## A Sudoku Solver - Specifications (1A)

- Richard Bird Implementation

[^0]Please send corrections (or suggestions) to youngwlim@hotmail.com.
This document was produced by using OpenOffice.

## Based on

Thinking Functionally with Haskell, R. Bird
https://wiki.haskell.org/Sudoku
http://cdsoft.fr/haskell/sudoku.html
https://gist.github.com/wvandyk/3638996
http://www.cse.chalmers.se/edu/year/2015/course/TDA555/lab3.html

## Basic Data Types

```
type Choices = [Digit]
type Matrix a = [Row a]
type Row a = [a]
type Grid = Matrix Digit
type Digit = Char
digits :: [Digit]
digits = ['1'..'9']
```

```
blank :: Digit -> Bool
blank = (== '0')
```

[ [a]]

The valid digits are '1' to '9' A list of non-zero characters ('1'to ‘9')
[ [Digit]] 9x9 matrix of digits
'0' standing for blank

## Matrix Digit \& Matrix Choices



## Sudoku

|  |  | 4 |  |  | 5 | 7 |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  |  |  |  |  | 9 | 4 |  |  |
| 3 | 6 |  |  |  |  |  |  | 8 |
| 7 | 2 |  |  | 6 |  |  |  |  |
|  |  |  | 4 |  | 2 |  |  |  |
|  |  |  |  | 8 |  |  | 9 | 3 |
| 4 |  |  |  |  |  |  | 5 | 6 |
|  |  | 5 | 3 |  |  |  |  |  |
|  |  | 6 | 1 |  |  | 9 |  |  |

> [ [ '0’, '0’, '4’, ‘0’, '0', ‘5’, '7’, ‘0’, ‘0’ ],
> [ '0’, ‘0’, ‘0’, ‘0’, ‘0’, ‘9’, ‘4', ‘0’, ‘0’ ],
> [ '3', ‘6’, '0’, '0’, '0’, ‘0’, '0', '0’, '8’ ],

$$
\begin{aligned}
& \text { [ '0’, ‘0’, ‘0’, '0’, ‘8’, ‘0’, '0’, '9', '3' ], }
\end{aligned}
$$

> [ '0’, ‘0’, ‘5’, ‘3’, ‘0’, ‘0’, ‘0’, ‘0’, ‘0’ ],
> [ '0’, ‘0’, ‘6’, ‘1’, ‘0’, ‘0’, ‘9’, ‘0’, ‘0’ ] ]
type Grid $=$ Matrix Digit
[Row Digit]
[[Digit]]

## Specification (0)

| solve1 | : | Grid | -> [Grid] |
| :---: | :---: | :---: | :---: |
| choices | : | Grid | -> Matrix Choices |
| expand | : | Matrix Choices | -> [Grid] |
| cp | : | [[ a ]] | -> [[ a ]] |
| valid | : | Grid | -> Bool |
| nodups | :: Eq a => | [ a ] | -> Bool |
| rows | :: | Matrix a | -> [Row a] |
| cols | : | Matrix a | -> [Row a] |
| boxs | : | Matrix a | -> [Row a] |
| ungroup group [] | $\begin{aligned} & =\text { concat } \\ & =[] \end{aligned}$ |  |  |
| group (x:y | :xs) $=[x, y, z]$ | group xs |  |

## Function Types : choices, expand

| solve1 | : |  | -> [Grid] |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Matrix Digit | [Matrix Digit] |  |  |
| choices | : | Grid | -> Matrix |  |  |
|  |  | Matrix Digit | Matrix [Digit] |  |  |
| expand | : | Matrix Choices | -> [Grid] |  |  |
|  |  | Matrix [Digit] | [Matrix Digit] |  |  |
|  |  | type Digit = | Char |  |  |
|  |  | type Choices = | [Digit] |  |  |
|  |  | type Row a = | [a] |  |  |
|  |  | type Matrix a = | [Row a] |  |  |
|  |  | Matrix Digit | [Row Digit] | [[Digit]] |  |
|  |  | Matrix Choices | [Row Choices] | [[Choices]] | [[[Digit]]] |
|  |  | type Grid = | Matrix Digit | [Row Digit] | [[Digit]] |

