



Abgabe: 02.12.2008 (vor der Vorlesung)

Aufgabe 7.1 (H) Matrixmultiplikation

In dieser Aufgabe soll ein Webseiten-Ranking berechnet werden. Wir betrachten folgende Webseiten:

- 1) www.nichtsowichtig.de
- 2) www.tum.de
- 3) wwwseidl.in.tum.de
- 4) portal.mytum.de

Untersuchungen haben ergeben, dass ein Benutzer der www.nichtsowichtig.de besucht, mit einer Wahrscheinlichkeit von 40 % danach die Seite www.tum.de besuchen. Mit einer Wahrscheinlichkeit von 60 % wird danach wwwseidl.in.tum.de besucht. Insgesamt lassen sich die Ergebnisse der Untersuchung in folgender Matrix zusammenfassen:

$$P = \begin{pmatrix} 0 & 0 & 0.2 & 0.3 \\ 0.4 & 0 & 0.8 & 0 \\ 0.6 & 0 & 0 & 0.7 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Dabei bezeichnet der Eintrag in der i -ten Zeile und der j -ten Spalte die Wahrscheinlichkeit, dass nach Webseite j die Webseite i besucht wird. Ein Webseiten-Ranking sortiert die Seiten danach, mit welcher erwarteten Wahrscheinlichkeit sie besucht werden. Auf Grundlage solcher Daten ergibt sich das Webseiten-Ranking dann durch

$$\lim_{i \rightarrow \infty} P^i \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Ihre Aufgabe besteht darin, eine Approximation dieses Wertes zu bestimmen.

Zur Lösung der gestellten Aufgabe gehen wir wie folgt vor. Einen Vektor $x = (x_1, \dots, x_n)$ repräsentieren wir durch eine streng monoton aufsteigend sortierte Liste von Tupeln (i, x_i) . Dabei werden nur Paare (i, x_i) mit $x_i \neq 0$ aufgeführt. Z.B. wird der Vektor $(0, 0, 0.1, 0, 0.5, 0.4, 0, 0)^\top$ durch $[(3, 0.1); (5, 0.5); (6, 0.4)]$ repräsentiert.

Eine Matrix wird durch eine Liste von Paaren (j, v) repräsentiert, wobei j ein Zeilenindex und v ein Vektor ist, welcher die j -te Zeile repräsentiert. Z.B. wird die Matrix P durch die Liste

$$[(1, [(3, 0.2); (4, 0.3)]); (2, [(1, 0.4); (3, 0.8)]); (3, [(1, 0.6); (4, 0.7)]); (4, [(2, 1)])]$$

repräsentiert.

- a) Schreiben Sie eine Funktion `v_mult`, die das Skalarprodukt zweier Vektoren berechnet.
Zur Erinnerung: $(a_1, \dots, a_m)(b_1, \dots, b_m)^T = \sum_{i=1}^m a_i \cdot b_i$
- b) Schreiben Sie eine Funktion `m_v_mult`, die für eine Matrix

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} \in \mathbb{R}^{m \times n}$$

und einen Vektor $x \in \mathbb{R}^n$ das Produkt Ax berechnet.

Zur Erinnerung: $Ax = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} x = \begin{pmatrix} a_1 x \\ \vdots \\ a_n x \end{pmatrix} \in \mathbb{R}^m$

- c) Schreiben Sie eine Funktion `iter`. Falls `p` die Matrix P und `x` den Vektor x repräsentieren, soll der Aufruf `iter p x i` den Vektor $x_i := P^i x$ berechnen. Hinweis: Für alle $i \in \mathbb{N}_0$ gilt:

$$x_i = \begin{cases} Px_{i-1} & \text{falls } i > 0 \\ x & \text{falls } i = 0. \end{cases}$$

- d) (**freiwillige Zusatzaufgabe**) Schreiben Sie eine Funktion `m_m_mult`, die zwei Matrizen miteinander multipliziert.

Aufgabe 7.2 (P) Listenfunktionale, Funktionen höherer Ordnung

Definieren Sie die Funktion `fold_right` mithilfe der Funktion `fold_left`. Konstruieren Sie eine aus 1.000.000 Einsen bestehende Liste `list`. Testen Sie ihre Funktion anhand des Aufrufs `fold_right (+) list 0`. Testen Sie den gleichen Aufruf anhand der in der Vorlesung angegebenen Implementierung. Was fällt Ihnen auf?

Aufgabe 7.3 (P) MiniJava-Interpreter

Ziel dieser Aufgabe und der Hausaufgabe auf dem nächsten Blatt ist es, einen Interpreter für eine Java-ähnliche Sprache in OCaml zu implementieren. Die arithmetischen Ausdrücke e dieser Sprache sind durch folgende Grammatik spezifiziert:

$$e ::= \text{Const}(\langle \text{int} \rangle) \mid \text{Var}(\langle \text{string} \rangle) \mid \text{Add}(e, e) \mid \text{Sub}(e, e) \mid \text{Mul}(e, e) \mid \text{Div}(e, e)$$

Beispielsweise ist

$$\text{Add}(\text{Var}(x), \text{Mul}(\text{Const}(2), \text{Const}(3)))$$

ein arithmetischer Ausdruck. Gehen Sie wie folgt vor:

- Definieren Sie zunächst einen Typ `expr` zur Repräsentation arithmetischer Ausdrücke.
- Nutzen Sie den auf der Übungsseite bereitgestellten Parser, um Ausdrücke aus einer Datei zu lesen und diese anschließend in einen Wert vom Typ `expr` zu konvertieren.
- Definieren Sie eine Funktion `eval : expr -> int`, die arithmetische Ausdrücke auswertet. Gehen Sie dabei zunächst davon aus, dass alle Variablen den Wert 0 haben.
- In dieser Aufgabe werden Variablen als `string`-Werte repräsentiert. Eine Funktion

$$\text{rho} : \text{string} \rightarrow \text{int}$$

ist eine Variablenbelegung. Sie ordnet jeder Variablen einen `int`-Wert zu. Schreiben Sie eine Funktion

```
update : (string->int)->string->int->string->int
```

zum Aktualisieren von Variablenbelegungen. Der Aufruf `update rho x v` soll die Variablenbelegung `rho'` zurückgeben, die `rho` entspricht, bis darauf, dass `rho' x = v` gilt.

- e) Ändern Sie die Funktion `eval` entsprechend. Ihre Funktion `eval` benötigt jetzt zusätzlich zu dem auszuwertenden Ausdruck `e` eine Variablenbelegung `rho` als Parameter.
- f) (**Anspruchsvoll**) Ergänzen Sie Ihre Implementierung, so dass der Benutzer nach Start des Programms dazu aufgefordert wird Werte für alle im Ausdruck vorkommenden Variablen einzugeben. Die eingegebenen Werte sollen dann zur Auswertung des Ausdrucks verwendet werden.

Hinweis: Verwenden Sie die OCaml-Funktionen `string_of_int`, `int_of_string`, `print_string` und `read_int`.