4.3: Rekursivt definerede mængder.

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 (udvidede) binære træer kan defineres ved:

Basisskridt: $\emptyset$ er et binært træ.

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

4.4: Rekursive algoritmer.

Algoritme 2: Rekursiv beregning af a^n Side 312

procedure power (a reelt tal $\neq$ 0, n: ikke-negativt heltal)
if $n = 0$ **then** power$(a, n) = 1$
else power$(a, n) = a \cdot$ power$(a, n-1)$

Algoritme: Iterativ beregning af a^n

procedure iterativ power (a reelt tal $\neq$ 0, n: ikke-negativt heltal)
$p := 1$
for $i := 1$ **to** n
    $p := a \cdot p$
$\{ \ p = a^n \ \}$

procedure fibonacci (n: ikke-negativt heltal)
if $n = 0$ **then** $y := 0$
else
begin
    $x := 0$
    $y := 1$
    for $i := 1$ **to** $n - 1$
    begin
        $z := x + y$
        $x := y$
        $y := z$
    end
end
$\{\ y = f_n\ \}$

```
procedure fibonacci (n: ikke-negativt heltal)
if n = 0 then   y := 0
else
begin
    x := 0
    y := 1
    i := 1
    while i < n
    { invariant: x = f_{i-1}, y = f_i }
    begin
        z := x + y
        x := y
        y := z
        i := i + 1
    end
end
{ y = f_n }
```