## Function completions

```
solve :: Grid -> [Grid]
solve = filter valid . completions
```

completions :: Grid -> [Grid]
valid :: Grid -> Bool
completions $=$ expand. choices
[Grid]
[Matrix Digit]
expand
Matrix Choices [Digit]
Matrix $\quad \begin{gathered}\text { choices } \\ \text { Matrix Digit }\end{gathered}$

## Function: choices

```
choices :: Grid -> Matrix Choices
choices = map (map choice)
    where choice d | blank d = digits
    | otherwise = [d]
```

Installs the available digits for each cell
If the cell is blank, then all digits for possible choices
else there is only one choice and a singleton is returned

```
digits :: [Digit]
```

digits :: [Digit]
digits = ['1'..'9']
digits = ['1'..'9']
blank :: Digit -> Bool
blank :: Digit -> Bool
blank = (== '0')

```
blank = (== '0')
```

```
choices :: Grid -> Matrix [Digit]
```

choices :: Grid -> Matrix [Digit]
choices $=$ map (map choice)
choices $=$ map (map choice)
choice $d=$ if blank $d$ then digits else [d]

```
choice \(d=\) if blank \(d\) then digits else [d]
```

else there is only one choice and a singleton is returned

## Function: choices

## choices $=$ map $($ map choice $)$

## map choice

| [ [ 0 ', '0', '4', '0', '0', '5', '7', '0', '0, ], |  |
| :---: | :---: |
| © | [ '0, '0', '0', '0', '0', '9', '4', '0', '0’ ], |
|  | '3', '6', '0', '0', '0', '0, |
| ก | 2, ${ }^{\prime}$ |
| E | [ '0', '0', '0', ‘ 4 |
| $\begin{aligned} & \text { O} \\ & \stackrel{\text { ® }}{\underline{E}} \end{aligned}$ | , |
|  | [ '4', '0', '0', '0', '0', '0', '0', '5', '6' |
|  | [ '0, '0', ‘5', '3', '0’, '0', '0', '0, '0' |
|  | [ $0,0,6$ |

## Matrix Choices Example

$$
\text { Matrix Choices }=[\text { Row Choices }] \Rightarrow[[\text { Choices }]] \Rightarrow[[\text { [Digit }]]
$$

$$
\begin{aligned}
& \text { [ [ ['1'..'9'], ['1'..'9'], ['4'], ['1'..'9'], ['1'...'9'], ['5'], ['7'], ['1'..'9'], ['1'...'9'] ], } \\
& \text { [ ['1'..'9'], ['1'...'9'], ['1'..'9'], ['1'...9'], ['1'..'9'], ['9'], ['4'], ['1'...'9'], ['1'..'9'] ], } \\
& \text { [ ['3'], ['6'], ['1'..'9'], ['1'..'9'], ['1'..'9'], ['1'...'9'], ['1'..'9'], ['1'...'9'], ['8'] ], } \\
& \text { [ ['7'], ['2'], ['4'], ['1'..'9'], ['6'], ['1'...'9'], ['1'..'9'], ['1'...'9'], ['1 ...'9'] ], } \\
& \text { [ ['1'..'9'], ['1'...'9'], [ '1'..'9'], ['4'], ['1'..'9'], ['2'], ['1'..'9'], ['1'...'9'], ['1'..'9'] ], } \\
& \text { [ ['1'..'9'], ['1'...'9'], ['1'..'9'], ['1'...'9'], ['8'], ['1'..'9'], ['1'..'9'], ['9'], ['3'] ], } \\
& \text { [ ['4'], ['1'..9'], ['1'..'9'], ['1'..'9'], ['1'..'9'], ['1'..'9'], ['1'..'9'], ['5'], ['6'] ], } \\
& \text { [ ['1'..'9'], ['1'..'9'], ['5'], ['3'], ['1'..'9'], [ [1'..'9'], ['1'..'9'], ['1'..'9'], ['1'..'9'] ], } \\
& \text { [ ['1'..'9'], ['1'..'9'], ['6'], ['1'], ['1'..'9'], ['1'..'9'], ['9'], ['1'..'9'], ['1'..'9'] ]] }
\end{aligned}
$$

