

Operationen im Detail

+, -, *, :, -+, gib, keys, keyslike, meta, onrs und verti

(Stand: 17.08.2021)

Die + Operation

Hinter dem + Symbol verbergen sich mehrere Operationen. Im Einzelnen sind dies die Zahlenadditionen, das Verketteten von Texten (Konkatenation), die Addition von Tabment und Einzelwert, die Addition von Tabment und numerischem Tupel mit gleicher Spaltenanzahl sowie die Addition von Tabellen mit gleicher Struktur. Es gilt stets:

Der TT (Tabment Typ) des Ergebnisses stimmt mit der TT des ersten Inputwerts überein.

Zahlenadditionen

ZAHL: Addition von big_int: $1+2=3$

PZAHL: Addition von float: $1.2 + 2.1 = 3.3$

RATIO: Addition rationaler Zahlen: $1/3 + 1/4 = 7/12$

gemischte Zahlenadditionen:

$ZAHL + PZAHL = PZAHL + ZAHL = PZAHL$

$ZAHL + RATIO = RATIO + ZAHL = RATIO$

$PZAHL + RATIO = RATIO + PZAHL = PZAHL$

Textverkettungen:

$Auto + mobil = Automobil$

gemischte Textverkettungen

$WORT + TEXT = TEXT + WORT = TEXT$

Addition von Tabment und Einzelwert

<TAB!

X, Y1 m

1 2

3

4 5

!TAB>

+2

ergibt:

X, Y1 m

3 4

5

6 7

```

<TAB!
X, Y1 m
1 2
  3
Ha 5
!TAB>
+ 110
ergibt:
X,      Y1  m
  1 2
    3
Hallo 5

```

Hierbei gilt:
numerischer Wert + Textwert = numerischer Wert
Textwert + numerischer Wert = Textwert

Addition von Tabelle und elementarem Tupel

```

<TAB!
X, Y1 m
1 2
  3
Ha 5
!TAB>
+ (110, 3)
ergibt:
X,      Y1  m
  1 5
    6
Hallo 8

```

Man erkennt, dass der erste Wert des Tupels zu jedem Wert der ersten Spalte und der zweite Wert zu jedem Wert der zweiten Spalte hinzugefügt wird.

Addition von Tabellen gleicher Struktur:

```

<TAB!
X, Y1 m
1 2
  3
Ha 5
!TAB>
+ <TAB!
X, Y1 m
1 2
  3
Ha 5
!TAB>
ergibt:
X,      Y1  m
  2 4

```

Hierbei wird zu jedem Wert der i-ten Spalte und der j-ten Zeile der Wert der zweiten Tabelle der i-ten Spalte und j-ten Zeile addiert. Wenn an einer Stelle verschiedene Zeilen- oder Spaltenanzahlen existieren, ist die + Operation nicht definiert.

Eigenschaft der + Operation:

Der Typ (TT, Schema) des Ergebnisses stimmt mit dem Typ des ersten Inputwertes überein (bis auf die elementare Typen (aus ZAHL könnte beispielsweise PZAHL werden).

Die - Operation

Die Operation - unterscheidet sich von der + Operation nur in 2 Punkten. Erstens wird die Addition von Zahlen stets durch die entsprechende Subtraktion von Zahlen ersetzt und zweitens bedeutet das „Subtrahieren“ von Texten, dass der zweite Inputwert in jedem TEXT- oder WORT-Wert eliminiert wird.

<TAB!

```
1 Heute ist Monntag.
2 Morgen ist Dienstag.
```

!TAB>

- "nn"

ergibt

X, Y 1

```
1 Heute ist Montag.
2 Morgen ist Dienstag.
```

Die * Operation

Gegenüber der + Operation muss man hier nur die Zahlenaddition durch die Zahlenmultiplikation ersetzen und die Textaddition durch die Operation ersetzen, die den ersten Inputwert unverändert lässt. Eine Tabelle vom Typ A_1, A_2, \dots, A_n 1 ist eine Matrix. Haben 2 Matrixtabellen die gleiche Anzahl von Spalten und Zeilen, so stimmt die + Operation mit der Matrixaddition überein. Das gilt nicht für die Multiplikation. Bekanntlich wird bei der Matrixmultiplikation nicht einfach zellenweise multipliziert. Daher stellt o++o eine weitere Operation *mat für die Matrixmultiplikation bereit.

