Computability Assignment
Year 2012/13 - Number 5

Please keep this file anonymous: do not write your name inside this file.

More information about assignments at http://disi.unitn.it/~zunino/teaching/computability/assignments

Please do not submit a file containing only the answers; edit this file, instead, filling the answer sections.

1 Question

(I am re-proposing this exercise since only a few students solved it. This exercise is rather important, since it involves a reasoning which frequently appeared in past exam questions. While we shall see more examples of these concepts in class, it would be useful to start exercising on that. If you have already sumbitted an answer, skip this and do *not* resubmit your answer please.)

Let $\mathcal{F}$ be the set of partial functions $\{f \in (\mathbb{N} \rightsquigarrow \mathbb{N}) | \forall x \in \mathbb{N}.\ f(2 \cdot x) = x\}$.

- Define two distinct partial functions f_1, f_2 which belong to $\mathcal{F}$. (I.e, provide two such examples.)
- Define two distinct partial functions g_1, g_2 which do *not* belong to $\mathcal{F}$. (I.e, provide two such examples.)
- Define a partial function $f \in \mathcal{F}$, and consider the set of its *finite* restrictions $\mathcal{G} = \{g \in (\mathbb{N} \rightsquigarrow \mathbb{N}) | g \subseteq f\ \wedge\ \mathsf{dom}(g) \text{ finite}\}$.
  - Define two distinct partial functions h_1, h_2 which belong to $\mathcal{G}$. (I.e, provide two such examples.)
  - Prove whether $\mathcal{F} \cap \mathcal{G} = \emptyset$.

1.1 Answer

Let $\mathcal{F}$ be the set of partial functions $\{f \in (\mathbb{N} \rightsquigarrow \mathbb{N}) | \forall x \in \mathbb{N}.\ f(2 \cdot x) = x\}$.

- Define two distinct partial functions f_1, f_2 which belong to $\mathcal{F}$. (I.e, provide two such examples.)

- $f_1(x) = \begin{cases} \frac{x}{2} & even \\ x & odd \end{cases}$
- $f_2(x) = x$
  (RZ: f_2 is wrong...)

- Define two distinct partial functions g_1, g_2 which do *not* belong to $\mathcal{F}$. (I.e, provide two such examples.)

  - $g_1(x) = \begin{cases} x & x < 100 \\ undef. & ow. \end{cases}$
  - $g_2(x) = \begin{cases} x & x \leq 100 \\ undef. & ow. \end{cases}$

- Define a partial function $f \in \mathcal{F}$, and consider the set of its *finite* restrictions $\mathcal{G} = \{g \in (\mathbb{N} \rightsquigarrow \mathbb{N}) | g \subseteq f \ \wedge \ \mathsf{dom}(g) \text{ finite}\}$.

  - Define two distinct partial functions h_1, h_2 which belong to $\mathcal{G}$. (I.e, provide two such examples.)
    * $h_1(x) = \begin{cases} f_1(x) & x \leq 100 \\ undef. & ow. \end{cases}$
    * $h_2(x) = \begin{cases} f_2(x) & x \leq 100 \\ undef. & ow. \end{cases}$
  - Prove whether $\mathcal{F} \cap \mathcal{G} = \emptyset$.
    * The functions defined in $\mathcal{F}$, even if they are partial, are defined over $\mathbb{N}$, which means that the domain of such functions is infinite. (RZ: not really, they are defined only on even naturals.) By definition, on the other hand, the functions defined in the set $\mathcal{G}$ have the precise restriction that the domain must be finite. So we can conclude that $\mathcal{F} \cap \mathcal{G} = \emptyset$.

2 Question

Consider the following function:

$$f(n) = \sum_{i=0}^{n} i^2 + i$$

Write a FOR loop implementing f, then translate it in the λ-calculus as program F_1.

Then, write a recursive Java-like function implementing f, and translate it in the λ-calculus as program F_2.

2.1 Answer

$s := 0;$
FOR $i := 0$ TO n DO
    $s := s + ((i * i) + i);$
RETURN $s;$

F1 := λn.**Fst** (n **G1** (**Cons** [0] [0]))
G1 := λp.**Cons** **(Add** (**Fst** p) (**Add** (**Mul** (**Snd** b) (**Snd** b)) (**Snd** b))) **(Succ** (**Snd** p))
(RZ: you want to loop $n + 1$ times)

foo(n) {
    if $n = 0$ then return 0;
    else return $(((n * n) + n) + \text{foo}(n - 1));$
}
F2 := Θ(λg.(λn.**IsZero** n [0] (**Add** (**Add** (**Mul** n n) n) (g (**Pred** n)))))

Note.
I used the [,] instead of the symbols that we used in class for numbers, since I didn't find them in LyX.

3 Question

Consider the following function:

$$f(n) = \begin{cases} x^2 + y & \text{if } n = \mathsf{pair}(\mathsf{inL}(x), y) \\ x + 4 \cdot y & \text{if } n = \mathsf{pair}(\mathsf{inR}(x), y) \end{cases}$$

Convince yourself that f is defined for all naturals n, i.e. it is total.

Write a λ-term implementing function f, exploiting the programs $Pair, Proj1, Proj2, InL, InR, Case, \ldots$ we saw in class (also defined in the notes).

3.1 Answer

F := λn.(λxy.**(Case** x (**Add** (**Mul** x x) y) (**Add** x (**Mul** [4] y))) (**Proj1** n) (**Proj2** n))

(RZ: you can't take x, y in this way... this will not work as you expect)