



Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



# Baumstrukturen



Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Definitionen (1)

**Ein Baum (tree) ist entweder leer oder er besteht aus genau einer Wurzel (root) mit einer beliebigen Anzahl von Teilbäumen (subtrees). Jeder Teilbaum ist wieder ein Baum.**

**Die Wurzel des Baumes und die Wurzeln seiner Teilbäume heissen auch Knoten (nodes) des Baumes.**

**Knoten ohne Teilbäume sind die Blätter (leaves) des Baumes.**



Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Definitionen (2)

**Benachbarte Knoten eines Baumes sind durch Kanten (edges) verbunden.**

**Eine Folge von Knoten die durch Kanten miteinander verbunden sind heisst Pfad (path).**

**Zwischen zwei beliebigen Knoten eines Baumes gibt es genau einen Pfad.**



Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Geordneter Baum

**In einem geordneten Baum (ordered tree) ist die Reihenfolge der Teilbäume jedes Knotens relevant.**

**Bäume, die sich nur durch die Reihenfolge ihrer Teilbäume unterscheiden sind zueinander isomorph.**



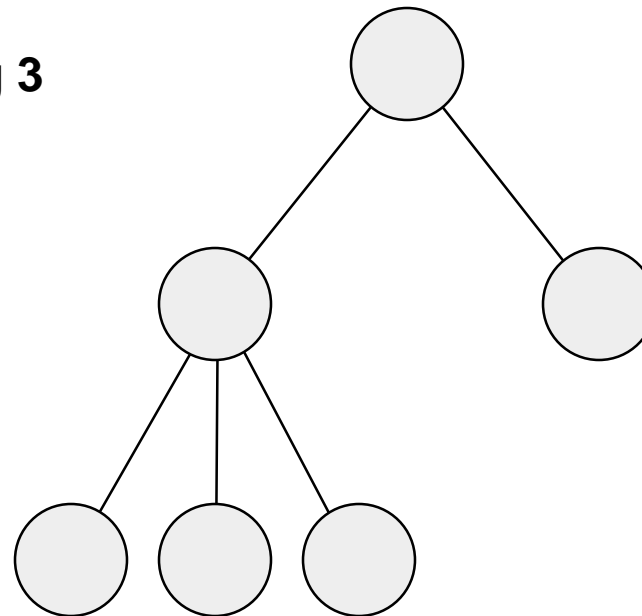
Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Ordnung eines Baumes

**Die maximale Anzahl von Teilbäumen eines Knotens entspricht der Ordnung eines Baumes.**

**Beispiel:  
Baum der Ordnung 3**





Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Binärer Baum

**Ein binärer Baum (binary tree) ist ein Baum in dem jeder Knoten maximal zwei Teilbäume hat.**

**Binäre Bäume haben die Ordnung 2.**

**Ist der binäre Baum geordnet, so sprechen wir von linken und rechten Teilbäumen.**



Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Perfekter Binärer Baum

**Ein binärer Baum heisst perfekt oder voll wenn der linke und rechte Teilbaum jedes Knotens aus gleich vielen Knoten besteht.**

**In einem perfekten binären Baum ist jeder Knoten entweder ein Blatt oder er hat genau zwei nicht leere Teilbäume.**



Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Höhe eines Baumes

**Die Höhe (height) eines Baumes ist die maximale Pfadlänge von der Wurzel zu den Blättern des Baumes.**

**Für einen perfekten binären Baum mit  $n$  Knoten und der Höhe  $h$  gilt**

$$n = 2^{h+1} - 1$$

**und**

$$h = \text{ld}(n+1) - 1$$

**Ein perfekter binärer Baum der Höhe  $h$  hat genau  $2^h$  Blätter.**

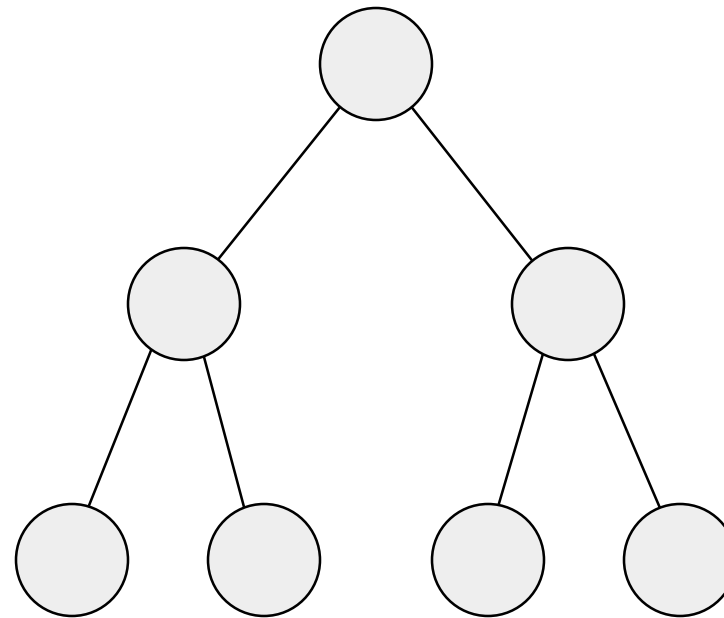




Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Beispiel: Perfekter Binärer Baum



