

Prevodioci i Interpretatori - Novembar 2010.

praktični deo

1. Konstruisati MDKA za prepoznavanje jezika nad azbukom $\Sigma = \{a, b\}$ čija svaka reč ne sme da sadrži faktor *aaab*, i zatim napisati C program koji proverava da li uneta niska pripada ovom jeziku.
2. *LL(1)* gramatikom opisati *select* upit u SQL-u, a zatim napraviti perl skript koji tabličnom simulacijom potisnog automata proverava da li je na ulazu zadat ispravan upit. Npr:

<code>select srbr</code>	<code>select *</code>	<code>select sime, jrbr</code>	<code>select *</code>
<code>from s</code>	<code>from s</code>	<code>from s, spj</code>	<code>from spj</code>
<code>where grad='Pariz' or status>10</code>		<code>where s.srbr = spj.srbr</code>	<code>where jrbr='j1'</code>

3. Napisati interpreter za rad sa regularnim izrazima nad azbukom $\Sigma = \{a, \dots, z\}$:

- (a) Prepoznati validne regularne izraze: (tačka predstavlja konkatenciju)

`a*.b+.c.d.e{3}|a+.b?.c.d| [abcde] | (a.b)*. [cde]?`

- (b) Regularni izrazi se naredbom BASIC prebacuju u osnovne regularne izraze:

BASIC <code>a*.b+.c.d.e{3}</code>	<code>(((((a*))((b)((b)*))) (c)) (d)) ((e) (e) (e)))</code>
BASIC <code>a* b</code>	<code>((a)* (b))</code>
BASIC <code>[cde]?</code>	<code>((c) (d) (e)) (EPS))</code>

- (c) Omogućiti rad sa regularnim definicijama:

DEFINITION `A = a.b*|b.a* , B = [ab]{2}`
BASIC `A` `((a)((b*)) | ((b)((a)*)))`

- (d) Regularne definicije se prave nad već definisanim regularnim definicijama ili nad regularnim izrazima:

DEFINITION `C = A | c.d*`
BASIC `C` `((((a)((b*)) | ((b)((a)*))) | ((c)((d)*)))`
BASIC `C . B` `(((((a)((b*)) | ((b)((a)*))) | ((c)((d)*))) ((a) | (b)) ((a) | (b))))`
BASIC `(A | c.d*) . B` `(((((a)((b*)) | ((b)((a)*))) | ((c)((d)*))) ((a) | (b)) ((a) | (b))))`

- (e) Za zadati regularni izraz naći dužinu najduže reči koja je opisana njime:

MAXLEN	<code>a*.b{3}. [abc]? . C</code>	<code>+Inf</code>
MAXLEN	<code>a?. [abc]. b{3} abc</code>	<code>5</code>

Srećno!