

Tabmentrepräsentationen

17.08.2021

1 Das Tabment

o++o besitzt ein grundlegendes Objekt: Das Tabment

Das Tabment ist eine Struktur, die rekursiv definiert ist. Man muss einzelne Regeln mehrfach anwenden, um nachzuweisen, dass bestimmte Objekte Tabmente sind. Die nachfolgenden Ausführungen können aber auch ohne diese Definition verstanden werden.

1. Jedes Objekt von elementarem Typ (TEXT, WORT, ZAHL, PZAHL, RATIO, ONR, BAR) ist ein Tabment.
2. Sind t_1, t_2, \dots, t_n Tabmente, so sind auch ein Paar t_1, t_2 und das Tupel t_1, t_2, \dots, t_n Tabmente (OCaml: `Tuple_t[t1;t2; ...,tn]`).
3. Ist t ein Tabment und n ein NAME, so ist auch t tag0 n (OCaml: `Tag0(n,t)`) ein Tabment. t wird in Anfangs- und Endtag n eingeschlossen.
4. Sind t_1, t_2, \dots, t_n Tabmente des Typs s und ist c (Menge, Liste, ...) ein Kollektionssymbol, so kann man aus diesen eine Kollektion bilden. (OCaml: `Coll_t(c,s,[t1;t2; ...,tn])`).
5. Es gibt ein leeres Tabment: `empty_t`
6. Ist t ein Tabment vom Typ sk , so ist auch `Alternate_t([s1;s2;...;sk;...;sn],t)` ein Tabment.

Das ist eine „unsichtbare“ abstrakte Definition des Objekts Tabment, die unmittelbar im Arbeitsspeicher umgesetzt werden kann. Für das optische Wahrnehmen, bietet o++o bislang die konkreten Darstellungen `tab`, `ment`, `hsq`, `tabh`, `hsqh`, `hsq2` und `xml` an. `xml` und `ment` sind sehr allgemein und ähnlich - sie unterscheiden sich zur Zeit nur in der Darstellung der Metadaten. Man kann mit diesen beiden Darstellungen Dokumente vorteilhaft repräsentieren. Die XML-Darstellung von tabellarischen Daten besitzt in der Regel jedoch viel Redundanz, insbesondere bei den Metadaten. Ferner ist sie nicht sehr übersichtlich. Daher verwenden wir auch die nicht so allgemeinen `tab`- und `hsq`-Darstellungen. Eine `hsq2`-Darstellung wird nur intern für die Erzeugung von Diagrammen verwendet. Ferner kann man bestimmte Tabmente auch als Bild darstellen. Wenn man jedes Element des Tabments als Punkt interpretiert (X,Y-Koordinaten und dazwischen die Farbe=RGB-Wert), kann die Menge aller Punkte ein gewünschtes Bild darstellen. Neben dem `bild`-Button gibt es noch den `graphik`- und `diagramm`-Button. Sind tabellarische Daten gegeben, so stellt der `graphik`-Button die Tabelle mit Gitternetzlinien dar. Der `diagramm`-Button überführt geeignete Tabellen in strukturierte Diagramme, wobei der Nutzer mit seiner Anfrage sogar die Farbgebung steuern kann.

Die verschiedenen Repräsentationen für ein einziges Tabment erlauben es, den abstrakten Tabmentbegriff leichter zu erfassen. Für Entwickler und Theoretiker stellen wir noch den `ocaml`-Button zur Verfügung.

2 tab- und tabh-Dateien

`tab`-Dateien enden auf `.tab`. Sie erlauben es nicht-rekursive strukturierte Tabellen mit Wiederholgruppen übersichtlich darzustellen. Hier hat jede Spalte eine bestimmte Breite, die sich aus dem Maximum der Breiten des Feldnamens und aller Werte der Spalte ergibt. Schon hieraus

ergibt sich, dass typische Dokumente nicht in tab-Form gebracht werden sollten. Hier sollte man auf die grafik-, ment- oder xml-Darstellung ausweichen. In der tab-Repräsentation sind nur die Spaltennamen, die von elementaren Typ sind, sichtbar. Übergeordnete Tags wie RGB oder der Dateiname sind nur sichtbar, wenn man die Metadaten nicht mit „normal“ ausgibt.

