

Pretty Printing with Lazy Sequences

Olaf Chitil

The University of York

```
if True
then if True then True else True
else
    if False
    then False
    else False
```

A Pretty Printing Library

functions for compositional definition of specific pretty printers

text :: String → Doc

line :: Doc

(<>) :: Doc → Doc → Doc

(nest) :: Int → Doc → Doc

group :: Doc → Doc

pretty :: Int → Doc → String

tree structure



Doc



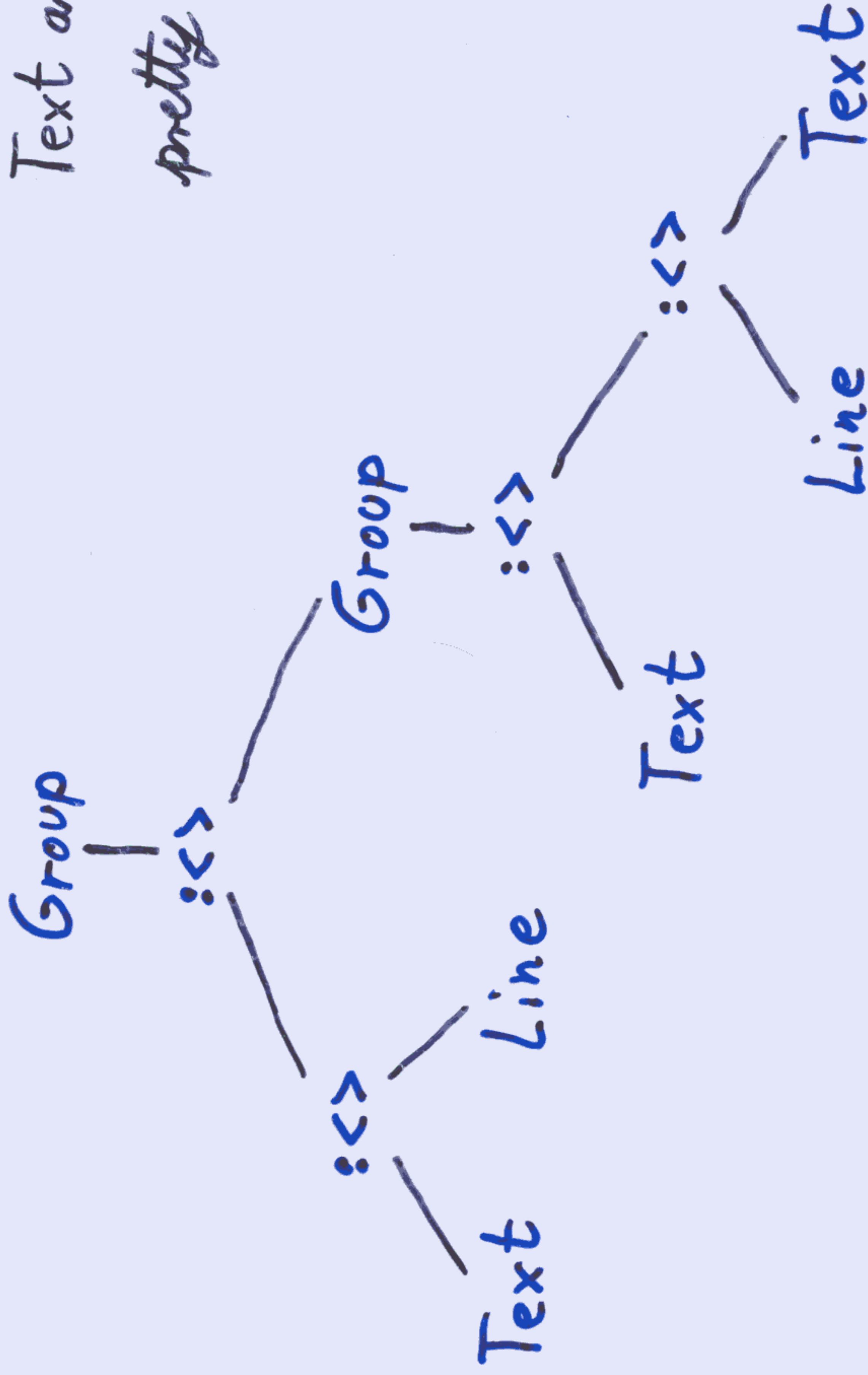
formatted String

Direct Representation

data Doc = Text String
| Line
| Doc :<> Doc
| Group Doc

Recursive Implementations

Text and Line have width 1
pretty print within width 3



Recursive Implementations

	time	space
Version 1: separate test if group fits		
a) width group $\leq$ available width	2^n	n
b) traverse group only up to available width	$n \cdot w$	w
Version 2: determine group width in parallel	n	n

Represent Document as Sequence

type Doc = [Token]

data Token = Text String

| Line

| Open

| Close

Version 3: Iterative Implementation

	Open	Text	Line	Open	Text	Line	Text	Close	Close
1 →	0	0	0	1	1	1	1	0	0
2 →	3	3	2	3	3	2	1	0	0
3 →	0	0	1	2	2	3	4	5	5
4 ←	[]	[S]	[S]	[S]	[S, S]	[S, S]	[S, S]	[S]	[]

- 1 depth of fitting groups
- 2 remaining space on line
- 3 absolute position (document in single line)
- 4 absolute positions at Close tokens

Version 4: move decision from Open to Close

	Open	Text	Line	Open	Text	Line	Text	Line	Text	Close	Close
1 →	0	0	0	0	1	1	1	0	0	0	0
2 →	3	3	2	3	3	2	1	0	0	0	0
3 →	0	0	1	2	2	3	4	5	5	5	5
4 →	[]	[3]	[3]	[3]	[5, 3]	[5, 3]	[5, 3]	[5, 3]	[3]	[]	[]
5 →	[]	[F]	[F]	[F]	[T, F]	[T, F]	[T, F]	[T, F]	[F]	[]	[]

- 1 depth of fitting groups
- 2 remaining space on line
- 3 absolute position (document in single line)
- 4 absolute maximal end positions
- 5 fits infos

Final version: earlier decision through lazy dequeues

	Open	Text	Line	Open	Text	Line	Text	Line	Text	Close	Close
1 →	0	0	0	0	1	1	1	1	0	0	0
2 →	3	3	2	3	3	2	1	0	0	0	0
3 →	0	0	1	2	2	3	4	5	0	0	0
4 →	[]	[3]	[3]	[3]	[S, 3]	[S, 3]	[S]	[S]	[]	[]	[]
5 ←	[]	[F]	[F]	[F]	[T, F]	[T, F]	[T]	[T]	[]	[]	[]

Red annotations: A bracket groups the last three rows (4, 5, 6) with a red '3' below it. Arrows point from the red '3' in row 2 to the red '3' in row 3, and from the red '0' in row 3 to the red '3' in row 4.

- 1 depth of fitting groups
- 2 remaining space on line
- 3 absolute position (document in single line)
- 4 absolute maximal end positions
- 5 fits info

Conclusions

- purely functional implementation as efficient as Appen's imperative implementation
- library used by and distributed with GHC 3.8
- implicit recursive control
 - ⇒ explicit data structure (manipulate, exchange)
- lazy non-inductive data structures exist