Die : Operation

Hier gilt das für die * Operation Gesagte. Ferner bewirkt eine einzige Division durch Null nicht die Ungültigkeit der Gesamtoperation. Das Ergebnis enthält dann einen leeren optionalen Wert.

<TAB!

X, Y 1

```
1 Heute ist Sonntag.
2 Morgen ist Montag.
```

!TAB>

Z:= 3 : (X-2)

ergibt:
X, Y, Z? 1
1 Heute ist Sonntag. -3.
2 Morgen ist Montag.

Die -+ Operation

-+ ist eine Operation mit 3 Inputwerten. Der TT des ersten Inputwerts bleibt bestehen. Jedes Vorkommen des zweiten Inputwerts wird durch den dritten ersetzt.

```
<TAB!  
X, Y 1  
1 Heute ist Sonntag.  
2 Gestern ist Samstag.  
!TAB>  
-+ "n ist" ! "n war"  
ergibt:  
X, Y 1  
1 Heute ist Sonntag.  
2 Gestern war Samstag.
```

gib

Für die Beschreibung von gib benutzen wir Tabmente mit den Schemen:

```
t1! s1 = X1,X2,X3,(Y1,Y2 m) m
```

und

```
t2! s2 = X1,X2,X3,X4,(Y1,Y2,Z1,(W1,W2 m) m) m
```

Mit der gib-Anweisung können folgende Aufgaben realisiert werden:

Sortierung, Umstrukturierung, Vereinigung, Aggregation, Duplikateelimination, Taggen

1. Sortierung

```
t1
```

```
gib X2,X3,X1 m
```

sortiert zuerst nach X2, dann nach X3 und dann nach X1

```
t1
```

```
gib X2,Y2m m
```

sortiert die äußere Kollektion nach X2 und die innere Kollektion nach Y2; die innere Kollektion enthält keine Duplikate und die äußere Kollektion enthält jeden X2 Wert nur einmal.

2. Umstrukturierung

```
t1
```

```
gib X2,X3m m
```

Die X-Segmente werden nacheinander zuerst in die X2-Ebene und dann in die X3-Ebene eingefügt. Dadurch enthält die äußere Menge jeden X2-Wert nur einmal und jede innere Menge jeden X3-Wert nur einmal. Die Y-Segmente bleiben bei dieser Umstrukturierung unberührt.

```
t1
```

```
gib Y2,X1m m
```

Da die X-Segmente Y2 nicht enthalten, können sie nicht in die Zielstruktur eingefügt werden. Dagegen können die verlängerten Y-Segmente (X1, X2, X3, Y1, Y2) in beide Ebenen eingefügt werden. Die untergeordneten Y2-Werte werden somit den X1-Werte übergeordnet.

t2

gib Y2, Zm, W2m m

Da die X-Segmente nicht eingefügt werden können, erfolgt dieses erst durch die verlängerten Y-Segmente (X1, X2, X3, Y1, Y2). Für jedes solche Y-Segment werden Z-Elemente und anschließend die W-Segmente eingefügt. Keine Kollektion des Resultats enthält Duplikate.

t2

gib Y2, (Z, W2 m) m

Da die verlängerten Z-Segmente W2 nicht enthalten und die verlängerten W-Segmente nicht Z, bleibt jede innere (Z, W2 m) Kollektion leer. Alle Y2-Werte erscheinen aber im Ergebnis; jedoch ohne Duplikate.

3. Vereinigung

t1, t2

gib X2b

Hier werden zunächst alle X-Segmente von t1 eingefügt und anschließend alle X-Segmente von t2. Es werden hierbei keine Duplikate eliminiert. D.h., die „gewöhnliche“ mengentheoretische Vereinigung ergibt sich dann mit der Anweisung

t1, t2

gib X2m.