Die Elemente einer Kollektion gehören standardmäßig untereinander. Das soll mit der folgenden kleinen Tabelle mit Klausuren und Noten illustriert werden. Wir trennen Klausuren und Noten, damit wir gewichtete Durchschnitte berechnen können. Wir repräsentieren hier die Noten und Klausuren durch Listen (l), damit wir von einer ersten, zweiten,... letzten Klausur (Note) sprechen können und gleiche Noten wiederholt auftreten können. Da wir hier die Fächer mit den Klausuren und Noten als Menge spezifiziert haben, erscheint kein Fach mehrfach. Bei einer gib-Anweisung werden bei Mengen (m) stets Duplikate eliminiert.

```
FACH,  KLAU1, NOTE1  m
Mathe  1      1
        k      1
        5      2
                3
                2
                4
                3
Physik  3      3
        2      4
                1
                2
                1
(meinenoten.tab)
```

Meine Menge besteht hier aus 2 Elementen; Mathe mit den entsprechenden Noten und Physik mit anderen. In Mathematik wurden 3 Klausuren geschrieben, wobei ich zur zweiten Klausur krank war. Meine erste Note in Physik (in der NOTE-Liste) ist beispielsweise eine 3 und meine letzte 1. Von der Übersichtlichkeit der Struktur ist das sicher die beste Repräsentation. Sie benötigt aber relativ viele Zeilen auf dem Bildschirm. Daher lassen wir zu, dass die obigen einfachen Listen auch horizontal geschrieben oder ausgegeben werden können. Es folgt eine kompaktere Darstellung von meinenoten.tab (horizontale Klausur- und Noten- Wiederholgruppenelemente):

```
FACH,  KLAU1, NOTE1  m
Mathe  1 k 5  1 1 2 3 2 4 3
Physik 3 2   3 4 1 2 1
(meinenoten.tabh)
```

Diese Repräsentation wird mit dem tabh-Button (h für horizontal) standardmäßig erzeugt, wenn die vorgegebene Breite in width ausreicht. meinenoten1.tab wird dagegen mit dem tab-Button erzeugt. Die Breite einer Spalte wird aus der maximalen Breite seiner Werte und der Breite der Überschrift berechnet. Man könnte die Breite einer Spalte also durch zusätzliche Zeichen des Spaltennamens vergrößern.

Die Breite der Klausurspalte reicht beispielsweise vom K bis ein Zeichen vor dem N.

Die tab-Darstellung (meinenoten.tab) ist sicher vom methodischen Standpunkt vorzuziehen, wird sich aber sicher nicht in der Praxis durchsetzen. Wenn ein kleinerer width-Wert vorgegeben wird (hier 23), könnte das gleiche Tabment auch so aussehen:

```
FACH,  KLAU1, NOTE1  m
Mathe  1 k 5  1 1 2 3
                2 4 3
```

Physik 3 2 3 4 1 2

1

(meinenoten.tabh)

Die Spaltennamen (Metadaten) dürfen keine Kleinbuchstaben enthalten. In den Elementen (Primärdaten) können beliebige Zeichen vorkommen. tab-Dateien sind recht übersichtlich, sollten aus Effizienzgründen aber nicht für sehr große Datenmengen zum Abspeichern genutzt werden.

3 ment- und xml-Dateien

ment-Dateien sind für DokuMENTe gedacht; sind vom methodischen Standpunkt aber auch für Tabellen interessant. Hier kann eine Spalte sehr breit werden, ohne dass die Darstellung unübersichtlich wird. Ein Spaltenwert wird durch einen Anfangs- und Endtag (meist ein Spaltenname) begrenzt. Beispiel:

```
<FLUSS>Iilm</FLUSS>
```

```
<QUELLE>Stützerbach (Thüringen)</QUELLE>
```

```
<MUENDUNG>Saale</MUENDUNG>
```

...

Die Syntax der Tags wurde von XML beibehalten, obwohl wir diese als nicht sehr „endnutzerfreundlich“ ansehen. ment-Dateien unterscheiden sich bislang nur in der Darstellung der Metadaten von den xml-Dateien. ment-Metadaten stimmen mit den tab-Metadaten und denen der anderen Dateikonzepte überein. Unsere xml-Dateien benötigen eine DTD (Document Type Definition), damit wir zwischen Kollektion und Tupel (Paar) unterscheiden können.

