

Control 2 – Lenguajes de Programación

Departamento de Ciencias de la Computación

Universidad de Chile

Profesor: Éric Tanter

31 de Mayo del 2010

2 horas / 2 puntos por pregunta

1. Una serie de Taylor permite aproximar numéricamente una función matemática. En esta pregunta implementará la serie geométrica de Taylor, que corresponde a la siguiente función:

$$\frac{1}{(1-x)} = \sum_{i=0}^{\infty} x^i \quad |x| < 1 \quad (1)$$

(notese que esa definición es válida sólo en el intervalo $] - 1; 1[$, por lo que los valores de x en lo siguiente serán de tipo `Float`)

- La lista (infinita) de los números naturales puede ser definida en Haskell usando

```
nat :: [Int]
nat = [0..]
```

(0.5) ¿En qué casos puede evaluar esta lista? ¿Qué característica de Haskell le permite hacer este tipo de definiciones?

- (0.25) Escriba el código que le permite obtener los primeros 15 números naturales
- (0.25) Defina la función `pow`. Esta función recibe dos parámetros `x` y `n`, y retorna x^n .
- (0.5) Usando la lista de los números naturales y la función `pow`, defina la función `geom` que, dado un valor de `x`, retorna la lista de *todos* los elementos de la serie evaluados en `x`.
`geom(x) = [1, x, x2, x3, ...]`
- (0.5) Defina la función `approx` que le permite calcular la aproximación de la serie geométrica en algún punto `x` hasta cierto orden `n`.

No se olvide incluir la firma de cada función que define.

2. Supongamos que tenemos la siguiente API para interactuar con el beeper del computador:

- `(beep-open)`: inicializa el beeper
- `(beep-close)`: cierra el beeper
- `(beep freq time)`: emite un sonido a la frecuencia `freq` durante el tiempo `time` (en ms).

Por ejemplo:

```
(begin (beep-open)
       (beep 440 1000)
       (beep-close))
```

provoca la emisión de la nota La (440Hz) durante 1 segundo.

Ahora, queremos definir un tipo de dato abstracto para crear melodías y tocarlas en el beeper. Las funciones asociadas son:

- `(start m)`: comienza una melodía `m`, haciendo el trabajo de inicialización del beeper que luego permitirá tocar la melodía
- `(end)`: termina una melodía, cerrando el beeper
- `(note freq time m)`: retorna una nueva melodía que consiste de la nota (frecuencia `freq`, duración `time`) seguida por la melodía `m`.
- `(play m)`: toca la melodía `m`

Por ejemplo:

```
(play (start
      (note 440 500
           (note 523 1000
                (end))))))
```

toca la melodía La-Do (La negra, 1/2s, Do blanca, 1s). Como pueden ver, toda melodía debe empezar con `start` y terminar con `end` de modo de inicializar/cerrar el beeper como corresponde.

- (0.75) Defina una implementación del tipo de dato abstracto anterior usando `define-type`:

```
(define-type Melody ...)
(define (start m) ...)
(define (end) ...)
(define (note f t m) ...)
(define (play m) ...)
```

- (0.75) Ahora, defina una *representación procedural* de las melodías.
- (0.5) En el caso de la representación procedural, explique por qué el scope estático juega un rol central en el buen funcionamiento de sus melodías. Ilustre explicando un caso concreto dónde ocurriría un error si el lenguaje tuviese scope dinámico.

3. Recursión por la cola.

- a) Recuerde la definición del `interp` del FAE con sustitución:

```
;; interp : FAE -> FAE (only num or fun)
(define (interp expr)
  (type-case FAE expr
    (num (n) expr)
    (add (l r) (num+ (interp l) (interp r)))
    (id (v) (error "free identifier"))
    (fun (bound-id bound-body) expr)
    (app (fun-expr arg-expr)
      (let ((fun-val (interp fun-expr))
            (interp (subst (fun-body fun-val)
                          (fun-param fun-val)
                          (interp arg-expr)))))))
```

(0.5) Identifique los llamados recursivos a `interp` e indique, para cada uno de ellos, si está en posición de cola o no.

- b) (0.5) ¿En que consiste la optimización de llamados por la cola? ¿Está implementada en Java? ¿en Scheme?
- c) En el RCFAE, podemos definir funciones recursivas, por ejemplo:

```
{rec {even {fun {n}
          {cond {{= n 0} #t}
                {{= n 1} #f}
                {else {not {even {- n 1}}}}}
      {even 10}}}
```

- (0.5) ¿Es esta función recursiva por la cola? Si no, reescribala para que lo sea. Si sí, explique por qué.
- (0.5) ¿Que puede decir del desempeño de esa función en nuestro intérprete RCFAE, y por qué?