4. Taggen

Manchmal ist es sinnvoll zusätzliche Tags einzufügen oder Tags an völlig neuen Stellen zu verwenden. Z.B., wenn man Daten aus Tabellenkalkulationen bequem weiterverarbeiten will.

t: ID, LAND, BREITE, LAENGE, MEERESHÖHE?, JAN, FEB, MRZ, APR, MAI, JUN, JUL, AUG, SEP, OKT, NOV, DEZ 1

aus t

gib ID, LAND, BREITE, LAENGE, MEERESHÖHE?, MONATE 1

MONATE! JAN, FEB, MRZ, APR, MAI, JUN, JUL, AUG, SEP, OKT, NOV, DEZ

Hierbei werden die Monate jeweils durch den Tag MONATE eingeschlossen. Durch sie können bestimmte Formulierungen verkürzt werden. Z.B.:

gib LAND, (ID, MONATE m) m

keys

keys ist eine Selektionsoperation, d.h., der Typ der Inputtabelle wird durch keys nicht verändert.

keys liefert gegenüber avec keine neue Ausdruckskraft. Die Selektion ist in der Regel nur effizienter. Das liegt daran, dass für eine Selektion nicht die gesamte Tabelle durchforstet werden muss. Es wird bei jeder Menge (Liste) direkt auf den gegebenen Schlüssel (das n-te Element) zugegriffen. Da für Selektion die Spaltennamen der gegebenen Werte bedeutungslos sind, reicht es nackte Werte vorzugeben. Weiterhin ist es nicht erforderlich, dass die Kollektionstypen der Schlüsseltable mit denen der Ausgangstabelle übereinstimmen.

t1

keys [k1 k2 k3]

angewandt auf die Tabelle t1 mit dem Schema $s1=(X1, X2, X3, X4, (Y1, Y2 m)m)$ ergibt eine maximal 3-elementige Tabelle vom Typ $X1, X2, X3, X4, (Y1, Y2 m)m$ mit den X1-Werten k1, k2 und k3. Ist X1, X2 der Schlüssel, so wird jeweils ein Element ausgewählt, das den X1-Wert k1 besitzt. In einem solchen Fall sollte man für jedes Element den gesamten Schlüssel vorgeben:

keys <! [k11, k12 k21, k22 k31, k32] !>

Bei der komplexeren Tabelle t2 mit $s2=(X1, X2, X3, X4, (Y1, Y2, Z1, (W1, W2 m) m) m)$, kann auch auf innere Elemente direkt zugegriffen werden. Bei Listen wird eine natürliche Zahl x

vorgegeben und dann wird das x-te Element ausgegeben. Man kann auch auf innere Elemente direkt zugreifen, wenn der Schlüssel der vorangehenden Kollektion nicht gegeben ist:

```
t2
keys <! [kx1, kx2, [ky1, ["*"], [kw1]]] !>
```

Hier wird das Element der s2-Kollektion mit dem Schlüssel kx1, kx2 und dem untergeordneten Schlüssel ky1 selektiert. Das ky1-Element enthält die vollständige Z1-Kollektion, sowie die (W1, W2 m)-Kollektion mit dem (W1, W2)-Element mit dem Schlüssel kw1.

Es wird stets gefordert, dass die Schlüsselstruktur nach keys mit dem Anfang der Schlüsselstruktur der gegebenen Tabelle übereinstimmt. Den Werten aus dem 2. Argument von keys müssen einfache Felder der betrachteten Tabelle entsprechen. Damit wäre beispielsweise

```
keys 1234, Berlin
```

unzulässig, wenn die gegebene Tabelle den Typ (STID, ORT?, ... m) besitzt. Die Überspezifizierung des Schlüssels STID ist dagegen erlaubt, wenn das Schema (STID, ORT, ... m) lautet. Das Gleiche trifft auf keyslike (siehe unten) zu.

keyslike

keyslike basiert auf keys und der like Funktion, unterscheidet sich deswegen auch in einigen Punkten von keys.

