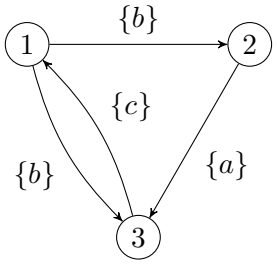
13. Übung (27.–31. Januar 2020)

Algorithmen und Datenstrukturen

Aufgabe 1 (AGS 9.5.19)

Der gewichtete Graph $G = (V, E, c)$ über dem Semiring $(\mathcal{P}(\{a, b, c\}^*), \cup, \circ, \emptyset, \{\varepsilon\})$ sei durch folgende graphische Darstellung gegeben:



Es soll für den Graph G das Prozessproblem mit Hilfe des Aho-Algorithmus gelöst werden.

(a) Geben Sie für G die modifizierte Adjazenzmatrix mA_G an.

(b) Berechnen Sie für den Aho-Algorithmus die Matrizen $D_G^{(1)}$ und $D_G^{(2)}$. Sie müssen nur die Matrixelemente aufschreiben, die sich gegenüber mA_G geändert haben.

(c) Geben Sie die letzte Zeile der Ergebnismatrix D_G des Aho-Algorithmus an.

(d) Wie verändert sich der Eintrag $D_G(3, 3)$, wenn zu dem Graphen G eine Kante von Knoten 3 zum Knoten 2 mit dem Gewicht $\{b\}$ zugefügt wird?

Zusatzaufgabe 1 (AGS 2.2.58 ⋆)

(a) Gegeben sei die Sprache $L = \{(ab)^n c^{m+1} d^k b^{n+m} \mid n, m \geq 0, k \geq 1\}$. Geben Sie eine EBNF-Definition $\mathcal{E}$ an, sodass $W(\mathcal{E}) = L$.

(b) Gegeben sei die EBNF-Definition $\mathcal{E} = (V, \Sigma, S, R)$ mit $V = \{S, A, B\}$, $\Sigma = \{a, b, c, d\}$ und

$$R = \{ \quad S ::= A \,\hat{\{}\, B \,\hat{\}}, \qquad A ::= aA \,\hat{(}\, bc \,\hat{|}\, d \,\hat{)}, \qquad B ::= \hat{[}\, B \,\hat{]}\, b \quad \}.$$

Übersetzen Sie $\mathcal{E}$ gemäß der Übersetzungsvorschrift *trans* aus der Vorlesung in ein System von Syntaxdiagrammen und geben Sie das Startdiagramm an. Sie müssen *keine Zwischenschritte*, sondern können direkt das übersetzte Syntaxdiagrammsystem angeben.

Zusatzaufgabe 2 (AGS 3.2.45 a, b)

(a) Vervollständigen Sie die Funktion `contains`, so dass ihr Rückgabewert 1 ist, falls das übergebene Array der Länge `bound` den Wert `elem` enthält, und sonst 0.

```
1   int contains(int array[], int bound, int elem) { ... }
```

(b) Gegeben sei der folgende Datentyp für Binärbäume.

```
1   typedef struct node * tree;
2   struct node { int value; tree left, right; };
```

Geben Sie eine Funktion **`void`** `path(tree t)` an, welche in einem Binärbaum `t` die Beschriftung jedes Knotens durch die Anzahl der Knoten auf dem Pfad zum Wurzelknoten ersetzt. Dem Wurzelknoten wird also die Zahl 1 zugeordnet, seinen direkten Nachfolgern die Zahl 2, deren direkten Nachfolgern die Zahl 3, und so weiter. Falls Sie eigene Hilfsfunktionen verwenden, geben Sie diese vollständig an!