## Cartesian Product (cp)

```
cp [[1, 2, 3], [2], [1, 3]]
[[1, 2, 3] x [2] x [1, 3]]
[[1, 2, 1], [1, 2, 3], [2, 2, 1], [2, 2, 3], [3, 2, 1], [3, 2, 3]]
cp [] = [[]]
cp [[1], [2], [3]] => [[1, 2, 3]]
cp [[1], [], [4, 5]] => []
```


## Cartesian Product (cp)

$$
\begin{aligned}
& \mathbf{c p}[[1,2,3],[2],[1,3]] \\
& {[[1,2,1],[1,2,3],[2,2,1],[2,2,3],[3,2,1],[3,2,3]]} \\
& \mathbf{c p}[[2],[1,3]]=[[2,1],[2,3]] \\
& \mathbf{c p}([1,2,3]:[[2],[1,3]])=[[1,2,1],[1,2,3], \\
& {\left[\begin{array}{r}
{[2,2,1],[2,2,3],} \\
[3,2,1],[3,2,3]]
\end{array}\right.}
\end{aligned}
$$

$[1,2,3] \times \mathbf{c p}[[2],[1,3]]=[1,2,3] \times[[2,1],[2,3]]$
list comprehension
cp (xs:xss) $=[\mathrm{x}: \mathrm{ys} \mid \mathrm{x}<-\mathrm{xs}, \mathrm{ys}<-\mathbf{c p} \mathrm{xss}]$

## Cartesian Product (cp)

```
cp (xs:xss) = [x:ys | x <- xs, ys <- cp xss]
```

$$
\begin{array}{r}
\mathbf{c p}(x s: x s s)=[x: y s \mid x<-x s, y s<-y s s] \\
\text { where yss }=\mathbf{c p} \text { xss }
\end{array}
$$

```
cp [xs] = cp (x:[ ])
    \(=[\mathrm{x}: \mathrm{ys} \mid \mathrm{x}<-\mathrm{xs}, \mathrm{ys}<-\mathbf{c p}[]]\) if \(\mathbf{c p}[]=[]\)
    \(=[\mathrm{x}: \mathrm{ys} \mid \mathrm{x}<-\mathrm{xs}, \mathrm{ys}<-\) [ ] ]
    = [] contradict
```

$\mathbf{c p}[]=[]$ results in $\mathbf{c p}$ xss = [] therefore $\mathbf{c p}[]=[[]]$

## Expand

```
expand :: Matrix Choices -> [Grid]
expand = cp . map cp
```

$\mathbf{c p} \cdot \operatorname{map} \mathbf{c p}=[[[a]]]$-> [ [[a]] ]

$$
\begin{aligned}
& \text { digits :: [Digit] } \\
& \text { digits }=[11 ' . .9 \text { ' }] \\
& \text { blank }:: \text { Digit -> Bool } \\
& \text { blank }=(==\text { '0') }
\end{aligned}
$$

[Grid]
[Matrix Digit]
expand Matrix Choices

## Expand Operations

expand :: Matrix Choices -> [Grid]
[ [[a]] ] -> [ [[a]] ]
expand $=\mathbf{c p} \cdot \operatorname{map} \mathbf{c p}$
map $\mathbf{~ c p ~}$


