PPL 5

1b:

claim: the procedure *pipe$* is CPS equivalent to the procedure *pipe* and for each continuation *cont:*

(**pipe**$ fs cont) = (**cont** (**pipe** fs))

proof by induction on the size of function list:

Base case:

Induction assumption: assume

Step:

suppose

therefore, the induction assumption applies, so the code above is CPS equivalent to

which is equivalent to:

2a:

We will say two lazy lists *lzl1* and *lzl2* are equivalent if:

* , when calculating the th item of the lists:

2b:

Proof by induction on the lazy-lists length:

Base case:

so , therefore by our criterion

Induction assumption:

suppose ;

Step:

suppose . We need to prove :

let us mark

while :

and while :

By the induction assumption,

so, we can mark and .

we will calculate for

fibs1:

(lambda (a b)

(cons-lzl a (lambda () (fibgen b (+ a b)))) **k2 k1**) k2+k1

fibs2:

On the th calculation of fibs2 we get the closure

(lambda ()( lz-lst-add (tail fibs2) fibs2))

where ) and

hence when *lz-lst-add* will be executed we it will return .

therefore and by our criterion it means

.

3.3.a: