Laboratorio di Programmazione a.a. 2017-18

Problema 3

8 / 9 Novembre 2017

Descrizione

L'obiettivo che ci proponiamo è di riuscire a determinare il valore di rappresentazioni numeriche in una data base, composte da segno, parte intera e parte frazionaria. Questo compito è svolto sistematicamente, per esempio, dai programmi che acquisiscono le informazioni numeriche digitate attraverso la tastiera, normalmente espresse in base 10 perché ciò corrisponde alle nostre abitudini.

Cominciamo dalla notazione in base 2 (binaria). Il dato è una stringa composta esclusivamente dai caratteri `0` (cifra zero), `1` (cifra uno), `+` (segno più), `-` (segno meno) e `.` (punto); il risultato che vogliamo determinare è il numero razionale rappresentato dalla stringa. Tale risultato ci verrà comunicato in base 10, ma occorre distinguere stringhe e numeri perché sono dati di tipo diverso.

Definisci in Scheme una procedura bin-`rep->number` che, data una stringa in notazione binaria, restituisce il valore razionale che questa rappresenta. A tal fine converrà organizzare il programma in procedure specializzate in compiti specifici, in particolare: la determinazione del segno, il calcolo della parte intera e della parte frazionaria (o, in alternativa, il calcolo del numero intero rappresentato da tutte le cifre, a cui si applica un opportuno riscalamento).

Esempi

```
(bin-rep->number "+1101")  ⟶  13

(bin-rep->number "0")  ⟶  0

(bin-rep->number "10110.011")  ⟶  22.375

(bin-rep->number "-0.1101001")  ⟶  -0.8203125
```

La tecnica può essere estesa facilmente a qualsiasi base. Per rappresentare un numero in base *b*, oltre all'eventuale segno e al punto che separa la parte intera dalla parte frazionaria, sono necessarie *b* cifre di valore 0, 1, ... *b–1*, cifre che sono convenzionali e pertanto possono essere rappresentate da qualsiasi insieme di *b* simboli.

Definisci in Scheme una procedura `rep->number` che, date due stringhe, la prima contenente una sequenza di *b* cifre ordinate per valore crescente e la seconda una rappresentazione numerica in base *b* che utilizza le cifre convenute (e può eventualmente comprendere anche un segno e/o un punto per separare la parte frazionaria), restituisce il valore razionale rappresentato dalla stringa.

Esempi

```
(rep->number "zu" "-uuzz")  ⟶  -12

(rep->number "0123" "+21.1")  ⟶  9.25

(rep->number "01234" "-10.02")  ⟶  -5.08

(rep->number "0123456789ABCDEF" "0.A")  ⟶  0.625

(rep->number "0123456789ABCDEF" "1CF.0")  ⟶  463
```