## Matrix Choices Example

```
expand :: Matrix Choices -> [Grid]
expand = cp . map cp
```

```
[ cp [[1'...9'], [1'...'9'], [4'], ['1'...'9], [1'1...'9], ['5'], ['7'], ['1'..'9'], ['1'...'9']],
cp [ ['1'..'9'], ['1'...'9'], ['1'...'9'], ['1'..'9'], ['1'..'9'], ['9'], ['4'], ['1'..'''], ['1'..'9']],
    cр [ ['3'], ['6'], ['1'...'9], ['1'..'9'], ['1'..'9'], [1'..'9'], [1'..'9], ['1'..'9'], ['8'] ],
    cр [['7'], ['2'], ['4'], ['1'..'9'], ['6'], ['1'..'9'], ['1'...9'], ['1'...9'], ['1'..'9']],
    cp [['1'..'9'], ['1'..'9'], [1'...'9'], ['4'], ['1'..'9'], ['2'], ['1'...'9'], ['1'..'9'], ['1'..'9']],
    cр [ ['1'..'9'], ['1'..'9'], [1'...'9'], ['1'..'9'], ['8'], ['1'..'9], ['1'..'9], ['9], ['3'] ],
    ср [['4'], ['1'...'9], ['1'..'9'], ['1'..'9'], ['1'..'9'], [1'...'9], ['1'..'9'], ['5'], ['6'] ],
    ср [['1'..'9], ['1'..'9'], ['5'], ['3], ['1'..9'], [1'...9'], [1'...'9], ['1'..'9'], ['1'..'9]],
    ср [['1'...'9], ['1'..'9'], ['6'], ['1'], ['1'..'9'], [1'..'9'], ['9'], [['1'..''], ['1'..'9']]]
```

```
cp [[1, 2, 3], [2], [1, 3]]
[[1, 2, 1], [1, 2, 3], [2, 2, 1], [2, 2, 3], [3, 2, 1], [3, 2, 3]]
```


## Matrix Choices Example

```
expand :: Matrix Choices -> [Grid]
expand = cp . map cp
```

| row1 choices |  |  | ], |
| :---: | :---: | :---: | :---: |
| row2 choices |  | - •• | ], |
| row3 choices |  | - - - | ], |
| row4 choices |  |  |  |
| row5 choices |  | - •• | ], |
| row6 choices |  | - - - | ], |
| row7 choices |  | - - - | ], |
| row8 choices |  |  | ], |
| row9 choices |  | - •• |  |

```
cp [[1, 2, 3], [2], [1, 3]]
[[1, 2, 1], [1, 2, 3], [2, 2, 1], [2, 2, 3], [3, 2, 1], [3, 2, 3]]
```


## [Grid]



## Expand

```
> solve1 :: Grid -> [Grid]
> solve1 = filter valid . expand . choices
> type Choices = [Digit]
> choices :: Grid -> Matrix Choices
> choices = map (map choice)
\(>\) where choice d|blank d = digits
> | otherwise = [d]
> expand :: Matrix Choices -> [Grid]
> expand \(=\mathbf{c p} . \operatorname{map} \mathbf{c p}\)
> cp :: [[a]] -> [[a]]
\(>\mathbf{c p}[]=[[]]\)
\(>\mathbf{c p}(x s: x s s)=[x: y s \mid x<-x s, y s<-c p x s s]\)
```

```
digits :: [Digit]
```

digits :: [Digit]
digits = ['1'..'9']
digits = ['1'..'9']
blank :: Digit -> Bool
blank :: Digit -> Bool
blank = (== '0')

```
blank = (== '0')
```


## Specification (1)

```
solve1 :: Grid -> [Grid]
solve1 = filter valid . expand . choices
type Choices = [Digit]
choices :: Grid -> Matrix Choices
choices = map (map choice)
    where choice d | blank d = digits
    | otherwise = [d]
expand :: Matrix Choices -> [Grid]
expand = cp . map cp
cp :: [[a]] -> [[a]]
cp [] = []]
cp (xs:xss) = [x:ys | x <- xs, ys <- cp xss]
```

```
digits :: [Digit]
```

digits :: [Digit]
digits = ['1'..'9']
digits = ['1'..'9']
blank :: Digit -> Bool
blank :: Digit -> Bool
blank = (== '0')

```
blank = (== '0')
```


## Specification (2)

```
valid :: Grid -> Bool
valid g = all nodups (rows g) &&
    all nodups (cols g) &&
    all nodups (boxs g)
```

nodups :: Eq a => [a] -> Bool
nodups [] = True
nodups (x:xs) = x `notElem` xs \&\& nodups xs


