Algorithmen und Datenstrukturen

14. Übungsblatt

Zeitraum: 30. Januar – 03. Februar 2017

Übung 1 (AGS 10.13)

Die Personen A und B spielen ein Spiel mit einer Münze mit den Beschriftungen 1 und 2, sowie mit einem dreiseitigen Würfel mit den Beschriftungen 1, 2, und 3. In jeder Runde werden die Münze und der Würfel geworfen. Der Spieler A gewinnt die Runde, falls die gefallene Zahl des Würfels echt größer ist als die auf der Münze. Ansonsten gewinnt B die Runde. Die Menge der möglichen Ergebnisse ist somit

$$X = \{M_1, M_2\} \times \{W_1, W_2, W_3\} ,$$

wobei das Element M_i für „die Münze zeigt die Zahl i“ stehen soll, und analog für W_i.

(a) Geben Sie den Analysator A für dieses Szenario an. Bestimmen Sie also A(„A gewinnt“) und A(„B gewinnt“).
Die Spieler spielen 31 Runden des Spiels und A gewinnt dabei 21 Mal. Sie erzeugen also einen Korpus h mit unvollständigen Daten. Bestimmen Sie h(„A gewinnt“) und h(„B gewinnt“)!

(b) Wir wollen aus diesem Korpus h mit dem EM-Algorithmus die Wahrscheinlichkeitsverteilungen von Münze und Würfel bestimmen. Die initiale Wahrscheinlichkeitsverteilung $q_0 = q_0^M \times q_0^W$ ist gegeben durch $q_0^M(M_1) = 1/3$ und $q_0^W(W_1) = q_0^W(W_2) = 1/4$. Dabei ist q_0^M die Wahrscheinlichkeitsverteilung der Münze, q_0^W die des Würfels. Welche Werte haben $q_0(M_1, W_1)$, $q_0(M_1, W_2)$, $q_0(M_1, W_3)$, $q_0(M_2, W_1)$, und $q_0(M_2, W_2)$?
Führen Sie den E-Schritt aus, vervollständigen Sie also den Korpus h zum Korpus h_1. Bestimmen Sie $h_1(M_1, W_1)$, $h_1(M_1, W_2)$, $h_1(M_1, W_3)$, $h_1(M_2, W_1)$, $h_1(M_2, W_2)$.

(c) Führen Sie nun den M-Schritt aus. Bestimmen Sie dafür zunächst die Teilkorpora h_1^M und h_1^W für die Münze bzw. den Würfel, also $h_1^W(W_1)$, $h_1^W(W_2)$, $h_1^M(M_1)$, und $h_1^M(M_2)$.
Schätzen Sie nun die Wahrscheinlichkeitsverteilung q_1^M der Münze sowie q_1^W des Würfels, indem Sie die relative Häufigkeit der Teilkorpora bestimmen, also $q_1^W(W_1)$, $q_1^W(W_2)$, $q_1^M(M_1)$, und $q_1^M(M_2)$.

Übung 2 (AGS 10.14)

Die Personen A und B spielen ein Spiel. In jeder Runde werden zwei Zufallszahlen x und y bestimmt, wobei x Werte aus der Menge $\{1, 2\}$ und y aus $\{1, 2, 3\}$ annehmen kann. Der Spieler A gewinnt die Runde, falls die Summe $x+y$ eine ungerade Zahl ist. Ansonsten gewinnt B die Runde. Die Menge der möglichen Ergebnisse ist somit $X = \{1, 2\} \times \{1, 2, 3\}$.

(a) Geben Sie den Analysator A für dieses Szenario an, also A(„A gewinnt“) und A(„B gewinnt“).
Die Spieler spielen 87 Runden des Spiels und A gewinnt dabei 60 Mal. Geben Sie den Korpus h mit unvollständigen Daten an, den sie erzeugen, also h(„A gewinnt“)und h(„B gewinnt“).

(b) Wir wollen aus diesem Korpus h mit dem EM-Algorithmus die Wahrscheinlichkeitsverteilungen der beiden Zufallszahlen bestimmen. Die initiale Wahrscheinlichkeitsverteilung $q_0 = q_0^x \times q_0^y$ ist gegeben durch $q_0^x(1) = 2/3$ und $q_0^y(1) = q_0^y(2) = 1/5$. Dabei ist q_0^x die Wahrscheinlichkeitsverteilung von x und q_0^y die von y. Bestimmen Sie die Werte $q_0(i, j)$ für $1 \le i \le 2$ und $1 \le j \le 3$.
Führen Sie den E-Schritt aus, vervollständigen Sie also den Korpus h zum Korpus h_1. Geben Sie für alle $1 \le i \le 2$ und $1 \le j \le 3$ die Werte $h_1(i, j)$ an.

(c) Führen Sie nun den M-Schritt aus. Bestimmen Sie dafür zunächst die Teilkorpora h_1^x und h_1^y für x bzw. y. Geben Sie also für alle $1 \leq i \leq 2$ und $1 \leq j \leq 3$ die Werte $h_1^x(i)$ bzw. $h_1^y(j)$ an.
Schätzen Sie nun die Wahrscheinlichkeitsverteilungen q_1^x sowie q_1^y von x bzw. y, indem Sie die relative Häufigkeit der Teilkorpora bestimmen. Geben Sie dafür wieder für alle $1 \leq i \leq 2$ und $1 \leq j \leq 3$ die Werte $q_1^x(i)$ bzw. $q_1^y(j)$ an.

