

# FliPpr: A Prettier Invertible Printing System

Kazutaka Matsuda (Univ. of Tokyo) Meng Wang (Chalmers Univ. of Tech.)

## Motivation

Consistent pretty-printer and parser:

$$\text{parse (prettyPrint ast)} = \text{ast}$$

## Our Proposal: FliPpr

pretty-printer +  $\alpha$

Program Inversion

parser

Based on Wadler (2003)'s combinators with additional information for parsing

CFG with Actions

## Overview of FliPpr

data AST = One | Sub AST AST

pretty-printer +  $\alpha$

```
pprMain x = ppr 0 x
ppr i x = manyParens (go i x)
go i One = text "1"
go i (Sub x y) = parensIf (i >= 6) (group (
  ppr 5 x <> nest 2 (line <>
    text "-" <> sp' <> ppr 6 y)))
manyParens d = d <+ parens (manyParens d)
line' = line <+ text ""
...
```

Extra Parens

Extra Spaces

<+>

Specialization

Supercompilation [Sørensen et al. 96]

Forgetting Layouts

text	s	→	s	line	→	sp
nest	i x	→	x	<>	→	++
group	x	→	x	<+>	→	?

x <+ y = x

Sub One One is printed as

1 - 1 or 1 - 1

nondeterministic printer

```
pprMain x = ppr0 x
ppr0 x =
  go0 x ? "(" ++ nil ++ ppr0 x ++ nil ++ ")"
go0 One = "1"
go0 (Sub x y) =
  ppr5 x ++ line' ++ "-" ++ sp' ++ ppr6 y
line' = sp ? ""
...
```

manyParens and parens are fused away

Nondet. Choice

x ? y = x  
x ? y = y

Sub One One is printed as

1 - 1 1 - 1 1 (1 - (1)) ...

Grammar-based Inversion

[Matsuda et al. 10]

parser

```
PprMain → Ppr0 { $1 }
Ppr0 → Go0 { $1 }
      | "(" Sp' Ppr0 Sp' ")" { $3 }
Go0 → "1" { One }
Go0 → Ppr5 Line' "-" Sp' Ppr6 { Sub $1 $5 }
Line' → Sp {}
      | "" {}
...
```

Users can use their favorite parser generators

## Advantages

- AST centric
- Fine-grained control on pretty-printing
- Efficient

## Future Work

- Lexing phase
- Off-side rules like Haskell
- Fusion for nondet. programs