Llenguatges de Programació, FIB, 26 de gener de 2018

Possibles solucions

1. Entrants

1. Eren la mateixa persona: Haskell B. Curry (1900–1982).
2. Comprovem que $NEG\ T \equiv F$ i que $NEG\ F \equiv T$:

$$\begin{cases} NEG\ T \equiv \lambda x.(xFT)(\lambda xy.x) \equiv \lambda z.(zFT)(\lambda xy.x) \equiv_\beta (\lambda xy.x)(FT) \equiv_\beta F \\ NEG\ F \equiv \lambda x.(xFT)(\lambda xy.y) \equiv \lambda z.(zFT)(\lambda xy.y) \equiv_\beta (\lambda xy.y)(FT) \equiv_\beta T \end{cases}$$

3. Així:

```
def compose(f, g):
    return lambda x: f(g(x))
```

4. La solució depèn del vostre llenguatge però no depèn del seu mode d'execució ni paradigma de programació.

2. Inferència de tipus

Fent l'arbre decorat, posant les equacions que en surten i resolent-les (no ho faig), s'obté que

$$fsmap\ ::\ a\ \to [a \to a] \to a$$

Errors comuns:

- No posar el primer $ a l'arbre: El sistema d'inferència de tipus no pot donar-li la semàntica d'obrir parèntesi. $ és una funció com qualsevol altra.
- No rebatejar les variables de tipus que apareixen a les fulles de l'arbre.

3. Arbre general a arbre binari en Python

```
def gal2bin(t):
    x, children = t
    if children == []:
        return [x, None, None]
    else:
        roots = map(gal2bin, children)
        for i in range(len(roots) - 1):
            roots[i][2] = roots[i+1]
        return [x, roots[0], None]
```

4. Programa misteriós

1. El tipus de l'operador de composició és

   $$(.) \;::\; (b \to c) \to (a \to b) \to a \to c$$

2. Donada una seqüència de línies, el programa escriu, les cinc (o menys, si n'ho hi ha prou) línies més petites en ordre lexicogràfic decreixent que tenen longitud parell.

3. La transformació és:

   $$main = \mathbf{getContents} \gg\!= \backslash bar \to \mathbf{putStr}\ (foo\ bar)$$

5. Fluffy i misty

$$\begin{aligned} &furry\ f\ \mathbf{Nothing} = \mathbf{Nothing} && -- 5.1\\ &furry\ f\ (\mathbf{Just}\ x) = \mathbf{Just}\ (f\ x) \end{aligned}$$

$$furry = \mathbf{map} \qquad -- 5.2$$

$$faun\ f = banana\ (unicorn\ .\ f) \qquad -- 5.3$$

$$\begin{aligned} &banana\ f\ \mathbf{Nothing} = \mathbf{Nothing} && -- 5.4\\ &banana\ f\ (\mathbf{Just}\ x) = f\ x\\ &unicorn = \mathbf{Just} \end{aligned}$$

$$\begin{aligned} &banana\ f\ xs = \mathbf{concat}\ \$\ \mathbf{map}\ f\ xs && -- 5.5\\ &unicorn\ x = [x] \end{aligned}$$

[Problema extret de *20 Intermediate Haskell Exercises* de Tony Morris.]