

Corso di Programmazione

Esame del 3 Febbraio 2017

cognome e nome

Risolvi i seguenti esercizi, riporta le soluzioni in modo chiaro negli appositi spazi e giustifica sinteticamente le risposte. Dovrai poi consegnare queste schede con le soluzioni, avendo cura di scrivere il tuo nome nell'intestazione e su ciascun eventuale foglio aggiuntivo.

1. Programmi in Scheme

```
(define f      ; intero
  (lambda (s)  ; lista di interi positivi
    (g s 0)
  ))
```

```
(define g
  (lambda (s b)
    (if (null? s)
        0
        (if (<= (car s) b)
            (g (cdr s) b)
            (max (g (cdr s) b)
                  (+ (g (cdr s) (car s)) 1)))
    )))
```

In relazione al programma riportato sopra, determina il risultato della valutazione delle seguenti espressioni:

(f '(5))	→	_____	(f '(5 3))	→	_____
(f '(1 2))	→	_____	(f '(5 3 1))	→	_____
(f '(1 3 5))	→	_____	(f '(1 5 3))	→	_____
(f '(1 4 16))	→	_____	(f '(4 2 3 1 5))	→	_____

2. Classi in Java

Definisci una classe in Java per modellare una piccola rubrica telefonica, analoga a quelle disponibili nei più semplici telefonini mobili. Il protocollo della classe deve prevedere un costruttore (che crea una rubrica vuota) e tre metodi: `addItem`, per aggiungere alla rubrica un nome e il corrispondente numero telefonico; `delItem`, per cancellare dalla rubrica il nome passato come argomento e il corrispondente numero telefonico; `phoneNumber`, per reperire il numero telefonico associato a un dato nome. Sia i nomi che i numeri telefonici sono rappresentati da stringhe.

3. Astrazione sui dati e dati procedurali

Si vuole introdurre in Scheme un dato astratto “*crivello di Eratostene*” per realizzare un modello del crivello utilizzato nell’algoritmo di Eratostene per determinare tutti i numeri primi che non superano un dato limite n . Più specificamente, il protocollo del dato astratto è definito dalle seguenti procedure:

```
(new-sieve)                ; crea un crivello con potenzialmente tutti gli interi dispari > 2
(is-in-sieve p sieve)      ; verifica se p è (ancora) presente nel crivello sieve
(remove-multiples-of p sieve) ; crivello con il contenuto di sieve tranne i multipli propri di p
(sieve-list sieve n)       ; restituisce la lista dei numeri  $\leq n$  contenuti nel crivello sieve
```

Applicando questa struttura, l’algoritmo per generare i numeri primi non superiori a n può essere formulato così:

```
(define primes-list
  (lambda (n) (cons 2 (eratosthenes 3 n (new-sieve))) ))
(define eratosthenes
  (lambda (p n s)
    (cond ((> p n) (sieve-list s n))
          ((is-in-sieve? p s) (eratosthenes (+ p 2) n (remove-multiples-of p s)))
          (else (eratosthenes (+ p 2) n s))
    )))
```

Completa la realizzazione del protocollo in modo tale che siano rispettate le specifiche illustrate sopra.

```
(define new-sieve                ; val: crivello di Eratostene (nuovo)
  (lambda ()                    ; "costruttore" senza argomenti
    (lambda (x) (not (even? x))) ; rappresentazione interna procedurale
  ))

(define is-in-sieve?            ; val: booleano
  (lambda (p sieve)            ; p: intero, sieve: crivello di Eratostene
    (sieve p)
  ))

(define remove-multiples-of      ; val: crivello di Eratostene
  (lambda (p sieve)              ; p: intero > 2, sieve: crivello di Eratostene
    (lambda (x)
      (if ( ..... )
          (= x p)
          .....
      )
    )
  ))

(define sieve-list               ; val: lista di interi
  (lambda (sieve n)              ; sieve: crivello di Eratostene, n: intero
    (scan-sieve 3 n sieve)
  ))

(define scan-sieve
  (lambda (x y s)
    (cond ((> x y) ..... )
          .....
          .....
          .....
    ))
  ))
```

4. Verifica formale della correttezza

In relazione alla procedura g definita nell'esercizio 1 si può dimostrare per induzione sul parametro k che

$$(\mathfrak{g} \ (n+1 \ n+2 \ \dots \ n+k) \ q) \quad \rightarrow \quad k$$

per qualsiasi terna di interi n, k, q tali che $k > 0$ e $n \geq q \geq 0$. (Il primo argomento denota una lista di interi.)

Completa l'impostazione della dimostrazione e dimostra la proprietà formulata attenendoti allo schema seguente:

- Formalizza la proprietà che esprime il caso / i casi base:
- L'ipotesi induttiva è la seguente: preso come riferimento $h \in \mathbb{N}^+$
 si assume che $\forall n, q \in \mathbb{N}$ tali che $n \geq q$. $(\exists (n+1 \ n+2 \dots n+h) \ q) \rightarrow h$
- Formalizza la proprietà da dimostrare come passo induttivo:
- Dimostra il caso / i casi base:
- Dimostra il passo induttivo:

5. Memoization

Il programma in Java riportato in questa pagina è sostanzialmente equivalente a quello dell'esercizio 1, per una stringa di caratteri anziché una lista di interi. Applica la tecnica *top-down* di *memoization* al fine di realizzarne uno più efficiente.

```
public static int f( String s ) {  
    return g( s, ' ' );  
}
```

```
public static int g( String s, char b ) {  
    if ( s.equals("") ) {  
        return 0;  
    } else {  
        char c = s.charAt( 0 );  
        String t = s.substring( 1 );  
        if ( c <= b ) {  
            return g( t, b );  
        } else {  
            return Math.max( g(t,b), g(t,c)+1 );  
        }  
    }  
}
```