Esercitazione 8

Istruzioni trattate: `number_properties`, `msprintf`.

Nella prima parte di questa esercitazione completeremo la presentazione della funzione predefinita `number_properties` e mostreremo come l'insieme dei numeri di macchina di *Scilab* sia fedelmente descritto da un insieme di numeri in virgola mobile con esponente limitato ed elementi denormalizzati. Nella seconda parte descriveremo come ottenere da *Scilab* la scrittura posizionale in base dieci di un numero di macchina.

Prima parte

Dati β numero intero maggiore di uno, m numero intero positivo, b_{min} e b_{max} numeri interi tali che $b_{min} < b_{max}$, indichiamo con:

$$F(\beta, m, b_{min}, b_{max})$$

l'insieme di tutti i numeri reali x tali che:

$$x = (-1)^s \beta^b \, 0.c_1 \cdots c_m$$

con $s \in \{0, 1\}$, b numero intero tale che $b_{min} \leqslant b \leqslant b_{max}$ e $c_1, \ldots, c_m$ cifre in base β *aritrarie*. Gli elementi dell'insieme che si ottengono scegliendo $c_1 \neq 0$ si dicono *normalizzati*. Per questi numeri b è l'esponente in base β e $0.c_1 \cdots c_m$ è la scrittura posizionale in base β della frazione. Gli elementi non nulli che si ottengono scegliendo $b = b_{min}$ e $c_1 = 0$ di dicono *denormalizzati*. Se x è un elemento denormalizzato e $c_1 = \cdots = c_k = 0, c_{k+1} \neq 0$ allora l'esponente in base β di x è $b_{min} - k$ e la scrittura posizionale in base β della frazione è $0.c_{k+1} \cdots c_m$. $F(\beta, m, b_{min}, b_{max})$ è un insieme di *numeri in virgola mobile con esponente limitato ed elementi denormalizzati*.

L'insieme $F(\beta, m, b_{min}, b_{max})$, al pari di $F(\beta, m)$, *è simmetrico rispetto a zero*. Inoltre, indicando con F_n^-, F_n^+, F_d^- e F_d^+, rispettivamente, gli elementi negativi normalizzati, positivi normalizzati, negativi denormalizzati e positivi denormalizzati di $F(\beta, m, b_{min}, b_{max})$ si ha:

* $F_n^- = -F_n^+$ e $F_d^- = -F_d^+$;
* $F(\beta, m, b_{min}, b_{max}) = F_n^- \cup F_d^- \cup \{0\} \cup F_d^+ \cup F_n^+$;
* Se $\xi_n^- \in F_n^-, \xi_d^- \in F_d^-, \xi_d^+ \in F_d^+$ e $\xi_n^+ \in F_n^+$ allora $\xi_n^- < \xi_d^- < 0 < \xi_d^+ < \xi_n^+$.

Contrariamente a $F(\beta, m)$ l'insieme $F(\beta, m, b_{min}, b_{max})$ ha un numero *finito* di elementi. Quindi:

* *è superiormente limitato* e:

$$\max F(\beta, m, b_{min}, b_{max}) = \max F_n^+ = \beta^{b_{max}} (1 - \beta^{-m}) = \xi_{max}$$

* *zero non è punto di accumulazione* e:

$$\min\{\xi \in F(\beta, m, b_{min}, b_{max}) : \xi > 0\} = \min F_d^+ = \beta^{b_{min} - m} = \xi_{min}$$

Inoltre, estendendo le definizioni di *predecessore, successore* e di *funzione arrotondamento*:

* $\sigma(0) = \xi_{min}$
* $\mathrm{rd}(x) = 0 \not\Rightarrow x = 0$, ad esempio:

$$\mathrm{rd}(\xi_{min}/2) = 0$$

Utilizzando la funzione predefinita `number_properties` è possibile constatare che l'insieme dei numeri di macchina di *Scilab* è $F(2, 53, -1021, 1024)$. Infatti:

```
number_properties('minexp')
```

restituisce *l'esponente minimo* b_{min},

```
number_properties('maxexp')
```

restituisce *l'esponente massimo* b_{max},

`number_properties('huge')`

restituisce *il più grande numero di macchina* ξ_{max},

`number_properties('denorm')`

dichiara se l'insieme dei numeri di macchina contiene elementi *denormalizzati*,

`number_properties('tiny')`

restituisce *il più piccolo numero di macchina positivo normalizzato* $\min F_n^+$,

