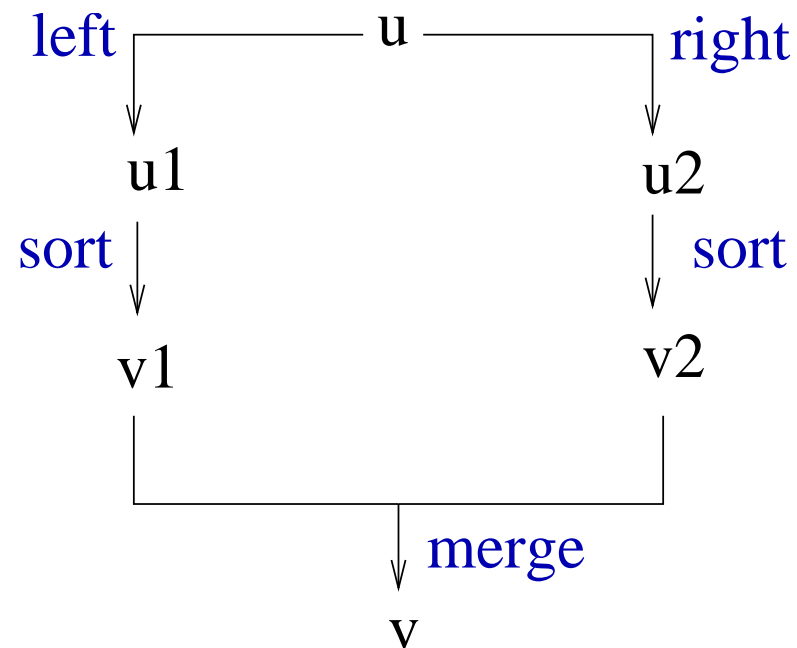


Beispiel: Sortieren durch Mischen

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



mergesort

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

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

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

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

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

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

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

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

Aufwand von sort

Benötigte Eigenschaften:

$$1. T^{left}(n) = T^{right}(n) = \begin{cases} \frac{n}{2} + 1 & \text{falls } n \text{ gerade,} \\ \frac{n+1}{2} & \text{sonst.} \end{cases}$$

$$2. T^{merge}(m, n) = \begin{cases} 1 & \text{falls } \min(m, n) = 0, \\ m + n & \text{sonst.} \end{cases}$$

$$3. \text{length}(\text{sort}(w)) = \text{length}(w).$$

$$4. \text{length}(\textit{left}(w)) = \begin{cases} \frac{\text{length}(w)}{2} & \text{falls } \text{length}(w) \text{ gerade,} \\ \frac{\text{length}(w)+1}{2} & \text{sonst.} \end{cases}$$

$$5. \text{length}(\textit{right}(w)) = \begin{cases} \frac{\text{length}(w)}{2} & \text{falls } \text{length}(w) \text{ gerade,} \\ \frac{\text{length}(w)-1}{2} & \text{sonst.} \end{cases}$$

$$6. \text{length}(\textit{left}(w)) + \text{length}(\textit{right}(w)) = \text{length}(w)$$

Aufwand von sort

1. Auswirkungen der Gleichungen auf den Aufwand:

$$T^{sort}(0) = T^{sort}(1) = 1$$

Für $n \geq 2$:

$$T^{sort}(n) = 2 + 2n + \begin{cases} 2 * T^{sort}\left(\frac{n}{2}\right) + 1 & \text{falls } n \text{ gerade,} \\ T^{sort}\left(\frac{n+1}{2}\right) + T^{sort}\left(\frac{n-1}{2}\right) & \text{sonst.} \end{cases}$$

2. Abschätzung nach oben:

$$T^{sort}(n) \leq 5n * k \text{ für } k \geq 1 \text{ und } 2^{k-1} < n \leq 2^k.$$