Profesor: Éric Tanter
Auxiliares: Rodrigo Urrea Tomás Vallejos
Ayudantes: R. Iturrieta P. Jaramillo A. Restrepo T. Rivas

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

Auxiliar 5
Repaso C1

P1. *Preguntas conceptuales.*
Considere el siguiente programa:

```
(let ([x 0])
    (let ([f (\lambda () x)])
        (let ([x 42]) (f))))
```

1. ¿Cuál es el resultado del programa? Argumente
2. Si Racket tuviera Scope dinámico por defecto, ¿Cuál seria el resultado del programa anterior?
3. En un lenguaje con evaluación temprana, ¿Un *if* puede ser función?

P2. *¿Cómo interpreto esto?*

```
(define (interp expr env)
    (match expr
        [(num n) (numV n)]
        [(fun id body) (closureV id body empty-env)]
        [(add l r) (num+ (interp l env) (interp r env))]
        [(sub l r) (num- (interp l env) (interp r env))]
        [(if0 c t f ) (if (num-zero? c)
                          (interp t env)
                          (interp f env))]
        [(id x) (env-lookup x env)]
        [(app fun-expr arg-expr)
            (def (closureV id body fenv) (interp fun-expr env))
            (interp body
                (extend-env id
                    (interp arg-expr env)
                fenv))]))
```

(a) Qué problemas tiene este intérprete?
(b) Piense en una expresión que exponga el error
(c) Qué cambios habría que hacer para corregir el error?

P3. *Clausuras.*

(a) Escriba un caso que muestre que el interprete visto en clases con *with* y *lambdas* puede capturar demasiados identificadores en las clausuras.
(b) ¿Puede ser problemático capturar demasiados identificadores en una clausura? En teoría? y en la práctica ?
(c) Comente qué cambios hay que hacer para arreglar el interprete para asegurar que las clausuras solamente capturen los identificadores necesarios.