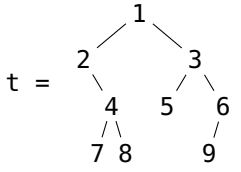
Zusatzaufgabe 1 (AGS 3.2.38 ⋆)
Für die folgenden Teilaufgaben gelten folgende `#includes`, und Typdefinitionen für Listen und Binärbäume:

```
#include <stdio.h>
#include <stdlib.h>

typedef struct elem *list;
typedef struct elem {
  int value;
  list next;
} elem;

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

(a) Schreiben Sie ein Programm, welches den Nutzer wiederholt zur Eingabe einer ganzen Zahl auffordert, bis er eine `0` eingibt. Danach soll das Programm ausgeben, wie viele gerade und wie viele ungerade Zahlen der Nutzer eingegeben hat. Die abschließende `0` soll dabei nicht mitgezählt werden.

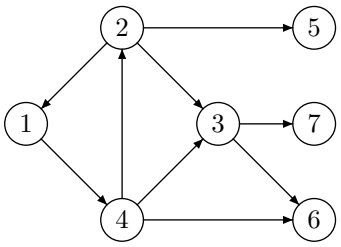
(b) Schreiben Sie die Funktion `void histogram(list l, int h[], int length)`, welche für alle `i` mit `0` ≤ `i` < `length` den Array-Eintrag `h[i]` durch die Anzahl der Vorkommen von `i` in `l` ersetzt. *Die Liste soll dabei nur einmal durchlaufen werden.* Gehen Sie davon aus, dass `h` genau `length` Einträge hat, treffen Sie jedoch keine Annahmen über die Einträge von `h` oder `l`.

(c) Ein Pfad eines Baumes ist die Liste aller Knotenbeschriftungen auf dem Weg von der Wurzel des Baumes zu einem Knoten. (Daraus folgt, dass der leere Baum keinen Pfad enthält und jeder Pfad mindestens ein Element enthält, nämlich die Beschriftung des Wurzelknotens.)
Implementieren Sie die Funktion `int isPath(list l, tree t)`, welche `1` zurück gibt, wenn der übergebene Baum einen als Liste übergebenen Pfad beinhaltet, und ansonsten `0` zurück gibt.
Beispiel: (Listen werden vereinfacht als Sequenz der Elemente in eckigen Klammern dargestellt.)




```
isPath([], t) = 0
isPath([1, 2], t) = 1
isPath([1, 2, 7], t) = 0
isPath([1, 2, 4, 7], t) = 1
```

Zusatzaufgabe 2 (AGS 9.2.8)
Der gerichtete Graph $G = (V, E)$ sei durch folgende Darstellung gegeben:



(a) Wenden Sie auf den Graphen G den DFS-Algorithmus mit dem Startknoten 1 an, und bestimmen Sie auf diese Weise einen depth-first tree. Geben Sie drei unterschiedliche Lösungen an. Zwischenschritte zu den Lösungen brauchen Sie nicht anzugeben.
(b) Transformieren Sie G in den ungerichteten Graphen $G' = (V', E')$, indem Sie $V' = V$ setzen und E' nach der Vorschrift $E' = E \cup \{(j,i) \mid (i,j) \in E\}$ erzeugen. Wenden Sie nun auf G' den BFS-Algorithmus mit dem Startknoten 1 an, und bestimmen Sie einen breadth-first-tree. Geben Sie hier zwei unterschiedliche Lösungen an. Zwischenschritte zu den Lösungen brauchen Sie nicht anzugeben.

Zusatzaufgabe 3 (AGS 9.3.1 ⋆)
Folgende Kanten eines gerichteten Graphen G seien gegeben:

$$(1,2,1),\ (2,4,4),\ (4,1,2),\ (3,1,5),\ (2,3,2),$$

wobei die Notation wie folgt zu lesen ist: $(i,j,c) =$ (Anfangsknoten, Endknoten, Entfernung).
(a) Geben Sie für G die zugehörige modifizierte Adjazenzmatrix mA_G an.
(b) Berechnen Sie mit Hilfe der Rekursionsformel für kürzeste Wege die Einträge $D_G^{(1)}(i,j)$ für $i = 4$ und $1 \leq j \leq 3$.
(c) Berechnen Sie für G die Matrix D_G der kürzesten Wege zwischen zwei Knoten. Zwischenschritte brauchen Sie hier keine anzugeben.

Zusatzaufgabe 4 (AGS 2.2.33 ⋆)
Sei $\mathcal{E} = (V, \Sigma, S, R)$ mit $V = \{S, B\},\ \Sigma = \{b, c\}$ und $R = \{\ S ::= [Bcc],\ B ::= (Sbb \mid bb)\ \}$.

Berechnen Sie die syntaktischen Kategorien $W(\mathcal{E}, S)$ und $W(\mathcal{E}, B)$ mit Hilfe der Fixpunktsemantik. Gehen Sie dazu in den folgenden Schritten vor:
(a) Dokumentieren Sie mindestens 5 Iterationsschritte,
(b) geben Sie explizit $f^i(\bot)$ für $i \geq 1$ an (wobei f entsprechend der Vorlesung definiert ist),
(c) und schreiben Sie in Mengenschreibweise die Sprachen $W(\mathcal{E}, S)$ und $W(\mathcal{E}, B)$ auf.