

## 6. Tupel und Verbunde

**Bisher:** Wertbereiche       $\text{int, char, ... , } t \rightarrow t'$

Typkonstruktor  
 $\downarrow$   
 Funktionstyp

**Jetzt:** zwei Möglichkeiten, strukturierte Werte aus Komponenten zu konstruieren

### 6.1 Tupel

Tupel sind endliche, geordnete Folgen von Werten; entsprechen n-Tupel aus einem kartesischen Produkt.

$(v_1, \dots, v_n)$  hat den Typ  $\underbrace{typ_1 * \dots * typ_n}_{\text{Tupeltyp}}$ , wenn  $v_i$  den Typ  $typ_i$  hat.  $(1 \leq i \leq n)$

#### **Tupelausdruck:**

$(e_1, \dots, e_n)$  Auswertung  $e_1, \dots, e_n$  in dieser Reihenfolge zu  $v_1, \dots, v_n$

Wert des Tupelausdrucks ist  $(v_1, \dots, v_n)$

**Gleichheit** auf Tupeltypen:  $(v_1, \dots, v_n) = (v_1', \dots, v_n')$  genau dann wenn  $v_i = v_i'$  für alle  $(1 \leq i \leq n)$

**Projektion** #i  $(v_1, \dots, v_n) = v_i$  wenn  $(1 \leq i \leq n)$

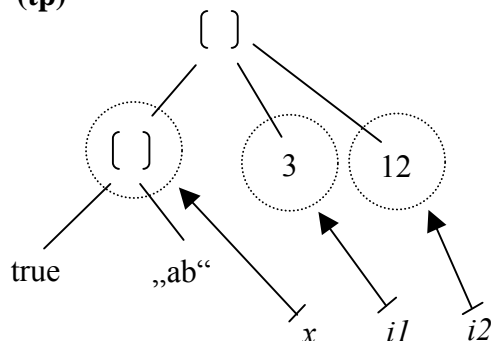
#### **Muster** (Pattern):

Mit Mustern kann man komfortabel, und zwar durch Namen, auf die Komponenten strukturierter Werte zugreifen (Tupel, Verbunde, Listen...).

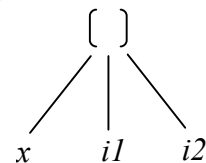
Ein Muster ist ein Ausdruck mit **Mustervariablen**.

**Musterabgleich** (Pattern Matching) untersucht, ob ein Muster den Wert „trifft“.

(tp)



(m)



allgemeine Form der Wertbindung:

val pat : typ = e

**Musterabgleich:**

1.  $e$  wird ausgewertet zu  $v$
2. Musterabgleich zwischen  $pat$  und  $v$

Ergebnis ist eine Wertumgebung für die Mustervariablen

Mustervariablen haben ein definierendes Vorkommen, dessen Gültigkeitsbereich am Ende der Deklaration beginnt. (später anders, bei val rec und bei fun)

**Muster haben Typen:**

1. Konstantes Muster  $c$  hat den Typ von  $c$
2. Typ eines variablen Musters  $x : t$  hat den Typ  $t$
3. Wildcard “\_” hat beliebigen Typ
4. Tupelmuster  $(p_1, \dots, p_n)$  hat den Typ  $t_1 * \dots * t_n$ , wenn  $p_i$  den Typ  $t_i$  hat  $(1 \leq i \leq n)$

Verallgemeinert: Wertbindung val  $p : t = e$  heißt **wohlgetypt**, wenn  $p$  und  $e$  den Typ  $t$  haben.

**Zusammenfassung:****Muster**

- sind „Ausdrücke mit Variablen“
- beschreiben Mengen von Werten
- passen auf die von ihnen beschriebenen Werte
- **Musterabgleich** stellt fest, ob ein gegebenes Muster  $p$  auf einen gegebenen Wert  $v$  passt
  - Wenn  $p$  auf  $v$  passt, dann ist das Ergebnis des Musterabgleichs die Bindung der in  $p$  auftretenden Variablen (Wertumgebung).
  - Ansonsten gibt es eine Fehlermeldung, semantisch ergibt sich der Wert  $\perp$  (fail)
- Musterabgleich zerlegt  $p$  und Wert  $v$  in Teilmuster  $p_1, \dots, p_n$  bzw. in Komponenten  $v_1, \dots, v_n$  und macht Musterabgleich zwischen  $p_1$  und  $v_1$ , ...,  $p_n$  und  $v_n$   
Rekursion terminiert an jedem Blatt von  $p$  oder bei fail.

## 6.2 Verbunde (Records)

Zusammenfassung von einer festen Zahl von Werten mit **Selektoren** (labels)

Zugriff auf die Komponenten mittels der Selektoren

**Verbundtyp:**  $\{lab_1:typ_1, lab_2:typ_2, \dots, lab_n:typ_n\}$  mit  $n \geq 0$ ,  $lab_i \neq lab_j$  für  $i \neq j$

**Verbundwert:**  $\{lab_1 = v_1, \dots, lab_n = v_n\}$  mit  $v_i$  hat den Typ  $typ_i$

**Beispiel:**

type inttelnr = {land:int, amt:int, anschluss:int}

val praes = {land=49, anschluss=3021000, amt = 681}

**Verbundmuster**  $\{lab_1=p_1, \dots, lab_n=p_n\}$  mit  $p_i$  hat Typ  $t_i$

**Musterabgleich mit Verbundmustern:**

Rekursion über gleiche Selektoren in Mustern und Werten

$$p = \{\text{lab}_1=p_1, \dots, \text{lab}_n=p_n\} \quad v = \{\text{lab}_1=v_1, \dots, \text{lab}_n=v_n\}$$

rekursiver Musterabgleich zwischen  $p_1$  und  $v_1, \dots, p_n$  und  $v_n$

**Wildcards in Verbundmustern:**

- anonyme Variable  $\_$  für eine Komponente
- $\dots$  für Folgen von nicht vorkommenden Komponenten