Obige Tabelle meinenoten.tabh nimmt die folgende Gestalt an, wenn sie mit dem ment- oder xml-Button ausgegeben wird.

```
<MEINENOTEN>
```

```
  <FACH>Mathe</FACH>
```

```
  <KLAU>1</KLAU>
```

```
  <KLAU>k</KLAU>
```

```
  <KLAU>5</KLAU>
```

```
  <NOTE>1</NOTE>
```

```
  <NOTE>1</NOTE>
```

```
  <NOTE>2</NOTE>
```

```
  <NOTE>3</NOTE>
```

```
  <NOTE>2</NOTE>
```

```
  <NOTE>4</NOTE>
```

```
  <NOTE>3</NOTE>
```

```
  <FACH>Physik</FACH>
```

```
  <KLAU>3</KLAU>
```

```
  <KLAU>2</KLAU>
```

```
  <NOTE>3</NOTE>
```

```
  <NOTE>4</NOTE>
```

```
  <NOTE>1</NOTE>
```

```
  <NOTE>2</NOTE>
```

```
  <NOTE>1</NOTE>
```

```
</MEINENOTEN>
```

Den Gültigkeitsbereich der einzelnen Tags FACH, KLAU, NOTE erkennt man klar. Der Dateiname MEINENOTEN ist ein Tag, der alle Daten der Datei einschließt. Er ist notwendig, wenn mehrere Dateien benötigt werden und dann Spaltennamen mehrfach vorkommen. MEINENOTEN und andere höhere Tags werden bei den vorangehenden Repräsentationen der Primärdaten ausgeblendet, um eine einfache, übersichtliche Tabellenstruktur zu garantieren.

Mit Hilfe von ment-Dateien kann man den Unterschied der folgenden beiden Miniprogramme gut verdeutlichen:

```
X := 1 .. 5
```

ergibt:

```
<X>
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
</X>
```

```
X1 := 1 .. 5
```

ergibt:

```
<TABM>
```

```
  <X>1</X>
```

```
  <X>2</X>
```

```
  <X>3</X>
```

```
  <X>4</X>
```

```
  <X>5</X>
```

```
</TABM>
```

D.h., im ersten Fall repräsentiert `X` eine Liste von 5 Zahlen und im zweiten Fall repräsentiert `X` nur jeweils eine Zahl (ein Listenelement). Der Tag `TABM` wurde vom System selbst erzeugt, da ein `ment`-Dokument (wie auch ein `xml`-Dokument) immer einen Tag für das ganze Tabment besitzen muss.

4 `hsq`-, `hsqh`- und `hsq2` -Dateien

Das `hsqh`-Format ist für die Repräsentation von Tabellen mit einer überschaubaren Anzahl von Spalten (und damit auch Hierarchieebenen) gedacht; also insbesondere für ein Handy. Unsere obige Notentabelle nimmt jetzt die folgende Gestalt an:

```
FACH, KLAU1, NOTE1 m
```

```
FACH
```

```
  KLAU
```

```
    NOTE
```

```
Mathe
```

```
  1 k 5
```

```
    1 1 2 3 2 4 3
```

```
Physik
```

```
  3 2
```

```
    3 4 1 2 1
```

```
    (meinenoten.hsqh)
```

Die verschiedenen Hierarchieebenen unterscheiden sich durch die Anzahl der vorangestellten Leerzeichen. Zur nullten Ebene gehören 0 Leerzeichen und die beiden weiteren Ebenen haben jeweils ein zusätzliches Leerzeichen.

