DMat-10

Rekursivt definerede mængder.

For at definere en mængde S rekursivt skal vi
Basisskridt: angive et eller flere elementer, der tilhører S
Rekursionsskridt: angive en eller flere regler, der hver ud fra et eller flere elementer i S konstruerer et nyt element i S.

Strukturel induktion.

Lad S være en rekursivt defineret mængde. Og lad $P(x)$ være et udsagn for et ethvert $x \in S$.

For at bevise at $P(x)$ er sand for alle $x \in S$ skal vi:
Basisskridt: bevise at $P(x)$ er sand for ethvert x indført i basisskridtet af definitionen af S
Rekursionsskridt: bevise at hvis x er konstrueret fra $x_1, \ldots, x_\ell$ i rekursionsskridtet af definitionen af S og hvis $P(x_1), \ldots, P(x_\ell)$ er sande så er $P(x)$ sand.

Σ : et alfabet, altså en endelig mængde af symboler.

Definition. Σ^*, mængden af strenge over Σ defineres ved:

Basisskridt: Den tomme streng $\lambda \in \Sigma^*$.

Rekursionsskridt: Hvis $w \in \Sigma^*$ og $x \in \Sigma$ så er $wx \in \Sigma^*$.

Definition. Konkatenering af strenge $w_1 \cdot w_2$ af to strenge $w_1, w_2 \in \Sigma^*$ defineres ved:

Basisskridt: $w_1 \cdot \lambda = w_1$

Rekursionsskridt: Hvis $w_2 \in \Sigma^*$ så er

$$w_1 \cdot (w_2 x) = (w_1 \cdot w_2) x.$$

Definition. Mængden af fulde binære træer kan defineres ved:

Basisskridt: Et træ, der består ét punkt r, er et fuldt binært træ med rod r.

Rekursionsskridt: Hvis T_1 og T_2 er fulde binære træer så er $T_1 \cdot T_2$ et fuldt binært træ, der består af T_1, T_2 og en rod r samt en kant fra r til roden af T_1 og en kant fra r til roden af T_2.

Definition. Mængden af udvidede binære træer kan defineres ved:

Basisskridt: $\emptyset$ er et udvidet binært træ.

Rekursionsskridt: Hvis T_1 og T_2 er udvidede binære træer så er $T_1 \cdot T_2$ et udvidet binært træ, der består af T_1, T_2 og en rod r samt en kant fra r til roden af T_i, hvis $T_i \neq \emptyset$, for $i = 1, 2$.

$n(T)$: antal punkter i et fuldt binært træ T opfylder:

Basisskridt: Hvis T består af en rod så er $n(T) = 1$.

Rekursionsskridt: Hvis $T = T_1 \cdot T_2$ så er

$$n(T) = n(T_1) + n(T_2) + 1.$$

$h(T)$: højden af et fuldt binært træ T defines ved

Basisskridt: Hvis T består af en rod så er $h(T) = 0$.

Rekursionsskridt: Hvis $T = T_1 \cdot T_2$ så er

$$h(T) = \max\{h(T_1), h(T_2)\} + 1.$$

procedure fibonacci (n: ikke-negativt heltal)
if $n = 0$ **then** fibonacci(0) := 0
else
    if $n = 1$ **then** fibonacci(1) := 1
    else fibonacci(n) := fibonacci($n-1$)+fibonacci($n-2$)

Ved rekursiv beregning af f_n beregnes f_1 i alt f_n gange.

procedure mergesort($L = a_1, \ldots, a_n$ liste af tal)
if $n > 1$ **then**
begin

$m := \lfloor \frac{n}{2} \rfloor$

$L_1 := a_1, \ldots, a_m$

$L_2 := a_{m+1}, \ldots, a_n$

$L :=$ merge(mergesort(L_1), mergesort(L_2))

end

{ L er sorteret i ikke-aftagende rækkefølge}

Sætning 1. Antal sammenligninger, der i alt bruges af Mergesort for at sortere en liste med $n = 2^k$ elementer med er højst

$$2^k(k-1)+1 = n(\log(n)-1)+1.$$

Mergesort har kompleksitet $O(n\log(n))$.