

3. Zentralübung

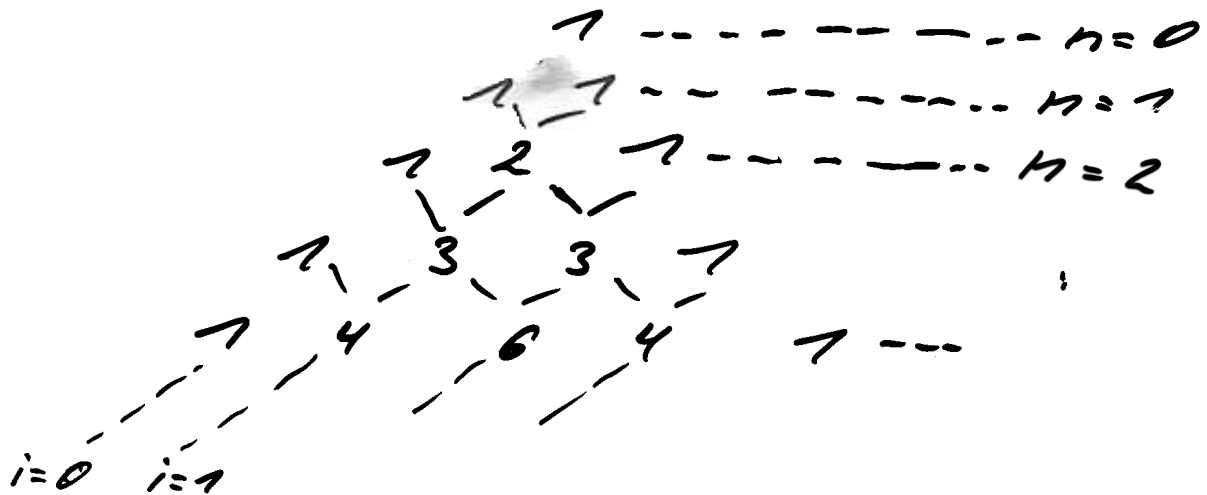
6.11.2002

3

3.1 Einbettung der Binomialkoeffizienten

Def: $Bi(n, i) : (a+b)^n = \sum_{i=0}^n Bi(n, i) a^i b^{n-i}$
mit $Bi(n, i) = \frac{n!}{i!(n-i)!}$

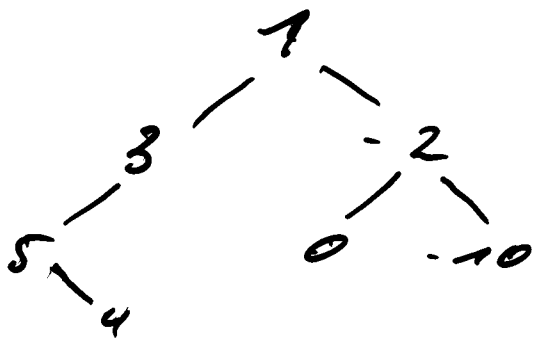
Die Binomialkoeff. lassen sich mit Hilfe des Pascalschen Dreiecks berechnen



$$\Rightarrow Bi(n, i) = \begin{cases} 0, & i > n \\ 1, & i = 0 \text{ || } i = n \\ Bi(n-1, i-1) + Bi(n-1, i) \end{cases}$$

Lösungsidee

- Berechnung der n -ten Zeile des Δ und Auswahl des i -ten Elements
- Berechnung der n -ten Zeile:
 - rekursiv aus der Zeile $n-1$
 - die vorhergehende Zeile wird als Parameter übergeben



Problem: Ein fügen eines Elements
in den sort. Binärbaum.

let rec insert (e, b) = match (e, b) with

| (e, Leer) → Wurzel (Leer, e, Leer)

| (e, b) when (e >= root b) && left b = Leer
→ Wurzel (Wurzel (Leer, e, Leer), root b,
right b)

when (e >= root b)

→ Wurzel (insert (e, left b), root b,
right b)

⋮

Typdefinition in Ocaml

```
type baum = Leer | Wurzel of  
           (baum * int * baum)
```

Beispiel

einelementiger Baum:

② let L1 = Wurzel (Leer, 2, Leer);;

zweielementiger Baum:



[]

```
let L2 = Wurzel (Wurzel (Leer, 5, Leer),  
                2, Leer);;
```

Bemerkung: Die durch L1 und L2
definierten Terme sind Konstruktor-
terme (bestehen im wesentl. aus Konstr.);
Pattern matching ist nur mit Konstruktor-
termen möglich.

Beispiel: Funktion zur Rückgabe der
Wurzel eines Baumes (s.o.):

```
let root b = match b with  
  | Leer → fail with ("undefined")  
  | Wurzel (l, w, r) → w ;;
```

Implementierung:

let $bi(m, i) = m^{th} \text{ (zeile } m) \text{ } i ; ;$

6

let $zeile\ m = h([1], 0, \overset{\uparrow}{m}) ; ;$
Zielzeile

let rec $h(lz, j, n) = \text{match } (lz, j, n) \text{ with}$
| (lz, j, n) when $j = n \rightarrow lz$
| $(lz, j, n) \rightarrow h(1 :: (\text{nextline } lz), j+1, n) ; ;$

let rec $\text{nextline } lz = \text{match } lz \text{ with}$
| $[x] \rightarrow [1]$
| $x :: y :: ll \rightarrow (x+y) :: \text{nextline } (y :: ll) ; ;$

Bemerkung: Die eingetragte Variante hat maximal n^2 Aufrufe (vgl. 2^n bei nichtlinearer Variante)

$B: (30, 15)$

3.2 Bäume in Ocaml

13.11.02

Bäume ganzer Zahlen, speziell sortierte Binärbäume

