Proiectarea algoritmilor
Fișă de exerciții
Seminarul 7
Căutarea peste șiruri

Ștefan Ciobâcă, Dorel Lucanu
Universitatea Alexandru Ioan Cuza, Iași

Anul universitar 2019-2020

1. În curs este dată următoarea definiție pentru expresii regulate:

```
expression ::= term ("+" term)*
term ::= maybeStar
       | maybeStar ("." maybeStar)*
maybeStar ::= factor ["*"] /* equiv to  factor | factor "*"  */
factor ::=
    Sigma
    | "(" expression ")"
```

   (a) Să se investigheze dacă următoarea definiție este echivalentă cu cea de mai sus:

```
expression ::= ("+" term)* term
term ::= maybeStar
       | ("." maybeStar)* maybeStar
maybeStar ::= factor ["*"] /* equiv to  factor | factor "*"  */
factor ::=
    Sigma
    | "(" expression ")"
```

   (b) Cum s-ar scrie algoritmii de parsare în acest caz.

   (c) Care dintre forme este mai potrivită?

2. Să se execute algoritmul de parsare (din parser.alk) pentru input-ul "a(b+c)*a" și să se deseneze AST-ul afișat de parser.

3.

   (a) Să se scrie șirurile de lungime cel mult 5 din limbajul specificat de expresia "a(b+c)*a".

   (b) Să se execute algoritmul de generare a limbajului mărginit (din lang.alk) pentru expresia "a.(b+c)*.a" și limita 5.

   (c) Comparați rezultatele.

   Observație. Funcția `lang` acceptă ca parametru un AST:

```
print(lang(["_+_", <["a", <>], ["b", <>]>], 2));

$ ../../Linux_Mac/alki.sh -a lang.alk
{a, b}
```

Pentru a obține ASTul trebuie apelată funcția `expression` din parser.alk:

```
// update index
index = 0;
// the expression
input = "a+b";
// compute the ast
ast = expression();
print(ast);

$ ../../Linux_Mac/alki.sh -a parser.alk
[_+_, <[a, <>], [b, <>]>]
```

Se observă că etichetele din noduri sunt listate ca șiruri ("Alk feature"). Output-ul `[_+_, <[a, <>], [b, <>]>]` va trebui transformat în `["_+_", <["a", <>], ["b", <>]>]`. Implementarea completă a ASTului se găsește în ast.alk

4.

(a) Să se construiască automatul pentru expresia $b(a+c)^*b(aa+cc)*$.

(b) Să se execute algoritmul de generare a automatului din detaut.alk și să se deseneze automatul afișat de algoritm.

(c) Comparați rezultatele.

5.

(a) Să se construiască automatul pentru expresia $b(aca)^*b(a)*$.

(b) Să se explice cum se face căutarea unui șir descris de această expresie în textul (subiectul) *abacbbaabaababababba*, utilizând șablonul de mai jos.

Subiectul:

a	b	a	c	b	b	a	a	b	a	a	b	a	b	a	b	b	a
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

poziție de start în subiect (i)	poziție curentă în subiect (j)	stare stare	decizie/acțiune decizie/acțiune
0	0	0	nepotrivire, s = stare inițială (0), i++
1	1	0	potrivire, s = next(0,b), j++

(c) Să se compare rezultatul cu cel dat de algoritmul din detaut.alk.