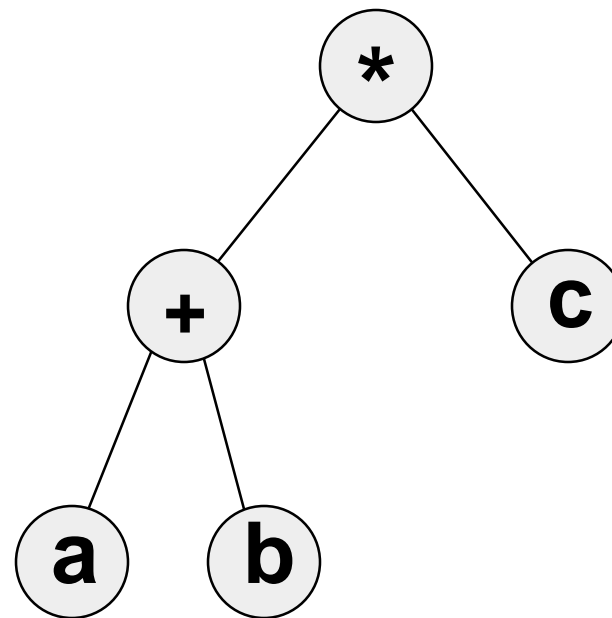
$n = 7$   
 $h = 2$



Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Beispiel: Arithmetischer Ausdruck $(a+b)*c$



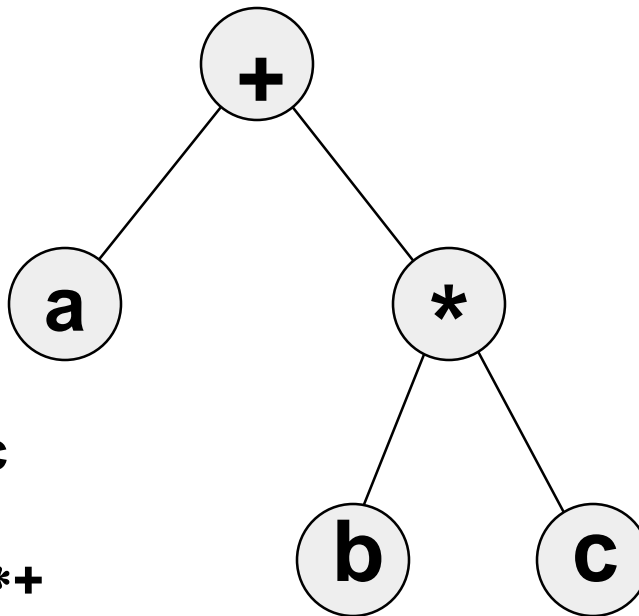
preorder:  $*+abc$   
inorder:  $(a+b)*c$   
postorder:  $ab+c*$



Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Beispiel: Arithmetischer Ausdruck $a+b*c$



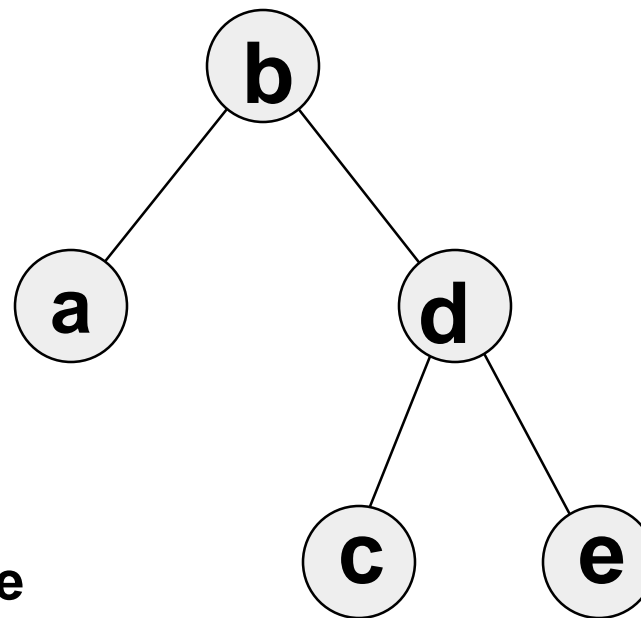
preorder:  $+a*bc$   
inorder:  $a+b*c$   
postorder:  $abc*+$



Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Binärer Sortierbaum



**inorder: abcde**  
**preorder: badce**  
**postorder: acedb**  
**levelorder: badce**



Helmut Schauer  
Educational Engineering Lab  
Department for Information Technology  
University of Zurich



## Binärer Sortierbaum

**Ein binärer Sortierbaum ist ein geordneter binärer attributierter Baum für den beim inorder-Traversieren alle Attribute gemäss ihrer Ordnungsrelation durchlaufen werden.**

**Bezeichnet  $x.key$  das Attribut eines Knotens  $x$  und  $x.left$  den linken und  $x.right$  den rechten Teilbaum des Knotens  $x$  so gilt bei steigender Sortiertheit eines Sortierbaumes  $T$ :**

**All  $x: x \in T$ :**

**$(\text{All } y: y \in x.left: y.key \leq x.key) \wedge (\text{All } y: y \in x.right: y.key \geq x.key)$**