`number_properties('tiniest')`

restituisce *il più piccolo numero di macchina positivo* ξ_{min} e si ottiene:

```
-->b_min = number_properties('minexp')
 b_min  =

  - 1021.

-->b_max = number_properties('maxexp')
 b_max  =

    1024.

-->xi_max = number_properties('huge')
 xi_max  =

    1.79D+308

-->denormalizzati = number_properties('denorm')
 denormalizzati  =

  T

-->min_FN = number_properties('tiny')
 min_FN  =

    2.22D-308

-->xi_min = number_properties('tiniest')
 xi_min  =

    4.94D-324
```

Coerentemente con quanto detto prima si ha:

```
-->m = number_properties('digits')
 m  =

    53.

-->xi_min == 2^(bmin - m)
 ans  =

  T

-->s0 = nearfloat('succ',0)
```

```
 s0  =

    4.94D-324

-->s0 == xi_min
 ans  =

  T

-->xi_min / 2
 ans  =

    0.

-->ans == 0
 ans  =

  T
```

Seconda parte

Il teorema che segue consente di utilizzare *Scilab* per ottenere la scrittura posizionale in base dieci di un numero di macchina.

- *Teorema* (scrittura posizionale in base dieci di un elemento non nullo di $F(2,m)$)

  Sia $\xi = 2^b\, 0.c_1 \cdots c_k$ con b numero intero, $c_1, \ldots, c_k$ cifre in base due e $c_1 \neq 0, c_k \neq 0$. Si ha:

  (1) *ξ è un numero intero se e solo se $b \geqslant k$.*

  (2) *Se $b < k$ allora $k - b > 0$ e la scrittura posizionale in base dieci di ξ ha $k - b$ cifre dopo il punto decimale*, ovvero: l'ultima cifra non nulla dopo il punto decimale è la $(k - b)$-esima.

  Dimostrazione: (1) Se $b \geqslant k$ allora $b - k \geqslant 0$ e $\xi = 2^{b-k}\, c_1 \cdots c_k$ è un numero intero. Altrimenti non lo è perché, essendo $c_k \neq 0$, $c_1 \cdots c_k$ non è divisibile per 2^{k-b}. (2) Se $b < k$ allora $k - b > 0$ e:

  (i) $10^{k-b}\xi = 5^{k-b}\, 2^{k-b}\, 2^{b-k} c_1 \cdots c_k = 5^{k-b}\, c_1 \cdots c_k$ è un numero intero.

  (ii) $10^{k-b-1}\xi = 5^{k-b-1}\, 2^{k-b-1}\, 2^{b-k}\, c_1 \cdots c_k = 5^{k-b-1}\, 2^{-1}\, c_1 \cdots c_k$ *non* è un numero intero perché $k - b - 1 \geqslant 0$ e, essendo $c_k \neq 0$, $c_1 \cdots c_k$ non è divisibile per due.

  L'asserto (i) mostra che la scrittura posizionale in base dieci di ξ ha *al più* $k - b$ cifre. L'asserto (ii) mostra che la $(k - b)$-esima cifra dopo il punto decimale è *diversa da zero.*

Per manipolare la scrittura posizionale in base dieci di un numero di macchina utilizzeremo la *funzione predefinita* `msprintf` di *Scilab*:

- `msprintf`

  Questa *funzione predefinita* restituisce *una stringa contenente il valore di una o più espressioni.* La stringa è costurita con lo stesso meccanismo utilizzato dalla *funzione predefinita* `printf` (vedere l'Esercitazione 3). Ad esempio:[1]

  ```
  -->s = msprintf('il valore di log(%d) è circa %.5f',2,log(2))
   s  =

   il valore di log(2) è circa 0.69315

  -->typeof(s)
   ans  =

   string
  ```

[1]La *funzione predefinita* `typeof` restituisce il *tipo* di una variabile.