## Specification (3)

```
rows :: Matrix a -> [Row a]
rows = id
cols :: Matrix a -> [Row a]
cols [xs] = [[x]|x<-xs]
cols (xs:xss) = zipWith (:) xs (cols xss)
boxs :: Matrix a -> [Row a]
boxs = map ungroup . ungroup .
    map cols .
    group .map group
```


## group and ungroup

```
ungroup = concat
group [] = []
group (x:y:z:xs) = [x,y,z] : group xs
[x,y, z, xs ] \longrightarrow[[x,y,z], group xs ]
```


## rows and cols

type Matrix a = [Row a]

$$
[[\mathrm{a}]]
$$

rows :: Matrix a -> [Row a]
rows :: Matrix a -> Matrix a
rows = id
id : identity function
If a matrix is given by a list of its rows
tt returns the same matrix

```
cols :: Matrix a -> [Row a]
cols :: Matrix a -> Matrix a
cols [xs] \(=[[x] \mid x<-x s]\)
cols (xs:xss) = zipWith (:) xs (cols xss)
```

transpose of a matrix

```
cols :: Matrix a -> [Row a]
cols :: Matrix a -> Matrix a
cols [xs] = [[x]|x<- xs]
cols (xs:xss) = zipWith (:) xs (cols xss)
```


boxs :: Matrix a -> [Row a]
boxs :: Matrix a -> Matrix a
boxs $=\quad$ map ungroup .
ungroup .
map cols .
group .
map group

$$
\begin{aligned}
& \text { type Matrix } \mathrm{a}=[\text { Row } a] \\
& \text { type Row } \mathrm{a}=[\mathrm{a}]
\end{aligned}
$$

[[a]]
takes a grouped list and ungroups it
group ( $x: y: z: x s$ ) $=[x, y, z]$ :group $x s$ splits a list into groups of three
group xs = take 3 xs : group (drop 3 xs)

## group.map group

## map group

$$
\text { type } \quad \text { Grid } \quad=\quad \text { Matrix Digit }
$$

$$
\begin{aligned}
& \text { [ '6’, '7’, '2’, ‘1', '9’, ‘5', ‘3', ‘4', ‘8’], } \\
& \text { [ '1', ‘9', ‘8’, '3', '4’, '2', ‘5', '6’, '7’], } \\
& \text { [ '8’, ‘5', '9', '7', '6’, '1', ‘4', '2', '3’], } \\
& \text { [ '4', '2', '6’, '8', '5', '3', ‘7’, '9', '1’], } \\
& \text { [ '7', '1', ‘3', '9', '2', '4’, ‘8', ‘5', '6' ], } \\
& \text { [ '9', '6’, ‘1', ‘5', '3', '7’, '2', '8', '4’], } \\
& \text { [ '2', '8’, '7’, '4', '1', '9', ‘6’, '3', '5’ ], } \\
& \text { [ '3', ‘4', ‘5', '2', '8', ‘6', '1', '7', '9'] ] }
\end{aligned}
$$

[ [ [ ['5', ‘3', ‘4'], ['6', ‘7', ‘8'], ['9', ‘1', ‘2'] ], [ ['6', ‘7', ‘2'], ['1', ‘9', ‘5’], ['3', ‘4', ‘8’] ], [ [‘1', ‘9’, ‘8’], [‘3’, ‘4’, ‘2’], [‘5’, ‘6’, ‘7’] ] ], [ [ ['8', ‘5', '9'], ['7’, ‘6’, '1'], ['4', '2', ‘3'] ], [ ['4', '2', ‘6'], ['8', ‘5', '3'], ['7', ‘9', '1’] ], [ ['7', ‘1', ‘3'], ['9', '2', '4’], ['8', ‘5', '6’] ] ], [ [ ['9’, ‘6', ‘1'], ['5’, ‘3', ‘7’], ['2’, ‘8’, ‘4'] ], [ ['2', ‘8', '7’], ['4', ‘1', ‘9'], ['6', ‘3', ‘5'] ],
[ ['3', ‘4', '5'], ['2', '8', '6'], ['1', '7', '9'] ] ] ]

