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

(Already solved)

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

```
int f(int n)
{
        int t = 0;
        for(int i = 0; i <= n; i++)
                t += i*i + i;
}
```

$ForBody = \lambda p.(\lambda it.(Pair(Succ\,i)(Add\,t(Add(Mul\,i\,i)\,i)))(Proj1\,p)(Proj2\,p)$

(RZ: OK, this is an advanced trick)

The $ForBody$ part receives a pair $\langle i, t\rangle$and returns the updated values. An intermediate lambda is used to rename $(Proj1\,p)$ and $(Proj2\,p)$ into i and t.

$F = \lambda n.Proj2\,((Succ\,n)\,ForBody\,(Pair\,\ulcorner 0 \urcorner \ulcorner 0 \urcorner))$

(RZ: it's correct, but in general we prefer to use $Cons/Fst/Snd$ rather than $Pair/Proj1/Proj2$)

The loop is actually executed $n+1$times, so there is an invocation to $Succ$. The $ForBody$ is then applied to that Church numeral, and it is initialized with $i = 0$ and $t = 0$. At the end of the loop, the second element, which represents t, is extracted and returned.

```
int f(int n)
{
        return (n*n + n) + (n == 0 ? 0 : f(n-1));
}
```

Lambda version: $F_2 = \Theta\lambda gn.Add(Add(Mul\,n\,n)\,n)(IsZero\,n\,\ulcorner 0 \urcorner(g(Pre\,c\,n)))$

The lambda version is almost a direct translation of the Java-like program, except that the $Theta$ function is used to define the recursive function.

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

$f = \lambda n.Case(Proj1n)(\lambda x.Add(Mulxx)(Proj2n))(\lambda x.Addx(Mul\ulcorner 4\urcorner(Proj2n)))$

If the first element of n is inleft, the calculation is performed using the first formula. This is done by passing an abstraction to $Case$ that will receive the content of the inleft via its argument x, something similar is done for the inright case. $Proj1$ and $Proj2$ are used to extract the desired element of a pair.