

Corso di Programmazione

I Accertamento del 14 Dicembre 2007 / A

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 che si renda necessario.

1. Procedure in Scheme

Con riferimento alla procedura *h* così definita:

```
(define h
  (lambda (x)
    (let ((n (string-length x)))
      (if (<= n 2)
          x
          (h (string-append (substring x 3 n) (substring x 0 2)))))))
```

calcola i risultati della valutazione di ciascuna delle seguenti espressioni Scheme:

(h "AB")	→	<u> "AB" </u>	(h "ABC")	→	<u> "AB" </u>
(h "ABCD")	→	<u> "DA" </u>	(h "ABCDE")	→	<u> "BD" </u>
(h "ABCDEF")	→	<u> "EA" </u>	(h "ABCDEFGH")	→	<u> "DG" </u>

2. Definizione di procedure in Scheme

Definisci una procedura *week-day* in Scheme che, dato il “giorno della settimana” del *primo* del mese e dato un numero *n* compreso fra 1 e 31, restituisce il “giorno della settimana” corrispondente al giorno *n* dello stesso mese. I giorni della settimana sono rappresentati da stringhe con le prime tre lettere: “lun”, “mar”, “mer”, “gio”, “ven”, “sab”, “dom”. Detto altrimenti, la procedura permette di rispondere a domande come la seguente: se il primo di marzo cade di giovedì, che giorno della settimana sarà il 23 marzo? La risposta è: (*week-day* “gio” 23).

```
(define week-day
  (lambda (1st day)
    (idf (remainder (+ day (pos 1st) -1) 7))
  ))

(define idf
  (lambda (k)
    (cond ((= k 0) "lun")
          ((= k 1) "mar")
          ((= k 2) "mer")
          ((= k 3) "gio")
          ((= k 4) "ven")
          ((= k 5) "sab")
          (else "dom")
    ))))

(define pos
  (lambda (d)
    (cond ((string=? d "lun") 0)
          ((string=? d "mar") 1)
          ((string=? d "mer") 2)
          ((string=? d "gio") 3)
          ((string=? d "ven") 4)
          ((string=? d "sab") 5)
          (else 6)
    )))
```

3. Definizione di procedure in Scheme

Valori numerici nell'intervallo $[0,1)$ possono essere rappresentati in forma binaria da una stringa di cifre "0" e "1" precedute dal carattere "." (punto), dove i singoli bit sono pesati da potenze negative di due. Per esempio, le stringhe ".1" e ".011" corrispondono ai numeri 0.5 e 0.375, rispettivamente, nella consueta notazione in base dieci. Definisci una procedura *r-val* in Scheme per determinare il valore numerico di stringhe del tipo descritto sopra (punto seguito da una o più cifre binarie).

```
(define r-val
  (lambda (n)
    (cond ((= (string-length n) 0) 0)
          ((char=? (string-ref n 0) #\1) (+ 1 (* 0.5 (val (substring n 1)))))
          (else (* 0.5 (val (substring n 1)))))
    )))
```

4. Procedure con argomenti e valori procedurali

Data una funzione f e un naturale i , la procedura *iter* restituisce l' i -ima iterata di f . La procedura *step* è invece in relazione al programma dell'esercizio 1.

```
(define iter
  (lambda (f i)
    (if (= i 0)
        (lambda (x) x)
        (lambda (x) (f ((iter f (- i 1)) x))))))

(define step
  (lambda (x)
    (string-append
      (substring x 3)
      (substring x 0 2))))
```

Completa la definizione della procedura *h2*, che utilizza *iter* e *step*, in modo che *h2* risulti equivalente alla procedura *h* dell'esercizio 1, cioè in modo che le espressioni $(h\ s)$ e $(h2\ s)$ assumano lo stesso valore per ogni stringa s .

```
(define h2
  (lambda (x)
    (
      (iter step _____)
      _____
    )
  ))
```

5. Verifica formale della correttezza

```

(define f
  (lambda (n)
    (f-tr n 0 1 6)
  ))

(define f-tr
  (lambda (k x y z)
    (if (= k 0)
        x
        (f-tr (- k 1) (+ x y) (+ y z) (+ z 6))
    )))

```

In relazione al programma in Scheme sopra riportato è possibile dimostrare per induzione che:

$$\forall k, x, y, z \in \mathbb{N}. (f\text{-tr } k \ x \ y \ z) \rightarrow k^3 + k(k-1)(z-6)/2 + k(y-1) + x \quad (*)$$