## map cols. group . map group

$$
\text { type Grid }=\text { Matrix Digit }
$$

$$
\begin{aligned}
& \text { [ [ [ ['5', '3', '4'], ['6', '7', '8'], ['9', '1', '2'] ], } \\
& \text { [ ['6', '7', '2'], [1', '9', '5'], ['3', '4', '8'] ], } \\
& \text { [ [ } 1 \text { ', '9', '8'], ['3', '4', '2'], ['5', '6', '7']] ], } \\
& \text { [ [ ['8', '5', '9'], ['7', '6', '1'], ['4', '2', '3'] ], } \\
& \text { [ [4', '2', '6'], [8', '5', '3'], [77, '9', '1'] ], } \\
& \text { [ ['7', '1', '3'], ['9', '2', '4'], ['8', '5', '6'] ] ], } \\
& \text { [ [ ['9', '6', '1'], [ [5', '3', '7'], ['2', '8', '4'] ], }
\end{aligned}
$$

$$
\begin{aligned}
& \text { [ ['3', '4', '5'], [ } 2 \text { ', '8', '6'], ['1', '7', '9'] ] ] ] }
\end{aligned}
$$

$$
\begin{aligned}
& \text { [ [ [ ['5', '3', '4'], ['6', '7', '2'], [ [1', '9', '8'] ], } \\
& \text { [ ['6', '7', '8'], [ [1', '9', '5'], [ [3', '4', '2'] ], } \\
& \text { [ ['9', '1', '2'], ['3', '4', '8'], ['5', '6', '7'] ] ], } \\
& \text { [ [ [8', '5', '9'], [ [4', '2', '6'], ['7', '1', '3'] ], } \\
& \text { [ [77', '6', '1'], [8', '5', '3'], [ [9', '2', '4'] ], } \\
& \text { [ [4', '2', '3'], ['7', '9', '1'], [8', '5', '6'] ] ], } \\
& \text { [ [ ['9', '6', '1'], [ [2', '8', '7'], ['3', '4', '5'] ], } \\
& \text { [ ['5', '3', '7'], [4', '1', '9'], ['2', '8', '6'] ], } \\
& \text { [ ['2', '8', '4'], [ [6', '3', '5'], [ [1', '7', '9'] ] ] ] }
\end{aligned}
$$

## ungroup . map cols . group . map group

ungroup

| ], | [ [ ['5', '3', '4'], ['6', '7', '2'], ['1', '9', '8'] ], |
| :---: | :---: |
| 6', '7', '8'], ['1', '9', '5'], ['3', '4', '2'] ], | [ ['6', '7', '8'], ['1', '9', '5'], ['3', '4', '2'] ], |
| ['9', '1', '2'], ['3', '4', '8'], ['5', '6', '7'] ] ], | [ ['9', '1', '2'], ['3', '4', '8'], ['5', '6', '7'] ] |
| [ [ ['8', '5', '9'], ['4', '2', '6'], ['7', '1', '3'] ], | [ ['8', '5', '9'], ['4', '2', '6'], ['7', '1', '3'] ], |
| [ ['7', '6', '1'], ['8', '5', '3'], ['9', '2', '4'] ], | [ ['7', '6', '1'], ['8', '5', '3'], ['9', '2', '4'] |
| [ ['4', '2', '3'], [ 7 ', '9', '1'], ['8', '5', '6'] ] ], | [ ['4', '2', '3'], ['7', '9', '1'], ['8', '5', '6'] ] |
| [ [ [9', '6', '1'], ['2', '8', '7'], ['3', '4', '5'] ], | [ ['9', '6', '1'], ['2', '8', '7'], ['3', '4', '5'] ], |
| [ ['5', '3', '7'], ['4', '1', '9'], ['2', '8', '6'] ], | [ ['5', '3', '7'], ['4', '1', '9'], ['2', '8', '6'] |
| [ ['2', '8', '4'], ['6', '3', '5'], ['1', '7', '9'] ]] ] | [ ['2', '8', '4'], ['6', '3', '5'], ['1', '7', '9'] ] |

type Grid $=$ Matrix Digit $\Rightarrow$ [Row Digit] $\Rightarrow$ [[Digit $]$ ]

