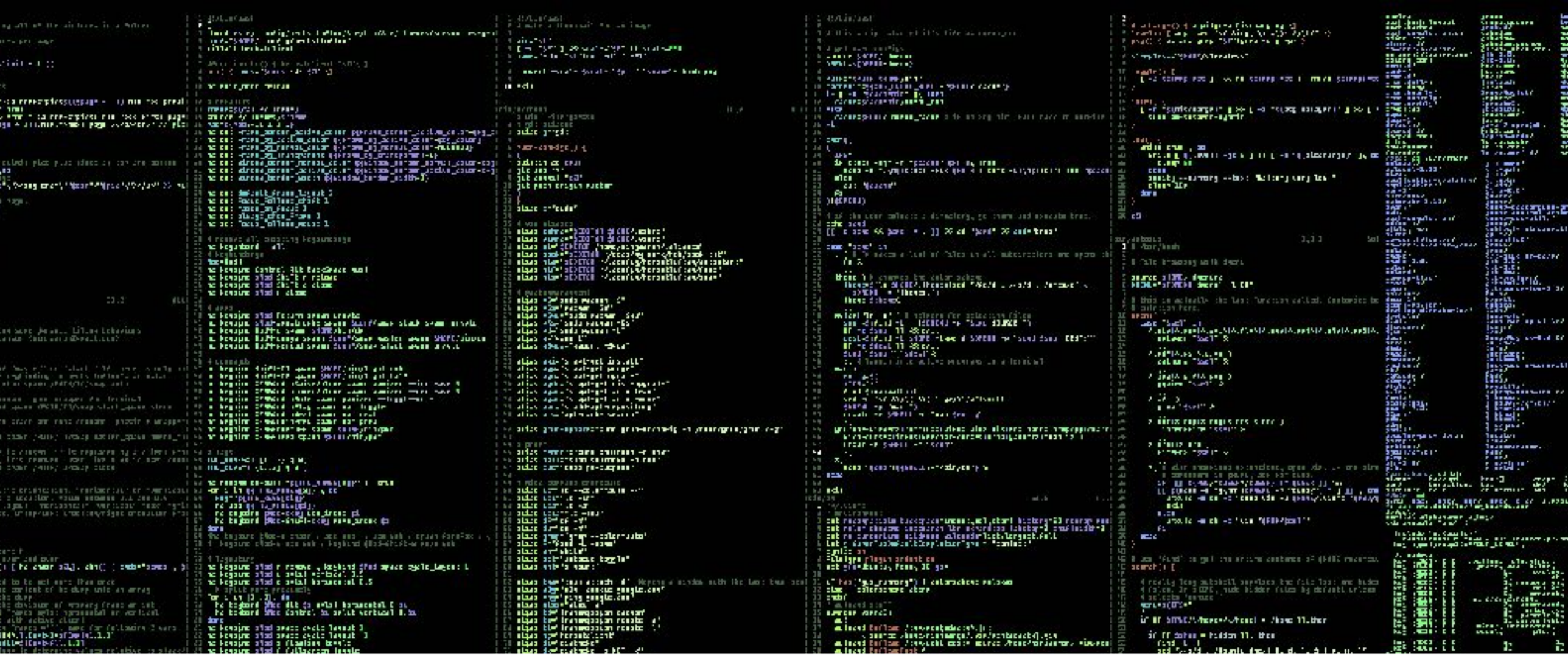
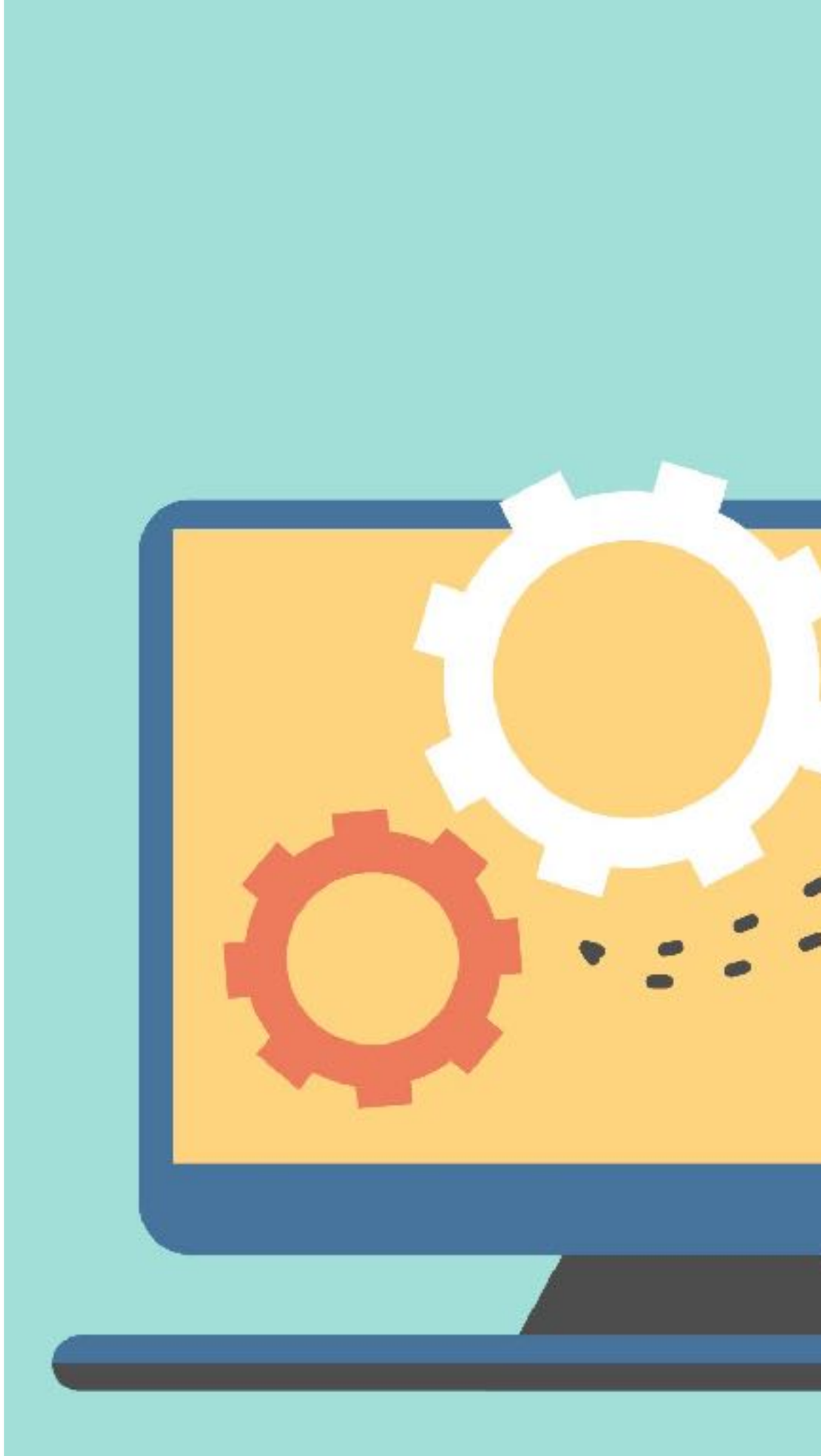