Die `hsq`-Darstellung ist etwas einfacher aber dafür in diesem Beispiel unübersichtlicher, da sie nicht so kompakt ist.

```
FACH, KLAU1, NOTE1 m
```

FACH
KLAU
NOTE

Mathe

1
k
5
1
1
2
3
2
4
3

Physik

3
2
3
4
1
2
1

(meinenoten.hsq)

hsq2-Dateien sind den hsq-Dateien sehr ähnlich. Abweichend davon wird die Anzahl der Leerzeichen durch eine Zahl repräsentiert. Als Trennzeichen zwischen den Ebenen wird nicht das Newline-Zeichen sondern das Symbol „#“ benutzt. Die Metadaten werden weiter aufgeschlüsselt. Metadaten und Primärdaten werden durch \$ getrennt.

```
#0 3 3 #1 FACH,KLAU1,NOTE1 m #2 1 FACH WORT 0 #2 2 KLA MIXE 1 #2 3  
NOTE ZAHL 2 #3 0 1 #3 1 1 #3 2 1 $ #0 Mathe #1 1 #1 k #1 5 #2 1  
#2 1 #2 2 #2 3 #2 2 #2 4 #2 3 #0 Physik #1 3 #1 2 #2 3 #2 4 #2 1  
#2 2 #2 1
```

(meinenoten.hsq2)

Im Einzelnen haben die Metadaten folgende Bedeutung:

#0 Anzahl_der_Felder Anzahl_der_Hierarchieebenen

#1 Schema_des_Tabments

#2 Feld_nummer (elementarer)_Typ_des_Feldes Hierarchieebenennummer

#3 Hierarchieebenennr Nummer_der_übergeordneten_Hierarchieebene

Da in den Klausuren auch Worte (k) vorkamen, ist KLAU nicht vom Typ ZAHL, sondern vom Typ MIXE, d.h. verschiedene elementare Typen ohne Leerzeichen sind möglich (hier WORT und ZAHL).

5 Die bild-Ausgabe

Wir denken, dass unser bild-Button Bilder vom mathematischen Standpunkt mit den einfachsten Konzepten darstellt. Ein Punkt ist ein Paar von Zahlen. Die erste Zahl ist die X-Koordinate (waagrecht) die zweite die Y-Koordinate (senkrecht). Davor könnte noch ein RGB-(Farbwert,RGB=RED,GREEN,BLUE) stehen.

Punkte müssen immer in Kollektionen verpackt sein.

```
<TAB!  
X,Y  
1 2  
!TAB>
```

bringt daher eine Fehlermeldung.

```
<TAB!  
X,Y 1  
1 2  
!TAB>
```

aber nicht. Allerdings sieht man einen einzelnen Punkt zur Zeit schwer.

```
<TAB!  
X,Y 1  
1 2  
1 3  
1 4  
2 5  
!TAB>
```

Hier liegen die ersten drei Punkte untereinander. Sie haben die gleiche X-Koordinate, weshalb man auch eine kompaktere Ausgangstabelle für das gleiche Bild wählen kann.

```
<TABH!  
X,Y1 1  
1 2 3 4  
2 5  
!TABH>
```

Für Bilder mit hunderten oder mehr Punkten muss man die Zahlen generieren.

```
X1 := 1 ..100  
Y:= 0.1*X
```

Hier werden einfach die ersten 100 natürlichen Zahlen generiert und es wird zu jedem X-Wert der Y-Wert berechnet. Die tab-Darstellung des Programms ist eine Wertetabelle mit 100 Punkten:

```
X, Y 1  
1 0.1  
2 0.2  
3 0.3  
4 0.4  
5 0.5  
6 0.6  
7 0.7  
8 0.8  
9 0.9  
10 1.  
.  
99 9.9  
100 10.
```

Offensichtlich ist die dargestellte Funktion eine Gerade mit dem Anstieg 0.1.

Wir wollen dazu noch die X-Achse einzeichnen:

```
X1 := 1 ..100  
Y:= 0.1*X  
Y0:=0*X
```

Die tab-Darstellung lautet jetzt:

```
X, Y, Y0 1
```

```

1  0.1  0
2  0.2  0
.  .  .
99 9.9  0
100 10.  0

```

Würde man jedoch das Programm in folgender Weise „vereinfachen“,

```

X1 := 1 ..100
Y:= 0.1*X
Y0:=0

```

so ergibt sich die Tabausgabe.

```

X,  Y  1, Y0
1  0.1  0
2  0.2
3  0.3
.  .  .
99 9.9
100 10.