Abkürzung:  $\{\text{lab}_1, \dots, \text{lab}_n\}$  statt  $\{\text{lab}_1=\text{lab}_1, \dots, \text{lab}_n=\text{lab}_n\}$

## 6.3 Syntax, Typregeln und Semantik von Tupeln, Verbunden und Mustern

### 6.3.1 Syntax

Siehe Skript Jörg Bauer:

<http://rw4.cs.uni-sb.de/~joba/Info1/Material/semantik.pdf>

### 6.3.2 Typberechnung

Siehe Skript Jörg Bauer:

<http://rw4.cs.uni-sb.de/~joba/Info1/Material/semantik.pdf>

**Bemerkung zum Musterabgleich:**

Wann kann der Musterabgleich fehlschlagen?

- bei Typberechnung: Typ von Muster  $\neq$  Typ von Wert
- bei der Auswertung: Muster passt nicht auf den Wert

**Typberechnung:**

$$[\text{valdecl}] \quad \frac{\Gamma \vdash e : t \quad \wedge \quad \Gamma \vdash p \succeq (\Gamma', t)}{\Gamma \vdash \text{val } p : t = e \triangleright \Gamma \oplus \Gamma'}$$

Typüberprüfung: durch Mehrfachvorkommen von  $t$

Deckt z.B. den Fall ab: (Typfehler!)

$$p : t_1 * \dots * t_n \quad \text{und} \quad n \neq m \quad \text{oder} \quad t_i \neq t_i' \quad \text{für ein} \quad 1 \leq i \leq n$$

$$e : t_1' * \dots * t_m'$$

**Muster:** Berechnung von  $T_p$ 

$$[\text{wild}] \quad \frac{}{\Gamma \vdash \_ : t \sqsupseteq (\{\}, t)} \quad \leftarrow \text{irgendein Typ}$$

$$[\text{scon}] \quad \frac{}{\Gamma \vdash c \sqsupseteq (\{\}, \text{typ}(c))}$$

$$[\text{tuple1}] \quad \frac{}{\Gamma \vdash () \sqsupseteq (\{\}, \text{unit})}$$

$$[\text{tuple2}] \quad \frac{\Gamma \vdash p_1 \sqsupseteq (\Gamma_1, t_1) \quad \wedge \quad \dots \quad \wedge \quad \Gamma \vdash p_n \sqsupseteq (\Gamma_n, t_n)}{\Gamma \vdash (p_1 : t_1, \dots, p_n : t_n) \sqsupseteq (\Gamma_1 \oplus \dots \oplus \Gamma_n, (t_1 * \dots * t_n))} \quad n \geq 2$$

Bemerkung: implizite Typüberprüfung durch mehrfaches Auftreten der  $t_1, \dots, t_n$

**Ausdrücke:**

$$[\text{unit}] \quad \frac{}{\Gamma \vdash () : \text{unit}}$$

$$[\text{lab1}] \quad \frac{\Gamma \vdash e : t_1 * t_2 * \dots * t_n}{\Gamma \vdash \#l e : t_{[l]}} \quad \begin{array}{l} l \in \{1, 2, 3, \dots\} \\ 1 \leq [l] \leq n \end{array}$$

$$[\text{row}] \quad \frac{\Gamma \vdash e_1 : t_1 \quad \wedge \quad \dots \quad \wedge \quad \Gamma \vdash e_n : t_n}{\Gamma \vdash \{l_1 = e_1, \dots, l_n = e_n\} : \{l_1 : t_1, \dots, l_n : t_n\}}$$

### 6.3.3 Semantik

Siehe Skript Jörg Bauer:

<http://rw4.cs.uni-sb.de/~joba/Info1/Material/semantik.pdf>

**Neue semantische Relationen:**

$$W_p \subseteq WU \times \text{Wert} \times \text{pat} \times (WU \cup \{\perp\})$$

In der Wertumgebung  $\rho$  ergibt sich beim Musterabgleich zwischen Muster  $p$  und Wert  $v$  die neue Wertumgebung  $\rho'$  oder der Musterabgleich schlägt fehl ( $\perp$ ).

**Deklarationen:**

$$[\text{valdecl}] \quad \frac{\rho \vdash e \Downarrow v \quad \wedge \quad \rho, v \vdash p \nearrow \rho'}{\rho \vdash \text{val } p : t = e \Uparrow \rho \oplus \rho'}$$

**Musterabgleich:**

$$[\text{wild}] \quad \frac{}{\rho, v \vdash \_ \nearrow \{\}}$$

$$[\text{scon}] \quad \frac{}{\rho, c_1 \vdash c_2 \nearrow \{\}} \quad \llbracket c_2 \rrbracket = c_1$$

$$[\text{sconF}] \quad \frac{}{\rho, c_1 \vdash c_2 \nearrow \perp} \quad \llbracket c_2 \rrbracket \neq c_1$$

$$[\text{tuple2}] \quad \frac{\rho, v_1 \vdash p_1 \nearrow \rho_1 \quad \wedge \quad \cdots \quad \wedge \quad \rho, v_n \vdash p_n \nearrow \rho_n}{\rho, (v_1, \dots, v_n) \vdash (p_1, \dots, p_n) \nearrow \rho_1 \oplus \cdots \oplus \rho_n}$$