INFORMATIK I

Tutorium 24.01.2017

BESPRECHUNG

Blatt 11



WIEDERHOLUNG

Vorlesung & Für Blatt 12

WIEDERHOLUNG: BINÄRBÄUME

- Definition Baum: Ein Baum ist ein spezieller Graph, der zusammenhängend ist und keine geschlossenen Pfade enthält
- Ein Baum besteht aus einer Wurzel, mehreren Knoten und mehreren Blättern
- Jeder Knoten hat einen Vater (Parent) und (möglicherweise) mehrere Kinder, hat ein Knoten keine Kinder (oder nur leere in unserem Fall) ist er ein Blatt
- Spezialform Binärbaum: Jeder Knoten hat maximal zwei Kinder

WIEDERHOLUNG: BINÄRBÄUME

- Wofür sind Bäume gut?
- Effiziente Datenstruktur, beispielsweise Suchbäume, AVL-Bäume, a-b-Bäume, etc.
- KI's verwenden beispielsweise Bäume um den Suchraum effizienter zu durchsuchen
- Binärbäume können zum Beispiel als einfacher Suchbaum dienen, linkes Kind ist immer kleiner als Vater, rechtes Kind immer größer gleich

WIEDERHOLUNG: BINÄRBÄUME

- Binärbäume in Scheme:
- Induktive Definition
- empty-tree ist in der Menge der Binärbäume T(M)

$$\forall x \in M \text{und l,r} \in T(M) : (make - node\ l\ x\ r) \in T(M)$$

- Jeder Knoten (make-node) in einem Binärbaum hat zwei Teilbäume (Kinder) l und r sowie eine Markierung (Label) x