```

Man erkennt, dass der Y0-Wert 0 jetzt keine zugehörige zweite Koordinate mehr besitzt, weshalb die Tabelle nicht mehr bildlich dargestellt werden kann.

Wollen wir weiter die Sinus-Funktion für die gleichen X-Werte darstellen, so ergibt sich das Programm:

```

X1 := 1 ..100
Y := 0.1*X
Y0 := 0*X
SIN:= X sin

```

Der Verlauf ist nur schwach zu erkennen, deshalb verzehnfachen wir die Anzahl der Punkte einfach:

```

X1 := 1 ...100!0.1
Y := 0.1*X
Y0 := 0*X
SIN:= X sin

```

Hier werden 1000 X-Werte von 1 bis 100 mit der Schrittweite 0.1 generiert. "...“ ist im Gegensatz zu "...“ eine Operation mit 3 Inputwerten (anfangswert ... endwert ! schrittweite).

Fügt man jetzt noch eine schneller wachsende Funktion (X^2) hinzu,

```

X1 := 1 ...100!0.1
Y := 0.1*X
Y0 :=0*X
SIN:= X sin
PARABEL:=X hoch 2

```

, so erhält man ein nicht so sehr schönes Bild, da $100 \text{ hoch } 2 = 10000$ ist. Damit besteht auch die Y-Achse aus 10000 Einheiten. Da die X-Achse nur aus 100 Einheiten besteht, kann das eigentlich nicht viel besseres gezeichnet werden, wenn man das Bild nicht verzerren will. Wenn man ein besseres Bild will, kann man den X-Bereich einschränken.

```

X1 := -5 ...5!0.01
Y := 0.1*X
Y0 := 0*X
SIN:= X sin

```

```
PARABEL:= X hoch 2
```

Fügt man hier eine noch schneller wachsende Funktion hinzu, so „verschlechtert“ sich das Bild wieder. Daher schränken wir den X-Bereich weiter ein:

```
X1 := -2 ...2!0.01
Y  := 0.1*X
Y0 := 0*X
SIN:= X sin
PARABEL:= X hoch 2
EHOCHX := e hoch X
```

Will man dennoch die letzten Funktionen in einem größeren Bereich darstellen (verzerren), so kann man die Logarithmusfunktion anhängen. Da man dann vielleicht nicht mehr zwischen der Normalparabel und der Exponentialfunktion unterscheiden kann, zeichnen wir die erste rot und die e-Funktion grün.

```
X1 := 0 ...10!0.01
Y0 := 0*X
SIN:= X sin
PARABEL:=X hoch 2 ln
EHOCHX := e hoch X ln
RGBPARABEL:=red leftat PARABEL
RGBEHOCHX :=green leftat EHOCHX
```

Eine andere Möglichkeit diese Funktionen ansprechend in einem Bild auszugeben, besteht darin, nur den Definitionsbereich der schnell wachsenden Funktionen einzuschränken:

```
X1 := -5 ...10!0.01
Y0 := 0*X
SIN:= X sin
PARABEL:= if X abs < 3 then X hoch 2
EHOCHX := if X abs < 2 then e hoch X
RGBPARABEL:= red leftat PARABEL
RGBEHOCHX := green leftat EHOCHX
```

Das Funktionsbild kann weiter verbessert werden. Da die Y-Achse aber keine Funktion ist, kann sie nicht in gleicher Weise erzeugt werden. Wir führen dafür eine zweite Kollektion ein, die nur den X2-Wert 0 beinhaltet.

```
X1 := -5 ...10!0.01
Y0 := 0*X
SIN:= X sin
PARABEL:= if X abs < 3 then X hoch 2
EHOCHX := if X abs < 2 then e hoch X
RGBPARABEL:= red leftat PARABEL
RGBEHOCHX := green leftat EHOCHX
X21 := 0
Y21 := -2 ... 8!0.01 at X2
```

Das Schema der tab-Ausgabe lautet hier:

```
X, Y0, SIN, (REDPARABEL, GREENPARABEL, BLUEPARABEL, PARABEL           ?),
(REDEHOCHX, GREENEHOCHX, BLUEEHOCHX, EHOCHX ?) 1, (X2, Y21 1)
```

