

Program Transformation and Analysis

Benjamin Brandt Ohrt, zpn492

April 27, 2019

1 Introduction

This is the first of five weekly assignments in the course Program Transformation and Analysis (PAT) at Copenhagen University. The course professor is Robert Glück. The course is held in block 4, 2019.

In this assignment the focus is on Programs as Data Objects and Reversible Computing.

Before we start on the assignment we introduce an inverse interpreter *invint* and a program inverter *invtrans*. For this introduction we need to define the following: Let p be a program written in a language L , x an input into p and y an output from p . We define p^{-1} as the inverse of p such that given y as input to p^{-1} produces x .

1.1 Inverse Interpreter

An inverse interpreter can be defined with the following definition [1, p. 271]:

$$[[invint]][p, y] = x \tag{1}$$

1.2 Program Inverter

A program inverter can be defined with the following definition [1, p. 271]:

$$\begin{aligned} [[invtrans]][p] &= p^{-1} \\ [[p^{-1}]] [y] &= x \end{aligned} \tag{2}$$

Exercise 1

Define a trivial program inverter given an inverse interpreter.

$$[[invtrans']] = \lambda p. \lambda y. [[invint]][p, y] \quad (3)$$

Exercise 2

Define a trivial inverse interpreter given a program inverter.

$$[[invint']] = \lambda [p, y]. [[Lint]] [[[[invtrans]]p], y] \quad (4)$$

Where Lint is an L-interpreter.1

Exercise 3

Run the Janus-program fib forward (call main_fwd) by hand. Trace the store (n, x1, x2) after each statement.

Recursion	State	n	x1	x2	branch
0	initial	4	0	0	else
1	initial	3	0	0	else
2	initial	2	0	0	else
3	initial	1	0	0	else
4	initial	0	0	0	then
4	terminating	0	1	1	fi:true
3	terminating	0	1	2	fi:false
2	terminating	0	2	3	fi:false
1	terminating	0	3	5	fi:false
0	terminating	0	5	8	fi:false

Exercise 4

Run the Janus-program fib backward (call main.bwd) by hand. Trace the store (n, x1, x2) after each statement.

Recursion	State	n	x1	x2	branch
0	initial	0	5	8	else
1	initial	0	3	5	else
2	initial	0	2	3	else
3	initial	0	1	2	else
4	initial	0	1	1	then
4	terminating	0	0	0	fi:true
3	terminating	1	0	0	fi:false
2	terminating	2	0	0	fi:false
1	terminating	3	0	0	fi:false
0	terminating	4	0	0	fi:false

Exercise 5

Describe formally but concisely, how you interpreted each statement ($+=$, $-=$, $<=>$, if, call) in the Janus-program fib in the fwd and bwd direction.

fwd	$+=$	$-=$	$<=>$	if	call
bwd	$-=$	$+=$	$<=>$	fi	uncall

Switching sourcecode between a Janus-program p and its inverse p^{-1} can be done by interpreting statements using above table. After the interpretation you just swap *if* and *fi*. and reverse the order of operations within the *then* and *else* statement. Ex. a list l of operations in p where $l := \{x1+=1; x2+=1\}$ is a list l' in p^{-1} where $l' := \{x2+=1; x1+=1\}$

References

- [1] Mogensen, Torben, Schmidt, David, Sudborough, I. Hal (Eds.). *The Essence of Computation - Complexity, Analysis, Transformation*. Springer-Verlag, Berlin Heidelberg, 2002.