## map ungroup . ungroup . map cols . group . map group

map ungroup

|  | [ '5’, '3', ‘4' , '6’, '7', '2’, '1', '9', '8' ], |
| :---: | :---: |
| [ ['6', '7', '8'], ['1', '9', '5'], ['3', '4', '2'] ], | [ '6', '7', '8', '1', '9', '5', '3', '4', '2' |
| [ ['9', '1', '2'], ['3', '4', '8'], ['5', '6', '7'] ] | [ '9', '1', '2', '3', '4', '8', '5', '6', '7' |
| [ ['8', '5', '9'], ['4', '2', '6'], ['7', '1', '3'] ], | [ '8', '5', '9', '4', '2', '6', '7', '1', '3' |
| [ ['7', '6', '1'], ['8', '5', '3'], ['9', '2', '4'] ], | [ '7, '6', '1', '8', '5', '3', '9', '2', '4' |
| [ ['4', '2', '3'], ['7', '9', '1'], ['8', '5', '6'] ] | [ '4', '2', '3', '7', '9', '1', '8', '5', '6' |
| [ ['9', '6', '1'], ['2', '8', '7'], ['3', '4', '5'] ], | [ '9', '6', '1', '2', '8', '7', '3', '4', '5' |
| [ ['5', '3', '7'], ['4', '1', '9'], ['2', '8', '6'] ], | [ '5', '3', '7', '4', '1', '9', '2', '8', '6' |
| [ ['2', '8', '4'], ['6', '3', '5'], ['1', '7', '9'] ] ] | [ '2', '8', '4', '6', '3', '5', '1', '7', '9' ] |

type Grid $=$ Matrix Digit $\Rightarrow$ [Row Digit] $\Rightarrow$ [[Digit $]$ ]
boxs
map group


| $\mathbf{a}$ | $\mathbf{b}$ | $\mathbf{c}$ | $\mathbf{d}$ |
| :---: | :---: | :---: | :---: |
| $\mathbf{e}$ | $\mathbf{f}$ | $\mathbf{g}$ | $\mathbf{h}$ |
| $\mathbf{i}$ | $\mathbf{j}$ | $\mathbf{k}$ | $\mathbf{l}$ |
| $\mathbf{m}$ | $\mathbf{n}$ | $\mathbf{o}$ | $\mathbf{p}$ |

map ungroup

| a | b | e |
| :---: | :---: | :---: |
| c | d | g |
| i | j | m |
| k |  | 0 |


| $\mathbf{a}$ | $\mathbf{b}$ | $\mathbf{e}$ | $\mathbf{f}$ |
| :---: | :---: | :---: | :---: |
| $\mathbf{c}$ | $\mathbf{d}$ | $\mathbf{g}$ | $\mathbf{h}$ |
| $\mathbf{i}$ | $\mathbf{j}$ | $\mathbf{m}$ | $\mathbf{n}$ |
| $\mathbf{k}$ | l | o | $\mathbf{p}$ |

group

| $\mathbf{a}$ | $\mathbf{b}$ | $\mathbf{c}$ | $\mathbf{d}$ |
| :---: | :---: | :---: | :---: |
| $\mathbf{e}$ | $\mathbf{f}$ | $\mathbf{g}$ | $\mathbf{h}$ |
| $\mathbf{i}$ | $\mathbf{j}$ | $\mathbf{k}$ | $\mathbf{l}$ |
| $\mathbf{m}$ | n | $\mathbf{o}$ | p |

map cols
ungroup

| $\mathbf{a}$ | $\mathbf{b}$ | e | f |
| :---: | :---: | :---: | :---: |
| $\mathbf{c}$ | d | g | h |
| $\mathbf{i}$ | $\mathbf{j}$ | m | n |
| k | I | o | p |