WIEDERHOLUNG: BINÄRBÄUME

- Definitionen für Binärbäume:

```
; Ein Knoten (node) besitzt
; - einen linken Zweig (left-branch),
; - eine Markierung (label) und
; - einen rechten Zweig (right-branch)
(: make-node (%a %b %c -> (node-of %a %b %c)))
(: node-left-branch  ((node-of %a %b %c) -> %a))
(: node-label        ((node-of %a %b %c) -> %b))
(: node-right-branch ((node-of %a %b %c) -> %c))

(define-record-procedures-parametric node node-of
  make-node
  node?
  (node-left-branch
   node-label
   node-right-branch))
```

WIEDERHOLUNG: BINÄRBÄUME

- Definitionen für Binärbäume:

```
; Der leere Baum (the-empty-tree) besitzt
; keine weiteren Eigenschaften
(: make-empty-tree (-> the-empty-tree))

(define-record-procedures the-empty-tree
  make-empty-tree
  empty-tree?
  ())

; Der leere Baum
(: empty-tree the-empty-tree)
(define empty-tree (make-empty-tree))
```

WIEDERHOLUNG: BINÄRBÄUME

- Definitionen für Binärbäume:

```
; Signatur für Binärbäume (btree-of t) mit Markierungen der Signatur t
; (im linken/rechten Zweig jedes Knotens findet sich jeweils wieder
; ein Binärbaum)
(define btree-of
  (lambda (t)
    (signature (mixed the-empty-tree
                      (node-of (btree-of t) t (btree-of t)))))))
;                               \_________/   \_________/
;                                    ↑             ↑
;                                zweifache Rekursion, s. (list-of t)
```

WIEDERHOLUNG: BINÄRBÄUME

- Definitionen für Binärbäume:

```
; Konstruiere ein Blatt mit Markierung x
(: make-leaf (%a -> (btree-of %a)))
(define make-leaf
  (lambda (x)
    (make-node empty-tree x empty-tree)))
```

WIEDERHOLUNG: BINÄRBÄUME

- Definition Tiefe eines Baumes: Die maximale Länge eines Weges von der Wurzel zu einem leeren Teilbaum wird Tiefe eines Baumes genannt

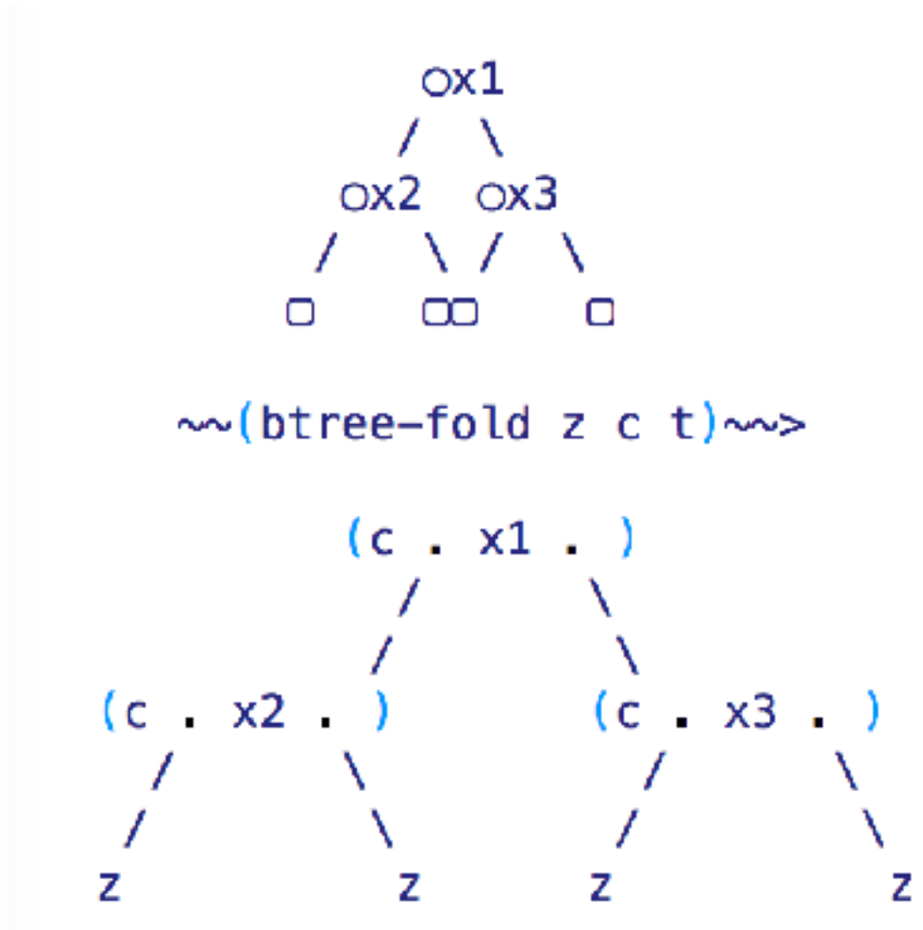
```
; Tiefe des Binärbaumes t
(: btree-depth ((btree-of %a) -> natural))

(check-expect (btree-depth empty-tree) 0)
(check-expect (btree-depth t1) 3)
(check-expect (btree-depth t2) 2)
(check-expect (btree-depth classifier) 4)

(define btree-depth
  (lambda (t)
    (cond ((empty-tree? t)
           0)
          ((node? t)
           (+ 1
              (max (btree-depth (node-left-branch t))
                   (btree-depth (node-right-branch t))))))))
```

