

Контролна работа 1

Инструкция

Важно: Решете задачите на R5RS Scheme. В решенията не може да се използват списъци, нито деструктивни операции.

Както и в задачите от домашното, множество S от естествени числа ще представяме чрез едно естествено число, което ще наричаме негово представяне. Числото N се съдържа в S тогава и само тогава, когато $N+1$ -вият разряд в двоичния запис на представянето е 1.

Например

Множество Представяне		Коментар
{5, 1, 0}	35	двоичният запис на 35 е 100011(2)
{4, 3, 1}	26	двоичният запис на 26 е 11010(2)
\emptyset	0	двоичният запис на 0 е 0(2)

Навсякъде в задачите, където се изисква да се работи с множества, това трябва да става в така описаното представяне.

Вариант В

Едно естествено число ще наричаме "валидно", ако в неговия запис никъде не се срещат две последователни нули. Например 0, 5, 123, 1230 и 1023012301 са валидни числа. 12003 и 120000000051 не са валидни числа.

Реализирайте следните функции:

А) `(number-valid? n)`, която проверява дали естественото число `n` е валидно.

Б) `(valid->nset n)`, която получава естествено число `n`. Ако то е валидно, функцията извлича от записа му всички числа, които се намират между нулите в записа му и от тях формира множество S , което се връща като резултат. Ако `n` не е валидно число, функцията да връща `#f`.

Примери:

- `(valid->nset 5050123050) →` множеството {5,123}
- `(valid->nset 0) →` празното множество
- `(valid->nset 120034) → #f`

В) (`make-nset a b pred?`), която връща множеството на онези числа от естествения интервал $[a,b]$, за които предикатът `pred?` връща истина.

В решението на тази подточка не можете да използвате рекурсия. Реализирайте `make-nset` с подходящо обръщение към функцията `accumulate`, която е дадена по-долу.

Забележка: `accumulate` трябва да се използва така както е дадена по-долу. Кодът ѝ не бива да се променя.

```
(define (accumulate op term init a next b)
  (define (loop i)
    (if (<= i b)
        (op (term i) (loop (next i)) )
        init)
  )
  (loop a)
)
```