Das zugehörige Gesamtschema kann mit META both erzeugt werden:

```
<META!
```

```
TABMENT! (X, Y0, SIN, (RGBPARABEL, PARABEL ?), (RGBEHOCHX, EHOCHX ?) 1),
(X2, Y21 1)
RGBEHOCHX RGBPARABEL! PZAHL, PZAHL, PZAHL
X2! ZAHL
EHOCHX PARABEL SIN X Y0 Y2! PZAHL
!META>
```

Man kann auch ein vollständiges Koordinatensystem programmieren und dieses dann „include“n, damit man es nicht jedes mal wieder programmieren bzw. kopieren muss. Das zeigen wir aber an anderer Stelle.

6 Die diagramm-Ausgabe

Hiermit lassen sich vom Nutzer strukturierte Diagramme erzeugen. Wir beschränken uns auf Säulendiagramme. Bei klassischen Ansätzen ist eine Tabelle des folgenden Typs gegeben, die beispielsweise durch eine Anfrage erzeugt werden kann.

```
<TAB!
NAME, BMI m
Klaus 23.4
Bernd 24.1
Julia 19.6
!TAB>
```

Mit dem Diagramm-Button wird hieraus ein Diagramm mit 3 Säulen und 3 Unterschriften erzeugt. Mit o++o kann man den Säulen Farben selbst zuordnen. Durch

```
RGB:= black leftat BMI
```

wird jede Säule schwarz dargestellt. Durch

```
RGB:=if BMI<23.5 then green else red leftat BMI
```

werden die Problemkandidaten sichtbar.

Hat man eine Tabelle des Typs

```
NAME, BMI, LAENGE, GEWICHT 1
```

gegeben oder erzeugt, so sollte man vor jeder darzustellenden Säule eine Farbe angeben. Beispielsweise:

```
RGBROT := red leftat BMI
RGBGRUEN := green leftat LAENGE
RGBBLAU := blue leftat GEWICHT
```

Die entstehende Tabelle hat dann das Schema:

```
NAME, RGBROT, BMI, RGBGRUEN, LAENGE, RGBBLAU, GEWICHT 1
```

Man kann auch Diagramme strukturierter Tabellen erzeugen.

```
ABTEILUNG, DURGEHALT, (NAME, GEHALT m) m
```

Wie in der XML- und hsq2-Darstellung folgen hier jeder Durchschnittsgehaltssäule alle Gehaltssäulen der Mitarbeiter der Abteilung. Hier bietet es sich an die DURGEHALT-Säulen kräftiger zu färben als die Gehaltssäulen.

```
RGB1:= darkgreen leftat DURGEHALT
```

```
RGB2:= green leftat GEHALT
```

Man erkennt, dass man zunächst eine gewünschte Tabelle mit einer o++o-Anweisung erzeugen kann und dass man dann die Farben den entsprechenden Säulen am Ende des Programms zuordnen kann. Damit bleiben inhaltliche und Formfragen getrennt.

Für Säulendiagramme gelten die folgenden Regeln:

1. TEXTE werden vom System in Worte umgewandelt, indem jedes Leerzeichen durch den Unterstrich ersetzt wird.
2. Jedem numerischen Wert (ZAHL, PZAHL, RATIO) der darzustellenden Tabelle entspricht eine Säule.
3. Jedes Wort wird als Unterschrift einer Säule genutzt, wobei die Säule leer sein kann (Höhe null).
4. Jedem numerischen Wert muss ein RGB-Wert unmittelbar vorangestellt werden.
5. In jedem Segment wird lediglich die erste Spalte als Unterschrift berücksichtigt.

Dass manchmal Zahlen auch als Unterschrift dienen sollen, ist kein Problem für o++o, wie das folgende Beispiel zeigt:

```
X1 := 1 ... 10!0.1
```

```
SINUS := X sin
```

```
WURZEL:= X sqrt
```

```
X ::=X wort
```

```
RGBROT :=red leftat SINUS
```

```
RGBGRUEN:=green leftat WURZEL
```

Durch das Update in der vierten Zeile wird jeder X-Wert formal in ein Wort umgewandelt. Dadurch kann man jedoch mit X-Werten nicht weiter rechnen.