boxs

| a |  | C | d | boxs | a | b | e | $f$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| e |  | g | h |  | c | d | g | h |
| i |  | k | I |  | i | j | m | n |
| m |  | 0 | $p$ |  | k | I | 0 | $p$ |
| a |  | c | d | boxs | a | b | e | $f$ |
| e |  | g | h |  | c | d | $g$ | h |
| i |  | k | I |  | i | J | m | n |
| m |  | 0 | p |  | k | I | 0 | p |


| a | b | c | d | cols | a | e |  | i | m |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| e | f | g | h |  | b | $f$ |  | j | n |
| i | j | k | I |  | C | g |  | k | 0 |
| m | n | 0 | $p$ |  | d | h |  | I | p |


| a | b | c |
| :---: | :---: | :---: |
| e | f | $g$ |
| i | j | k |
| m | n | 0 |



|  | e | m |  |
| :---: | :---: | :---: | :---: |
| b |  |  |  |
| C | g |  |  |
| d | h |  |  |

rows, cols, boxs

| $a$ | $b$ | $c$ | $d$ |
| :---: | :---: | :---: | :---: |
| $e$ | $f$ | $g$ | $h$ |
| $i$ | $j$ | $k$ | $l$ |
| $m$ | $n$ | $o$ | $p$ |




| $a$ | $b$ | $c$ | $d$ |
| :---: | :---: | :---: | :---: |
| $e$ | $f$ | $g$ | $h$ |
| $i$ | $j$ | $k$ | $l$ |
| $m$ | $n$ | $o$ | $p$ |

cols

| a | e |  | m |
| :---: | :---: | :---: | :---: |
| b | $f$ |  | n |
| c | g | k | 0 |
|  | h |  |  |


| $a$ | $b$ | $c$ | $d$ |
| :---: | :---: | :---: | :---: |
| $e$ | $f$ | $g$ | $h$ |
| $i$ | $j$ | $k$ | $I$ |
| $m$ | $n$ | $o$ | $p$ |

boxs

$$
\begin{array}{c|ccc}
a & b & e & f \\
c & d & g & h \\
\mathbf{i} & j & m & n \\
k & l & o & p
\end{array}
$$

## nodups

```
nodups :: (Eq a) => [a] -> Bool
nodups [] = True
nodups (x:xs) = x `notElem` xs && nodups xs
notElem :: (Eq a) => a -> [a] -> Bool
notElem x xs = all (/= x) xs
all p = and . map p
```

nodups :: (Eq a) => [a] -> Bool
nodups [] = True
nodups (x:xs) = all (/=x) xs \&\& nodups xs
all $p=$ and. $\operatorname{map} p$

## nodups

```
[ '6', '7', '2', '1', '9', '5', '3', '4', '8'] nodups (x:xs) =
    x `notElem` xs && nodups xs
    '6', ['7', '2', '1', '9', '5', '3', '4', '8']
    '6', '7', [ '2', '1', '9', '5', '3', '4', '8']
                                    all p = and . map p
    '6', '7', '2', [ '1', '9', '5', '3', '4', '8']
    '6', '7', '2', '1', [ '9', '5', '3', '4', '8']
    '6', '7', '2', '1', '9', ['5', '3', '4', '8']
    '6', '7', '2', '1', '9', '5', [ '3', '4', '8']
```


## nodups

$[66$ ', '7', '2', '1', '9', '5', '3', '4', '8'] nodups (x:xs) $=$
$\quad x$ notElem` xs \&\& nodups xs
'6', '7’, '2', '1', '9', ‘5', [ '3', ‘4’, '8’]
notElem $x$ xs $=$ all (/=x)xs
'6', '7’, '2', '1', '9', ‘5', '3', [‘4’, '8']
all $p=$ and. $\operatorname{map} p$
'6’, '7’, '2', '1', ‘9', ‘5', ‘3', ‘4’, ['8’]
'6’, '7’, '2’, ‘1’, '9’, ‘5’, ‘3’, ‘4’, ‘8’ []

## References

[1] ftp://ftp.geoinfo.tuwien.ac.at/navratil/HaskellTutorial.pdf
[2] https://www.umiacs.umd.edu/~hal/docs/daume02yaht.pdf


[^0]:    Copyright (c) 2016 Young W. Lim.
    Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

