Haskell Arrays

Funktionale Arrays

Christian Höner zu Siederdissen
christian.hoener.zu.siederdissen@uni-jena.de

Theoretische Bioinformatik, Bioinformatik Uni Jena

Januar, 2024

Haskell Arrays: Index

Eine Alternative zu den Listen für Rush-hour

```
1  -- Zugriff auf Arrays
2  import Data.Array.IArray as A
3
4  --Index-Typen in Arrays
5  class Ix a where
6    range :: (a, a) -> [a]
7    index :: (a, a) -> a -> Int
8    inRange :: (a, a) -> a -> Bool
9
10 ix = ( (1,1), (6,6) )    -- Spielplan Dimension
11 rn = range ix             -- aktive Zellen
12 at = index ix (2,3)       -- direkter Index
13 ok = inRange ix (7,5)     -- Zelle ist ausserhalb
```

Haskell Arrays

```
1  -- Arrays haben diesen Constraint
2  type IArray :: (* -> * -> *) -> * -> Constraint
3
4  -- Gegeben Array 'a', mit Index Typ i, WertTyp e
5  -- Beziehe Wert and Position i
6  (!) :: (IArray a e, Ix i) => a i e -> i -> e
7
8  -- Array-Konstruktion
9  array :: (IArray a e, Ix i) => (i, i) -> [(i, e)] -> a i e
10
11 -- Neues Array, gegeben altes Array und Update
12 (//) :: (IArray a e, Ix i) => a i e -> [(i, e)] -> a i e
```

Beispiel

```
 1  ix = ((1,1),(2,3))
 2  test :: A.Array Dim2 Int
 3  test = array ix [ (k,0) | k <- range ix ]
 4       // [ ((2,3),9) ]
 5
 6
 7
 8  % array ((1,1),(2,3))
 9  %   [ ((1,1),0), ((1,2),0), ((1,3),0)
10  %   , ((2,1),0), ((2,2),0), ((2,3),9) ]
```