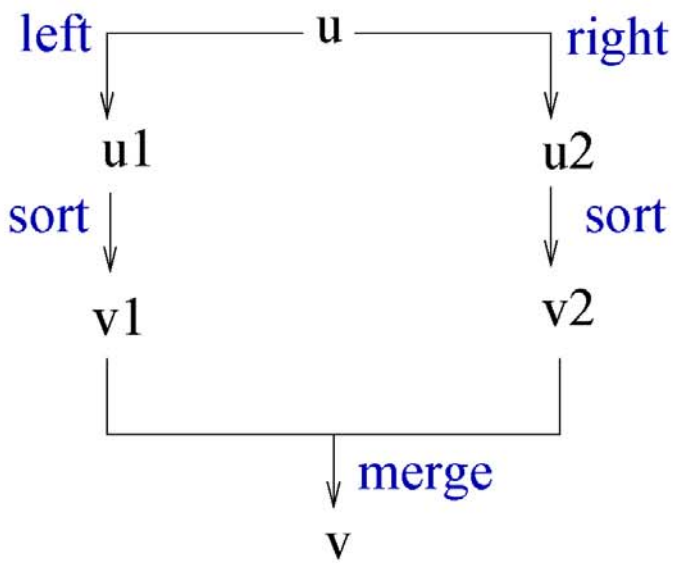
$insort(x, yv) = \text{if } x \leq y \text{ then } xyv \text{ else } y\, insort(x, v)$

$sort(\lambda) = \lambda$

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

Beispiel: Sortieren durch Mischen

Idee: Sortiere linke und rechte Hälfte eines Wortes und mische die sortierten Hälften zusammen.



CE-S-Ansatz zur Aufwandsermittlung

Zeitaufwand als maximale Zahl von Gleichungsanwendungen, die nötig sind, um eine Operation für bestimmte Argumente auszuwerten in Abhängigkeit von der Länge der eingegebenen Zeichenketten und der Größe der eingegebenen Zahlen.

Schreibweise: $T^{op}(n)$ bzw. $T^{op}(n_1, \ldots, n_k)$

k: Anzahl der Zeichenkettenargumente und der Zahlenargumente von op

Vorgehensweise (1)

- Auswirkung der spezifischen Gleichungen auf die Aufwandsfunktion

Beispiel

$$\boxed{insort(x, \lambda) = x} \quad \rightsquigarrow \quad \boxed{T^{insort}(0) = 1}$$

$$\boxed{insort(x, yv) = \left.\begin{array}{l} \text{if } x \leq y \text{ then } xyv \\ \text{else } y\,insort(x, v) \end{array}\right\} \quad \rightsquigarrow}$$

$$\boxed{\begin{aligned} T^{insort}(n+1) &=^{?} 1 + \left\{\begin{array}{l} 0 \text{ falls } x \leq y \\ T^{insort}(n) \text{ sonst} \end{array}\right. \\ &=_{Schlechtester\ Fall} 1 + T^{insort}(n) \end{aligned}}$$

Vorgehensweise (2)

- Aufstellen einer nicht rekursiven Aufwandshypothese (Beweis meist durch vollständige Induktion)

  Zum Beispiel:

$$T^{insort}(n) = n + 1 \text{ für alle } n \in \mathbb{N}$$

- Statt einer exakten Aufwandsaussage kann auch nach oben abgeschätzt werden (was oft einfacher ist).

Vorgehensweise (3)

- Beim Beweis von Aufwandsabschätzungen werden oft Eigenschaften der Operationen benötigt (Beweis mit Hilfe von Gleichwertigkeit oft mit vollständiger Induktion über Wörter).

  Zum Beispiel:

$$T^{sort}(n) = \frac{(n+2)(n+1)}{2}$$

  falls $T^{insort}(n) = n + 1$ und $length(sort(u)) = length(u)$.

mergesort

opns: $left, right, sort : A^* \to A^*,\ merge : A^* \times A^* \to A^*$

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

eqns: $left(\lambda) = \lambda\ ,\ left(x) = x\ ,\ left(xvy) = x\, left(v)$

$right(\lambda) = \lambda\ ,\ right(x) = \lambda\ ,\ right(xvy) = right(v)y$

$merge(\lambda, v) = v\ ,\ merge(u, \lambda) = u$

$merge(xu, yv) = \text{ if } x \leq y \text{ then } x\, merge(u, yv)$
$\text{else } y\, merge(xu, v)$

$sort(\lambda) = \lambda,\ sort(x) = x$

$sort(w) = \ merge(sort(left(w)), sort(right(w)))$
$\text{falls } length(w) \geq 2$