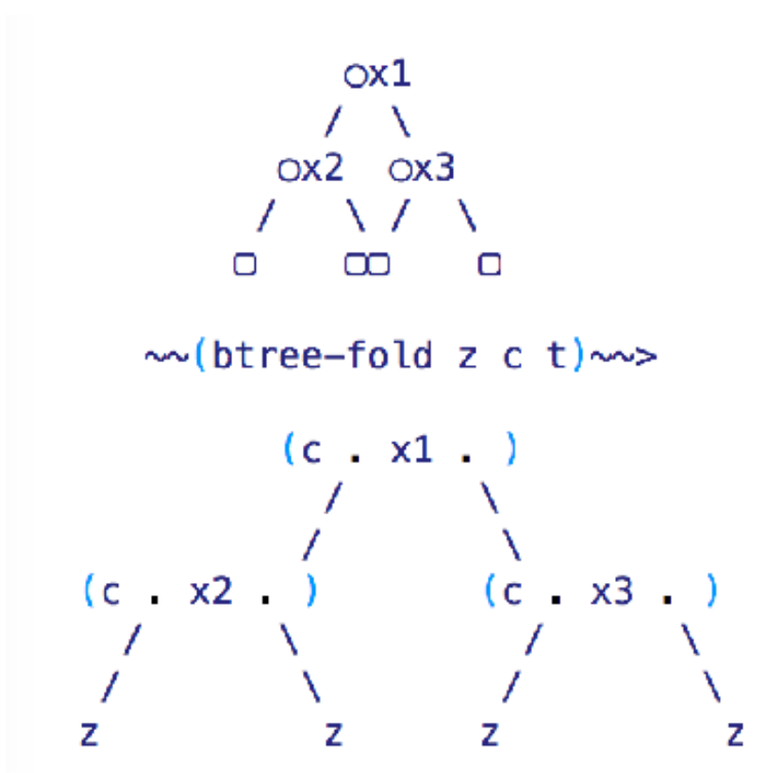
WIEDERHOLUNG: BINÄRBÄUME

- Wir haben bereits fold für Listen kennengelernt
- Fold lässt sich auch für Bäume definieren



WIEDERHOLUNG: BINÄRBÄUME

```
; Falte Baum t bzgl. z und c
(define btree-fold
  (lambda (z c t)
    (cond ((empty-tree? t)
           z)
          ((node? t)
           (c (btree-fold z c (node-left-branch t))
              (node-label t)
              (btree-fold z c (node-right-branch t)))))))
```



Steffen Lindner, Email: steffen.lindner@student.uni-tuebingen.de

WIEDERHOLUNG: BINÄRBÄUME

- Man kann Bäume auf verschiedene Arten durchlaufen
- Tiefensuche bzw. Breitensuche
- Es gibt verschiedene Arten einen Baum zu einer Liste umzuwandeln, inorder / preorder / postorder
- Dabei kommt es auf die Reihenfolder an, in der linker Teilbaum / Label / rechter Teilbaum eingefügt werden

BEISPIEL: BINÄRE SUCHE

- Alternative Implementierung:
- Nutze nicht Eigenschaft der binären Suchbäume sondern nutze btree-fold:

```
(: inTree2? (natural (btree-of natural) -> boolean))
(check-expect (inTree2? 10 t1) #f)
(check-expect (inTree2? 3 t1) #t)
(define inTree2?
  (lambda (n t)
    (btree-fold
     #f
     (lambda (l x r)
       (or l (= x n) r))
     t)))
```