Zusatzaufgabe 3 (AGS 4.23)
Gegeben sei folgendes C-Programm:

```
1  #include <stdio.h>
2
3  void g (int *x, int y);
4
5  void f (int z) {
6    int a = 1;
7    while (z != 0) {
8      /*label1*/
9      g(&z, a); /*$1*/
10     a = a * 2;
11   }
12   /*label2*/
13 }
14
15 void g (int *x, int y) {
16   /*label3*/
17   if (y >= 2) {
18     *x = 0;
19     /*label4*/
20     f(*x); /*$2*/
21   } else {
22     *x = *x + 1;
23   }
24   /*label5*/
25 }
26
27 int main () {
28   int m;
29   m = 2;
30   /*label6*/
31   f(m); /*$3*/
32   /*label7*/
33   return 0;
34 }
```

(a) Tragen Sie den Gültigkeitsbereich jedes Objektes in eine Tabelle ein. Nutzen Sie dazu die Zeilennummern.

(b) Setzen Sie das folgende Speicherbelegungsprotokoll fort. Dokumentieren Sie die aktuelle Situation beim Passieren der Marken (`label1` bis `label7`). Geben Sie jeweils den Rücksprungmarkenkeller und die *sichtbaren* Variablen mit ihrer Wertebelegung an. Die Inhalte von Speicherzellen nicht sichtbarer Variablen müssen Sie nur bei Änderungen eintragen. Die bereits festgelegten Rücksprungmarken sind *`$1`* bis *`$3`*.

Label	RM	1	2	3	4	5	6	7	8	9	10
label6	–	m 2									

Zusatzaufgabe 4 (AGS 6.1.13 ⋆)
Wenden Sie den Quicksort-Algorithmus auf die Folge $5, 9, 3, 8, 2$ an.

Zusatzaufgabe 5 (AGS 6.2.14)
Wenden Sie auf die Folge $4, 7, 2, 1, 6, 3, 5, 8, 0, 9$ den Heapsort-Algorithmus an.

Zusatzaufgabe 6 (AGS 7.1.1 ⋆)
Geben Sie zu den Pattern

(a) `abaabaaab`

(b) `aaabaaaa`

die jeweils mit Hilfe des KMP-Algorithmus (Knuth-Morris-Pratt) berechnete Verschiebetabelle an.

Zusatzaufgabe 7 (AGS 10.7 ⋆)

Bei beschränkten Modellen kann der Maximum-Likelihood-Schätzer im Allgemeinen nicht effizient bestimmt werden. Beinhaltet das Modell $\mathcal{M}$ aber die relative Häufigkeitsverteilung $\mathrm{rfe}(h)$ des gegebenen Korpus h, dann ist $\mathrm{mle}(\mathcal{M}, h) = \mathrm{rfe}(h)$, denn keine andere Wahrscheinlichkeitsverteilung erzeugt eine höhere Likelihood.

Bestimmen Sie für die folgenden Situationen das Wahrscheinlichkeitsmodell, zeigen Sie, dass die relative Häufigkeitsverteilung des betrachteten Korpus in diesem enthalten ist und bestimmen Sie den Maximum-Likelihood-Schätzer.

(a) Werfen eines Würfels, bei dem gegenüberliegende Seiten die gleiche Wahrscheinlichkeit aufweisen. Betrachten Sie den folgenden Korpus:

$$h(1) = 3\,, \qquad h(2) = 5\,, \qquad h(3) = 1\,, \qquad h(4) = 1\,. \qquad h(5) = 5\,, \qquad h(6) = 3\,.$$

(b) Werfen zweier unabhängiger Münzen. Betrachten Sie den folgenden Korpus:

$$h(K, K) = 2\,, \qquad h(K, Z) = 4\,, \qquad h(Z, K) = 4\,, \qquad h(Z, Z) = 8\,.$$

(c) Ziehen mit Zurücklegen aus einer Urne mit fünf Kugeln. Die Kugeln sind weiß, schwarz oder rot. Betrachten Sie den folgenden Korpus:

$$h(W) = 4\,, \qquad h(S) = 2\,, \qquad h(R) = 4\,.$$