

## Prevodioci i interpretatori - februar 2008.

1. Napisati perl skript koji simulacijom rada konačnog automata u svim datotekama koje se nalaze u direktorijumu koji se navodi kao argument komandne linije pronalazi one linije koje sadrže reč iz regularnog jezika  $((a|b) * ab?) + ba$ .
2. Napisati interpreter za fiktivni jezik MiniLisp. Osnovni izrazi ovog jezika su atomi (literali, numerali) ili liste koje u sebi kao elemente sadrže druge izraze. Prazne liste se označavaju sa (). Specijalna vrednost `nil` označava netačno, dok specijalna vrednost `t` označava tačno.

Svakom izrazu je moguće pridružiti vrednost. Vrednost atomičkog izraza je sam on. Vrednost izraza predstavljenih listama se izračunava tako što se prvi element liste tumači kao oznaka funkcije dok se ostatali elementi liste tumači kao operandi. Tako je npr. vrednost izraza `(+ 2 3)` atom 5. Ukoliko se ispred liste navede ' to označava da nije potrebno na pretnodno opisan način izračunavati vrednost ove liste, već ona doslovno predstavlja sebe.

Jezik podržava korišćenje aritmetičkih operatora `*` i `+`. Operator `car` izdvaja prvi element liste. Operator `cdr` izdvaja ostatak liste. Operator `cons` postavlja dati element na početak date liste, dok operator `append` nadovezuje dve liste. Operator `if` označava uslovni operator. Operator `null` proverava da li je lista prazna, dok operator `eq1` proverava jednakost.

Operator `defun` omogućava definisanje funkcija.

<code>(* 3 (+ 4 5))</code>	27
<code>(car '(1 2 3 4))</code>	1
<code>(cdr '(1 2 3 4))</code>	(2 3 4)
<code>(car (cdr (cdr '(1 2 3 4))))</code>	3
<code>(cons 1 '(2 3))</code>	(1 2 3)
<code>(cons '(1 2) '(3 4))</code>	((1 2) 3 4)
<code>(append '(1 2) '(3 4))</code>	(1 2 3 4)
<code>(append '(1 2 3) '() '(4) '(5 6))</code>	(1 2 3 4 5 6)
<code>(null '(1 2 3))</code>	nil
<code>(null '())</code>	t
<code>(eq1 1 2)</code>	nil
<code>(eq1 1 1)</code>	t
<code>(if (eq1 1 2) 3 4)</code>	4
<code>(defun member (obj lst)</code> <code>(if (null lst)</code> <code>nil</code> <code>(if (eq1 (car lst) obj)</code> <code>lst</code> <code>(member obj (cdr lst))))))</code>	
<code>(member 3 (1 2 3 5))</code>	3 5
<code>(member 3 (1 2 4 5))</code>	nil