

1. Exercițiul logică din predicatul F este dat de următoarea listă: S_1 is $S - H$. În cazul dat, S_1 este calculat, este o var. legată, însa S este un program. de fapt și nu are valoare. Dacă am fi vrut să calculăm $(-1) \cdot$ suma tuturor nr. din listă, atunci lista respectivă ar trebui să fie înlocuită cu aceasta: S is $S_1 - 1$.

5. Nr. de subeiste pt. care vrem un atom nu este număr.

Modele matematice.

$$\text{etimmue}(e_1 \dots e_n, \text{cae}) =$$

$$= \begin{cases} \text{cae}, n=0 \text{ (cae} = \text{cae}_1 \dots \text{cae}_n) \\ \text{etimmue}(e_2 \dots e_n, e_1 \oplus \text{cae}), e_1 \text{ e atom} \\ \text{etimmue}(e_2 \dots e_n, (\text{etimmue}(e_1, \text{cae})), \text{ae} + \text{pe} \text{e}) \end{cases}$$

$$\text{subeiste}(e) = \begin{cases} 0, e \text{ e atom} \\ 1 + \sum_{i=1}^n \text{subeiste}(e_i), \text{etimmue}(e, \emptyset) \text{ nu e nr.} \\ \sum_{i=1}^n \text{subeiste}(e_i), \text{ae} + \text{pe} \text{e} \end{cases}$$

(defun etimmue (e cae)

(cond

((null e) (car cae))

((atom (car e)) (etimmue (cdr e) (cons (car e) cae)))

(+ (etimmue (cdr e) (list (etimmue (car e) cae))))

)

)

```
(defun subeiste (e)
```

```
  (cond
```

```
    ((atom e) 0)
```

```
    ((not (numberp (elt e 0)))
```

```
      (+ 1 (apply #' + (mapcar #' subeiste e))))
```

```
      (+ (apply #' + (mapcar #' subeiste e))))
```

```
  )
```

▽ Dacă nu se ia în considerare lista inițială
ca subeiste:

$$\text{main}(l) = \sum_{i=1}^n \text{subeiste}(e_i)$$

```
(defun main (e)
```

```
  (apply #' + (mapcar #' subeiste e))
```