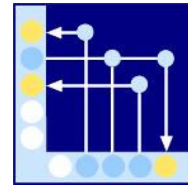




Hochschule Aalen

*Fakultät Elektronik und Informatik
Studienbereich Informatik*



Programmieren in MOSTflexiPL

Vorlesung im Sommersemester 2026
Prof. Dr. habil. Christian Heinlein

10. Aufgabenblatt (16. Juni 2026)

Aufgabe 17: Zeichenketten

Definieren Sie einen Typ `str` zur Repräsentation von Zeichenketten sowie folgende Operatoren:

- Für beliebig viele Zeichen `c ...` mit Typ `char` liefert `str c ...` eine Zeichenkette mit diesen Zeichen.
(Dieser Operator wird gemäß §4.5 implizit an Zeichenkettenlitterale übergeben.)

- Für eine Zeichenkette `s` mit Typ `str` liefert `#s` ihre Länge, d. h. die Anzahl ihrer Zeichen.

Wenn `s` weder durch ein Zeichenkettenliteral (d. h. indirekt durch den zuvor genannten Operator) noch durch einen der nachfolgend beschriebenen Operatoren erzeugt wurde, liefert `#s` den Wert 0.

Dieser Operator besitzt den gleichen Vorrang wie die vordefinierte Vorzeichenabfrage.

- Für eine Zeichenkette `s` mit Typ `str` und einen `int`-Wert `i` liefert `s[i]` das `i`-te Zeichen von `s`, sofern `i` zwischen 1 und `#s` (jeweils einschließlich) liegt. Andernfalls wird `nil` geliefert.

Dieser Operator besitzt ebenfalls den gleichen Vorrang wie die vordefinierte Vorzeichenabfrage.

- Für eine Zeichenkette `s` mit Typ `str` wertet `for name in s do ... end` den Schleifenrumpf `...` nacheinander für jedes Zeichen von `s` aus, wobei das aktuelle Zeichen im Schleifenrumpf jeweils als Konstante mit dem Namen `name` verfügbar ist.

Als Resultatwert wird die Anzahl der Schleifendurchläufe geliefert, d. h. die Länge von `s`.

- Für eine Zeichenkette `s` mit Typ `str` gibt `print s` bzw. `print only s` die Zeichenkette `s` mit bzw. ohne abschließenden Zeilentrenner aus.

- Für einen `int`-Wert `i` liefert `str i` die Zeichenkettendarstellung von `i` als Dezimalzahl, die bei einem negativen Wert mit einem Minuszeichen beginnt.

Für einen unnatürlichen Wert `i` wird eine leere Zeichenkette geliefert.

- Für Zeichenketten `s1` und `s2` liefert `s1 + s2` die Verkettung von `s1` und `s2` als neue Zeichenkette.

Für eine Zeichenkette `s` mit Typ `str` und einen Wert `x` eines prinzipiell beliebigen Typs `X` liefert `s + x` (bzw. `x + s`) die Verkettung von `s` mit der Zeichenkettendarstellung von `x` (bzw. umgekehrt), die durch Anwendung eines implizit übergebenen Operators `str (X) -> (str)` auf `x` ermittelt wird.

Diese Operatoren besitzen die gleichen Bindungseigenschaften wie die vordefinierte Addition.

Aufgabe 18: Matrizen (Bonusaufgabe)

Teilaufgabe 18.a)

Definieren Sie einen Operator $\bullet\{\bullet x \bullet\}$, sodass $T\{M \times N\}$ (z. B. $\text{int}\{3 \times 4\}$ oder $\text{rat}\{2 \times 3\}$) für jede Kombination eines Typs T und zweier int -Werte M und N jeweils einen eindeutigen neuen Typ liefert, der zur Repräsentation von Matrizen mit M Zeilen, N Spalten und Elementen des Typs T dient.

Teilaufgabe 18.b)

Definieren Sie einen Operator zur Konstruktion von Matrizen, der z. B. wie folgt verwendet werden kann:

```
$$ Definition einer Matrix A mit 3 Zeilen, 4 Spalten und Elementen
$$ des Typs int.
$$ Dementsprechend müssen in den runden Klammern genau 3x4 int-Werte
$$ als Elemente der Matrix stehen (je 4 Werte für jede der 3 Zeilen).
A := {3x4}(11 12 13 14   21 22 23 24   31 32 33 34);

## Definition einer Matrix B mit 2 Zeilen, 3 Spalten und Elementen
## des Typs rat.
B := {2x3}(1/1 1/2 1/3   2/1 2/2 2/3)
```

Der Elementtyp T der Matrix wird jeweils aus den angegebenen Elementen deduziert. Die Dimensionen M und N müssen jedoch explizit angegeben werden, weil sie nicht aus der Anzahl der angegebenen Elemente deduziert werden können, selbst wenn es noch ein Trennzeichen zwischen den einzelnen Zeilen der Matrix gäbe. Wenn die Anzahl der angegebenen Elemente dann nicht mit dem Produkt von M und N übereinstimmt, soll der Resultatwert nil sein, zum Beispiel:

```
# C und D sind jeweils nil, weil die Anzahl der Elemente in den runden
# Klammern jeweils verschieden von 2x2 ist.
C := {2x2}(1 2 3);
D := {2x2}(1 2 3 4 5)
```

Zur Speicherung und zum Zugriff auf die Elemente einer Matrix können ähnliche Operatoren wie in §4.4 verwendet werden, die zu jeder Kombination einer Matrix A und zweier int -Werte i und j jeweils eine neue eindeutige Variable mit passendem Inhaltstyp bzw. den Wert dieser Variablen liefern.

Teilaufgabe 18.c)

Definieren Sie einen generischen Operator `print`, der die Elemente einer Matrix zeilenweise ausgibt, sofern es einen Operator `print only` $\bullet$ zur Ausgabe der einzelnen Elemente gibt.

Teilaufgabe 18.d)

Definieren Sie generische Operatoren $\bullet+\bullet$ und $\bullet*\bullet$ (mit denselben Bindungseigenschaften wie die entsprechenden vordefinierten Operatoren für int -Werte) zur Addition und Multiplikation von „kompatiblen“ Matrizen, die als implizite Parameter Operatoren $\bullet+\bullet$ und ggf. $\bullet*\bullet$ für den Elementtyp der Matrix erhalten.