Imposta i passi della dimostrazione per induzione e determina i valori calcolati dalla procedura f , in particolare:

- Formalizza la proprietà che esprime i casi base (senza dimostrazione):

$$\forall x, y, z \in \mathbb{N}. (f\text{-tr } 0 \ x \ y \ z) \rightarrow 0^3 + 0(0-1)(z-6)/2 + 0(y-1) + x$$

- Formalizza l'ipotesi induttiva: fissato $q \in \mathbb{N}$

$$\forall x, y, z \in \mathbb{N}. (f\text{-tr } q \ x \ y \ z) \rightarrow q^3 + q(q-1)(z-6)/2 + q(y-1) + x$$

- Formalizza la proprietà da dimostrare come passo induttivo (senza svolgere i calcoli): per q fissato sopra

$$\forall x, y, z \in \mathbb{N}. (f\text{-tr } q+1 \ x \ y \ z) \rightarrow (q+1)^3 + (q+1)q(z-6)/2 + (q+1)(y-1) + x$$

- Scrivi un'espressione che rappresenti i valori assunti dalla procedura f per $n \in \mathbb{N}$:

$$\forall n \in \mathbb{N}. (f\ n) \rightarrow n^3$$

- Dimostra la correttezza della procedura f sulla base della proprietà (*).

$$(f\ n) \rightarrow (f\text{-tr } n \ 0 \ 1 \ 6) \rightarrow n^3 + n(n-1)(6-6)/2 + n(1-1) + 0 = n^3$$

6. Procedure con argomenti procedurali

Completa la definizione della procedura $f2$ in modo che $f2$ risulti equivalente alla procedura f dell'esercizio 5, cioè che le due espressioni $(f\ v)$ e $(f2\ v)$ assumano lo stesso valore per ogni naturale v .

```
(define f2
  (lambda (n)
    (f2-tr n 0 1 6
      +
      (lambda (x) (+ x 6))
    ))

(define f2-tr
  (lambda (k x y z p q)
    (if (= k 0)
      x
      (f2-tr (- k 1) (p x y) (p y z) (q z) p q)
    )))
```

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 che si renda necessario.

1. Procedure in Scheme

Con riferimento alla procedura *h* così definita:

```
(define h
  (lambda (x)
    (let ((n (string-length x)))
      (if (> n 1)
          (h (string-append (substring x 2 n) (substring x 0 1)))
          x)
      ))))
```

calcola i risultati della valutazione di ciascuna delle seguenti espressioni Scheme:

(h "A")	→	<u>"A"</u>	(h "AB")	→	<u>"A"</u>
(h "ABC")	→	<u>"C"</u>	(h "ABCD")	→	<u>"A"</u>
(h "ABCDEFG")	→	<u>"G"</u>	(h "ABCDEFGH")	→	<u>"A"</u>

2. Definizione di procedure in Scheme

Definisci una procedura *week-day* in Scheme che dato un numero *n* compreso fra 1 e 31 e dato il “giorno della settimana” del *primo* del mese, restituisce il “giorno della settimana” corrispondente al giorno *n* dello stesso mese. I giorni della settimana sono rappresentati da stringhe con le prime tre lettere della dizione inglese: “*mon*”, “*tue*”, “*wed*”, “*thu*”, “*fri*”, “*sat*”, “*sun*”. Detto altrimenti, la procedura permette di rispondere a domande come la seguente: se il primo di aprile cade di martedì, che giorno della settimana sarà il 26 aprile? La risposta è: (*week-day* 26 “*tue*”).

```
(define week-day
  (lambda (day 1st)
    (idf (remainder (+ day (pos 1st) -1) 7))
  ))

(define idf
  (lambda (k)
    (cond ((= k 0) "mon")
          ((= k 1) "tue")
          ((= k 2) "wed")
          ((= k 3) "thu")
          ((= k 4) "fri")
          ((= k 5) "sat")
          (else "sun")
    ))))

(define pos
  (lambda (d)
    (cond ((string=? d "mon") 0)
          ((string=? d "tue") 1)
          ((string=? d "wed") 2)
          ((string=? d "thu") 3)
          ((string=? d "fri") 4)
          ((string=? d "sat") 5)
          (else 6)
    )))
```

3. Definizione di procedure in Scheme

Valori numerici nell'intervallo $[0,1)$ possono essere rappresentati in forma binaria da una stringa di cifre "0" e "1" precedute dal carattere "." (punto), dove i singoli bit sono pesati da potenze negative di due. Per esempio, le stringhe ".1" e ".011" corrispondono ai numeri 0.5 e 0.375, rispettivamente, nella consueta notazione in base dieci.

