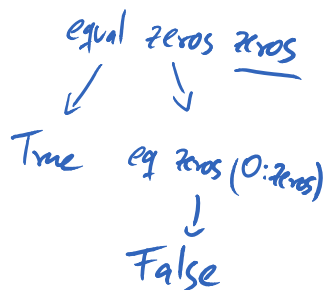


$f(0:zeros) [] = []$

Patterns $\hat{=}$

Ausdrücke aus
Datenkonstruktor
u. Variablen,
bei denen keine Var
mehrfach auftritt
("linear")

Grund: Sonst wäre
das Ergebnis abhängig
v. d. Auswertungsstra-
tegie:



Gebe bei jedem
Pattern an:

- auf welche Werte
passt der Pattern?
- Wie werden Var. des
Patterns instantiiert
beim Pattern Matching?

- Var

Bsp: square $x = x * x$
↑
Pattern

passt auf jeden Wert,
Var. wird mit dem
Wert des Arguments
instantiiert

square 5 =

$$5 * 5 =$$

$$25$$

- underscore
ist "Joker"-Pattern.
Passt auf alles,
aber es findet
keine Var-Instanz.
Statt.
Joker-Pattern darf
mehrfach links
vorkommen.
Mehrfache
Vorkommen dürfen
unterschiedl.
instantiiert werden.
= integer, ..., string

Diese Patterns passen
nur auf sich selbst.
Keine Inst. von
Var.

$$is_5 :: Int \rightarrow Bool$$

$$is_5 \ 5 = True$$

$$is_5 \ _ = False$$

- (constr pat₁ ... pat_n),
n ≥ 0

Wobei constr ein
n-stelliger Daten-
Konstruktor ist.
Passt auf alle Werte,
die mit constr
gebildet werden,
falls pat_i auf das

i -te Argument passt
(für alle $i \in \{1, \dots, n\}$).

Bsp: f überprüft, ob
Liste mit Zahl 5
beginnt

$f(5: _) = \text{True}$

$f(_) = \text{False}$

$= [\underline{pat}_1, \dots, \underline{pat}_n],$
 $n \geq 0$

passt auf alle Listen
der Länge n , falls
 \underline{pat}_i auf das i -te
Listenelement passt
(für alle $i \in \{1, \dots, n\}$).

$= (\underline{pat}_1, \dots, \underline{pat}_n), n \geq 0$

passt auf Tupel mit
 n Komponenten, falls
 \underline{pat}_i auf die i -te
Komponente passt
(für alle $i \in \{1, \dots, n\}$).