Siano, allora, ξ un numero di macchina non nullo di *esponente* b e c_k l'ultima cifra non nulla della scrittura posizionale in base due della *frazione* di ξ. Si ha:

- Se $b \geqslant k$, ξ è un numero intero e la stringa che rappresenta la sua scrittura posizionale in base dieci si ottiene con:

  $$\texttt{msprintf('\%.0f', }\ \xi\texttt{)}$$

- Se $b < k$, ξ è un numero non intero e la stringa che rappresenta la sua scrittura posizionale in base dieci si ottiene, posto $n = k - b > 0$, con:

  $$\texttt{msprintf('\%.}n\texttt{', }\ \xi\texttt{)}$$

Ad esempio:

∗ Per $\xi = \xi_{max}$ si ha: $b = 1024$ e $k = 53$. Dunque ξ_{max} è un numero intero. Poiché:

```
-->xi_max
 xi_max  =

    1.79D+308
```

ci si aspetta che la scrittura posizionale in base dieci di ξ_{max} sia una stringa di 309 cifre, risultato confermato dal seguente dialogo:

```
-->s = msprintf('%.0f',xi_max)
 s  =

 1797693134862315708145274237317043567980705675258449965989174768031572607800
      28538760589558632766878171540458953514382464234321326889464182768467546
      70353751698604991057655128207624549009038932894407586850845513394230458
      32369032229481658085593321233482747978262041447231687381771809192998812
      50404026184124858368

-->length(s)
 ans  =

    309.
```

∗ Per $\xi = \xi_{min} = 2^{-1021}\,0.0\cdots01 = 2^{-1073}\,0.1$ si ha: $b = -1073$ e $k = 1$. Dunque $b < k$ e $k - b = 1074$: la scrittura posizionale in base dieci di ξ_{min} ha 1074 cifre dopo il punto decimale. Poiché inoltre:

```
-->xi_min
 xi_min  =

    4.94D-324
```

nella scrittura posizionale in base dieci le prime 323 cifre dopo il punto decimale sono uguali a zero. Infatti:[2]

```
-->s = msprintf('%.1074f',xi_min);

-->part(s,1:326)
 ans  =

 0.00000000000000000000000000000000000000000000000000000000000000000000000000
      00000000000000000000000000000000000000000000000000000000000000000000000
      00000000000000000000000000000000000000000000000000000000000000000000000
      00000000000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000004
```

[2]Se $s = s_1 \cdots s_n$ è una stringa e i, j due numeri interi tali che $1 \leqslant i < j \leqslant n$, $\texttt{part(}s\texttt{,}i\texttt{:}j\texttt{)}$ è la stringa $s_i \cdots s_j$.

* Per $\xi = u = 2^{-53} = 2^{-52}\,0.1$ si ha: $b = -52$ e $k = 1$. Dunque $b < k$ e $k - b = 53$: la scrittura posizionale in base dieci di u ha 53 cifre dopo il punto decimale. Poiché inoltre:

```
-->u = 2^(-53)
 u  =

    1.110D-16
```

nella scrittura in base dieci le prime 15 cifre dopo il punto decimale sono zero. Infatti:

```
-->s = msprintf('%.53f',u)
 s  =

 0.00000000000000011102230246251565404236316680908203125

-->part(s,1:18)
 ans  =

 0.0000000000000001
```

Esercizi

---

1. Discutere il seguente dialogo con *Scilab*:

```
-->xi = nearfloat('pred',min_FN)
 xi  =

    2.22D-308

-->xi == (1 - 2^(-m+1)) * 2^(bmin - 1)
 ans  =

  T
```

2. Determinare esponente e frazione in base due di `xi_min` e `min_FN` e verificare il risultato utilizzando la funzione predefinita `frexp`.
3. Sia $\xi = u^2$. Determinare la scrittura posizionale in base dieci di ξ.
4. Sia `%pi` l'arrotondato di π in $F(2, 53)$. Si ha:

$$\texttt{\%pi} = 2^2 \cdot 0.11001001000011111101101010100010001000010110100011000$$

Determinare la scrittura posizionale in base dieci di `%pi`.