

Llenguatges de Programació, FIB, 16 de juny de 2021

L'examen dura tres hores. Es valorarà la concisió, claredat i brevetat a més de la completesa i l'exactitud de les respostes. Contesteu cada problema en un full diferent i poseu el vostre nom a cada full. Feu bona lletra.

1. Pragmàtica dels LPs

(1 punt)

En el seu assaig *Why Pascal is Not My Favorite Programming Language* (1981), Brian Kernighan (autor del famós llibre *The C Programming Language*) critica (entre d'altres) que el llenguatge Pascal no tingui suport per a instruccions tipus **break** o **continue**.

Des del punt de vista de la pragmàtica dels llenguatges de programació, doneu un argument a favor i un argument en contra d'aquesta decisió de disseny.

Limiteu la vostra resposta a 10 línies (menys també pot estar molt bé!).

2. Haskell

(2 punts)

En matemàtiques, un *monoide* és una estructura algebraica consistent en un conjunt dotat d'una operació interna associativa i d'un element neutre. En Haskell, la classe *Monoide* es pot definir així:

```
class Monoide m where
  neutre :: m
  (◊)    :: m → m → m      -- (la caixeta ◊ és un símbol d'operador binari)
```

Les lleis de la classe *Monoide* són les següents:

- (1) $\text{neutre} \diamond x = x$
- (2) $x \diamond \text{neutre} = x$
- (3) $x \diamond (y \diamond z) = (x \diamond y) \diamond z$

per a tots $x, y, z :: m$.

1. Feu que el tipus **Int** instanciï la classe *Monoide* de dues maneres diferents.
2. Feu que el tipus llista de ts ($[t]$ o **List** t) instanciï la classe *Monoide* i demostreu que la vostra instanciació compleix les lleis dels monoides.
3. Sabent que el tipus t és un monoide, feu que el tipus **Maybe** t instanciï la classe *Monoide* i demostreu que la vostra instanciació compleix les lleis dels monoides.

3. Inferència de tipus

(2 punts)

1. Apliqueu l'algorisme d'inferència de tipus per trobar el tipus de la funció f següent:

$$\begin{aligned} f\ 0\ c &= c\ 1 \\ f\ n\ c &= f\ (n - 1)\ (c\ .\ (*\ n)) \end{aligned}$$

Per a fer-ho, dibuixeu els arbres de sintàxi, etiqueteu els nodes amb els seus tipus i genereu i resoleu metodològicament les restriccions de tipus. Expliqueu els passos realitzats.

Suposeu que tots els literals i operadors numèrics són sobre el tipus **Int**.

2. Quin és el tipus de $f\ n\ \text{id}$?
3. I, ja que hi som, què calcula $f\ n\ \text{id}$?

4. Lambda càlcul

(2 punts)

1. Avalueu aquesta expressió en ordre normal i en ordre aplicatiu:

$$(\lambda x . \lambda y . (\text{ADD } y ((\lambda z . (\text{MUL } x z)) 3))) 7 5$$

Podeu suposar que ADD i MUL són funcions que sumen i multipliquen respectivament.

2. Avalueu aquesta expressió en ordre normal i en ordre aplicatiu:

$$(\lambda y . 5)((\lambda x . x x)(\lambda x . x x))$$

5. Compilació

(1 punt)

Escriviu, utilitzant notació ANTLR, una gramàtica LL(1) pel llenguatge dels parèntesis i claudàtors ben tancats.

Per exemple: “[([() []])”, “[” i “((()))” pertanyen al llenguatge, però “[()”, “[([() []])”, “[()”, “” i “unicorn” no.

Justifiqueu perquè la vostra gramàtica és LL(1).

6. Subtipos y herencia

(2 punts)

Considerad las definiciones siguientes en C++:

```
class A {  
public:  
    virtual void m1();  
    A m2();  
};
```

```
A a;  
A* p;
```

```
class B : public A {  
public:  
    virtual void m1();  
    void m3();  
};
```

```
B b;  
B* q;
```

Explicad si serían legales las siguientes secuencias de instrucciones:

- (1) `b.m1(); b.m3();`
- (2) `b = b.m2(); b.m3();`
- (3) `a = b; a.m3();`
- (4) `p = new B; p→m3();`
- (5) `q = new B; q→m3();`
- (6) `q = p; q→m3();`
- (7) `p = q; p→m3();`

¿Qué método *m1* (el de *A* o el de *B*) se ejecuta en los casos siguientes? Explicad porqué.

- (8) `p = new A; p→m1();`
- (9) `p = new B; p→m1();`
- (10) `a = b; a.m1();`