Wird ein Schlüssel, der aus mehr als einem Feld besteht, unvollständig angegeben, so werden alle Tupel bzw. Subtupel ausgegeben, die den Teilschlüssel besitzen.

Beispiel:

```
gegeben: tab! K1, K2, X, Y m
```

```
tab
```

```
keys k1
```

bestimmt ein 4-Tupel mit dem Teilschlüssel K1.

```
tab
```

```
keyslike k1
```

ermittelt alle 4-Tupel mit dem Teilschlüssel K1.

Ansonsten können die ausdrucksstärkeren aber ineffizienteren Möglichkeiten von like einbezogen werden.

```
tab
```

```
keyslike "?enec*", "Achim"
```

bestimmt alle Personen mit Vornamen Achim und einem Namen, dessen 2,3,4 und 5. Buchstabe das Teilwort enec ergibt. Hier wurde K1=NAME und K2=VORNAME angenommen.

meta

Die Umkehroperation zu verti ist meta. Hier ist als zweiter Inputwert der elementare Tag, dessen Werte Spaltennamen werden sollen, anzugeben.

```
meinenoten2.tabh
```

```
meta FACH
```

ergibt die folgende tabh-Tabelle:

NAME,	ORT,	KLASSE?,	MA1,	DEU1	1
Clara	Oehna	4	1 2	3 1 1 1	1
Claudia	Dallgow				
Sophia	Dallgow	2	1 2	1 2 1	1 1

Die beiden Fächerlisten für Claudia sind leer.

```
klimate_strahlung1.tab
verti MONAT,STRAHLUNG 1 :=JAN .. DEZ
meta MONAT
```

ergibt die Ausgangstabelle mit folgender TTD:

```
<META!
TABMENT! KLIMATE_STRAHLUNG1
KLIMATE_STRAHLUNG1!
(ID, LAND, BREITE, LAENGE, MEERESHOEHE?, JAN, FEB, MRZ, APR, MAI, JUN, JUL, AUG, SEP, OKT, NOV, DEZ 1)
ID LAND! TEXT
APR AUG BREITE DEZ FEB JAN JUL JUN LAENGE MAI MEERESHOEHE MRZ NOV OKT SEP! PZAHL
!META>
```

onrs

Die Operation `onrs` wurde eingeführt, um `o++o`-Nummern für die Lösung von Stücklistenproblemen bereitzustellen. Das gegebene Tabment muss vom Typ `X1,...,Xn, (Y1, ... Yk m) m`, wobei `X1` und `Y1` die Schlüssel der jeweiligen Kollektionen sind. Wir setzen voraus, dass beide Kollektionen Mengen sind. Dadurch haben wir im Arbeitsspeicher direkten Zugriff auf die entsprechenden Tupel bzw. Subtupel. Wir betrachten ein Beispiel:

```
<TAB!
OT,          EIGENSCHAFT, (UT,          ANZ m) m
Buchse       zylindrisch
Felge       glatt
Polo         modern      Rad           4
                                   Motor          1
                                   Karosse         1
Golf         schnell     Rad           4
                                   Karosse         1
                                   Klimaanlage     1
                                   Motor           1
Kolben       leicht     KolbRing      2
                                   Buchse          1
Motor        schwer     Kolben        6
                                   Schraube        8
Rad          rund       Schraube      5
                                   Reifen          1
                                   Felge          1
!TAB>
onrs OTTONR ! Golf
```

Ergebnis:

```
OT, EIGENSCHAFT, (OTTONR, UT, ANZ m) 1
Golf schnell      1      Rad      4
                  1.1    Schraube  5
                  1.2    Reifen    1
                  1.3    Felge     1
```

2	Motor	1
2.1	Schraube	8
2.2	Kolben	6
2.2.1	KolbRing	2
2.2.2	Buchse	1
3	Klimaanl	1
4	Karosse	1

Man erkennt, dass allen direkten Unterteilen vom Golf eine otto-Nummer, die nur aus einer Zahl besteht, zugeordnet wird. Der Motor ist ein solches Teil. Die direkten Unterteile vom Motor (Schraube und Kolben) erhalten otto-Nummern mit zwei Zahlen. Analog erhalten die direkten Unterteile vom Kolben otto-Nummern mit 3 Zahlen. Für den Golf wird also eine nichtrekursive Menge ohne Redundanz gebildet. Auf diese Menge könnte jetzt die otto-Rekursion angewandt werden, um die Vielfachheit des Enthaltenseins eines Unterteils zu berechnen.

