

Pal: A Grammar-Based Program Inversion System

Kazutaka Matsuda (The University of Tokyo)
joint work with Shin-Cheng Mu, Zhenjiang Hu, and Masato Takeichi

Motivation

Program Inversion: $p(x) = \dots \longrightarrow p^{-1}(y) = \dots$

What's Pal

Pal: An Implementation of Grammar-Based Inversion

(Available at: <http://www.ipl.t.u-tokyo.ac.jp/~kztk/PaI/>)

```
$ ./pai snoc.txt > invsnoc.hs
$ ghci invsnoc.hs
...
*Main> :t snoc
snoc :: (List a, a) -> List a
*Main> snoc (Cons 1 Nil, 2)
Cons 1 (Cons 2 Nil)
*Main> :t inv_Fsnoc
inv_Fsnoc :: List a -> (List a, a)
*Main> inv_Fsnoc (Cons 1 (Cons 2 Nil))
(Cons 1 Nil, 2)
```

```
snoc.txt
snoc(Nil,b)
= Cons b Nil
snoc(Cons a x,b)
= Cons a (snoc x b)
```



```
invsnoc.hs
inv_Fsnoc = ...
-- Inverse Program
-- based-on Parsing
...
```

Experiments:
(with GHC 6.8.2)

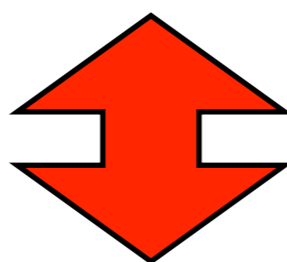
Program	#input	handwritten	derived	speed ratio
snoc	8M	0.67s	0.95s	1.4
double	10M	0.11s	0.23s	2.1
zip	8M	0.28s	0.70s	2.5
runlength	9M	0.33s	0.76s	2.3

Acceptable
Overhead!

Underlying Theory: Grammar-based Inversion

$$\frac{\frac{a = 1 \vdash a \downarrow 1 \quad \frac{\frac{b = 2 \vdash b \downarrow 2 \quad \vdash Nil \downarrow Nil}{b = 2 \vdash Cons b Nil \downarrow Cons 2 Nil}}{x = Nil, b = 2 \vdash snoc(x,b) \downarrow Cons 2 Nil}}{a = 1, x = Nil, b = 2 \vdash a:snoc(x,b) \downarrow Cons 1 (Cons 2 Nil)}}{x = Cons 1 Nil, b = 2 \vdash snoc(x,b) \downarrow Cons 1 (Cons 2 Nil)}$$

bijection between
evaluation & production



$$\frac{\frac{\frac{\top \xrightarrow{*} 2 \quad Nil \xrightarrow{*} Nil}{Exp1 \xrightarrow{*} Cons 2 Nil}}{\top \xrightarrow{*} 1 \quad Snoc \xrightarrow{*} Cons 2 Nil}}{Exp2 \xrightarrow{*} Cons 1 (Cons 2 Nil)}}{Snoc \rightarrow Cons 1 (Cons 2 Nil)}$$

Program

```
snoc(Nil,b)
= Cons b Nil
snoc(Cons a x,b)
= Cons a (snoc x b)
```

Grammar

```
Snoc → Exp1
Snoc → Exp2
Exp1 → Cons ⊤ Nil
Exp2 → Cons ⊤ Snoc
Nil → Nil
```

Status

- Inversion with Regular Tree Grammar
- Inversion with Other Grammars