Definisci una procedura *q-val* in Scheme per determinare il valore numerico di stringhe del tipo descritto sopra (punto seguito da una o più cifre binarie).

```
(define q-val
  (lambda (n)
    (cond ((= (string-length n) 0) 0)
          ((char=? (string-ref n 0) #\1) (+ 1 (* 0.5 (val (substring n 1)))))
          (else (* 0.5 (val (substring n 1)))))
    )))
```

4. Procedure con argomenti e valori procedurali

Data una funzione f e un naturale i , la procedura *iter* restituisce l' i -ima iterata di f . La procedura *step* è invece in relazione al programma dell'esercizio 1.

```
(define iter
  (lambda (f i)
    (if (= i 0)
        (lambda (x) x)
        (lambda (x) (f ((iter f (- i 1)) x)))))

(define step
  (lambda (x)
    (string-append
      (substring x 2)
      (substring x 0 1))
    )))
```

Completa la definizione della procedura *h2*, che utilizza *iter* e *step*, in modo che *h2* risulti equivalente alla procedura *h* dell'esercizio 1, cioè in modo che le espressioni $(h\ s)$ e $(h2\ s)$ assumano lo stesso valore per ogni stringa s .

```
(define h2
  (lambda (x)
    (
      (iter step _____)
      _____
    )
  ))
```

5. Verifica formale della correttezza

```
(define f
  (lambda (n)
    (f-tr n 6 1 0)
  ))
```

```
(define f-tr
  (lambda (u v x y)
    (if (= u 0)
        y
        (f-tr (- u 1) (+ v 6) (+ v x) (+ x y))
    )))
```

In relazione al programma in Scheme sopra riportato è possibile dimostrare per induzione che:

$$\forall u, v, x, y \in \mathbb{N}. (f\text{-tr } u \ v \ x \ y) \rightarrow u^3 + u(u-1)(v-6)/2 + u(x-1) + y \quad (*)$$

Imposta i passi della dimostrazione per induzione e determina i valori calcolati dalla procedura f , in particolare:

- Formalizza la proprietà che esprime i casi base (senza dimostrazione):

$$\forall v, x, y \in \mathbb{N}. (f\text{-tr } 0 \ v \ x \ y) \rightarrow 0^3 + 0(0-1)(v-6)/2 + 0(x-1) + y$$

- Formalizza l'ipotesi induttiva: fissato $q \in \mathbb{N}$

$$\forall v, x, y \in \mathbb{N}. (f\text{-tr } q \ v \ x \ y) \rightarrow q^3 + q(q-1)(v-6)/2 + q(x-1) + y$$

- Formalizza la proprietà da dimostrare come passo induttivo (senza svolgere i calcoli): per q fissato sopra

$$\forall v, x, y \in \mathbb{N}. (f\text{-tr } q+1 \ v \ x \ y) \rightarrow (q+1)^3 + (q+1)q(v-6)/2 + (q+1)(x-1) + y$$

- Scrivi un'espressione che rappresenti i valori assunti dalla procedura f per $n \in \mathbb{N}$:

$$\forall n \in \mathbb{N}. (f\ n) \rightarrow n^3$$

- Dimostra la correttezza della procedura f sulla base della proprietà (*).

$$(f\ n) \rightarrow (f\text{-tr } n \ 6 \ 1 \ 0) \rightarrow n^3 + n(n-1)(6-6)/2 + n(1-1) + 0 = n^3$$

6. Procedure con argomenti procedurali

Completa la definizione della procedura $f2$ in modo che $f2$ risulti equivalente alla procedura f dell'esercizio 5, cioè che le due espressioni $(f\ z)$ e $(f2\ z)$ assumano lo stesso valore per ogni naturale z .

```
(define f2
  (lambda (n)
    (f2-tr n 6 1 0
      _____
      +
      _____ )
    ))

(define f2-tr
  (lambda (u v x y p q)
    (if (= u 0)
        y
        (f2-tr (- u 1) (p v) (q v x) (q x y) p q)
    )))
```