Neben dem Inputtabment benötigt onrs noch den Namen der ONR-Spalte (hier OTTONR) und ein oder mehrere Teile (hier nur Golf) für die die ONR-Auflösung vorgenommen werden soll. Demnach ist auch die Programmzeile

```
onrs STRUK_NR ! [Polo Motor]
```

korrekt und sinnvoll.

verti

Mit der Operation `verti` können bestimmte Tupel in Listen von Paaren umgewandelt werden. Das hat den Vorteil, dass man auf diese Listen Listenoperationen, wie z.B. das Selektieren anwenden kann. Dabei werden die Metadaten in Primärdaten überführt und es wird ein neues Schema in das bestehende eingefügt. Die folgende Datei wurde aus einer EXCEL-Tabelle gewonnen.

`klimate_strahlung1.tab` hat das Schema:

```
ID, LAND, BREITE, LAENGE, MEERESHOEHE?, JAN, FEB, MRZ, APR, MAI, JUN, JUL, AUG, SEP, OKT, NOV, DEZ 1
```

Sie enthält 17 Spalten. Mittels `verti` kann man die Spaltenzahl in folgender Weise auf 7 reduzieren:

```
aus klimate_strahlung1.tab
```

```
verti MON, STRAHLUNG 1:= JAN ..DEZ
```

Es entsteht aus dieser flachen Tabelle eine strukturierte, in der die Strahlungen vertikal angeordnet sind und die Monate in einer zusätzlichen Spalte ausgegeben werden:

ID,	LAND,	BREITE,	LAENGE,	MEERESHOEHE?,	(MON,	STRAHLUNG	1) 1
BG0001a-Varna	Bulgaria	43.21	27.91	44.	Jan	63.	
					Feb	68.	
					Mrz	81.	
					Apr	87.	
					Mai	88.	
					Jun	81.	
					Jul	86.	
					Aug	100.	
					Sep	95.	
					Okt	88.	
					Nov	66.	
					Dez	59.	
BG0002a-Shumen	Bulgaria	43.283	26.933	242.	Jan	59.	
					Feb	68.	
					Mrz	83.	
					Apr	88.	
					Mai	88.	
					Jun	81.	
					Jul	88.	

. . .

Jetzt sei eine Tabelle enkel.tabh gegeben:

NAME,	ORT,	GEBDAT,	KLASSE?,	(HOBBY,	STUNDEN	1),	MA1,	DEU1	1
Clara	Oehna	12.6.11	4	Reiten	5		1 2	3 1	1 1
				Schach	1		1		
Claudia	Dallgow	14.9.17		Chinesisch	5				
				Essen	4				
Sophia	Dallgow	7.9.13	2	Malen	5		1 2	1 2	1
				Radschlagen	4		1 1		
				Chinesisch	6				

Dann können diese Noten mit entsprechenden Fächern vertikal angeordnet werden:

enkel.tabh

verti FACH,NOTE1 1:= MA1 ..DEU1

ergibt bei tabh-Ausgabe:

NAME,	ORT,	GEBDAT,	KLASSE?,	(HOBBY,	STUNDEN	1),	(FACH,	NOTE1	1) 1
Clara	Oehna	12.6.11	4	Reiten	5		Ma	1 2 1	
				Schach	1		Deu	3 1 1 1	
Claudia	Dallgow	14.9.17		Chinesisch	5				
				Essen	4				
Sophia	Dallgow	7.9.13	2	Malen	5		Ma	1 2 1 1	
				Radschlagen	4		Deu	1 2 1	
				Chinesisch	6				

meinenoten2.tabh