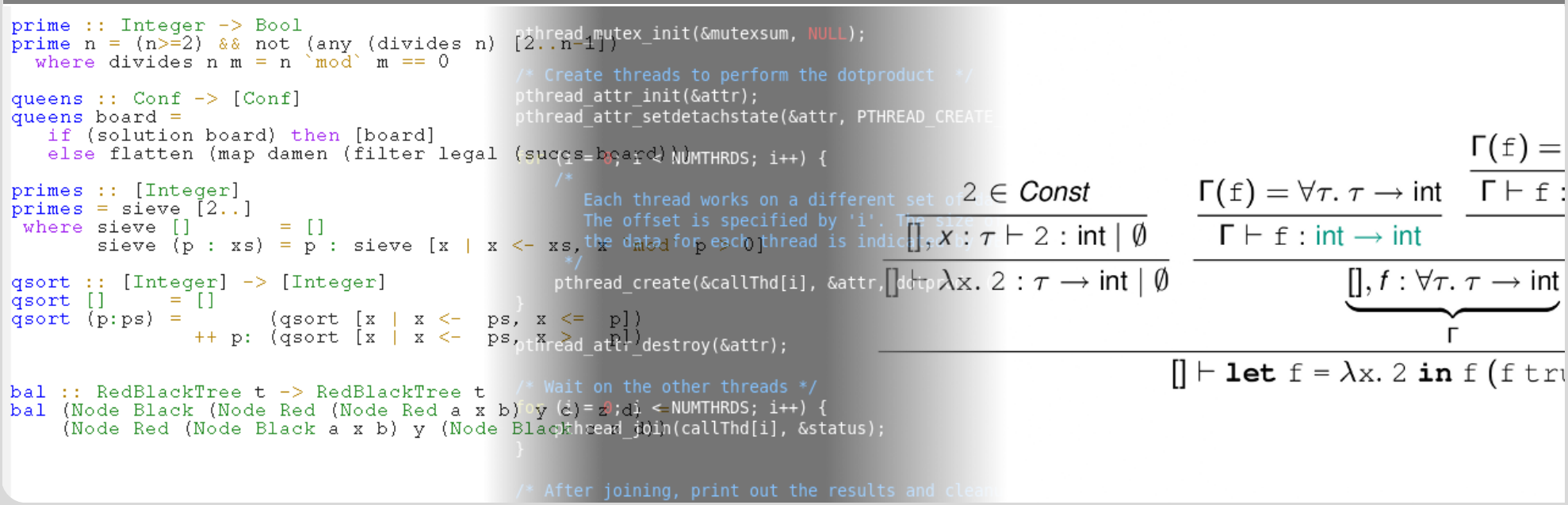
Karlsruher Institut für Technologie

Programmierparadigmen
Tutorium: Lazyness, Streams

Prof. Dr.-Ing. Gregor Snelting | WS 2012/2013

LEHRSTUHL PROGRAMMIERPARADIGMEN



Listen im λ-Kalkül

rekursive `data`-Typen: darstellbar im λ-Kalkül, z.B. Listen:

$$[1,2,3] \simeq \lambda \mathtt{c}.\ \lambda \mathtt{n}.\ c\ 1\ (c\ 2\ (c\ 3\ \mathtt{n}))$$

- möglicher Typ: $\tau_{\mathtt{list}(\alpha)} = (\alpha \to \beta \to \beta) \to \beta \to \beta$
  α: Elementtyp
- aber: Funktionstypen passen nicht
  $tail = \lambda l\ c\ n.\ l\ (\lambda x\ xs\ i.\ i\ x\ (xs\ c))\ (\lambda f.\ n)\ (\lambda x\ xs.\ xs)$

$\vdash tail : ((\alpha_2 \to (\alpha \to \alpha_3) \to (\alpha_2 \to \alpha_3 \to \alpha_4) \to \alpha_4) \to (\alpha_5 \to \alpha_1) \to (\alpha_6 \to \alpha_7 \to \alpha_7) \to \alpha_8) \to \alpha \to \alpha_1 \to \alpha_8$

- zumindest nicht: $\tau_{\mathtt{list}(\alpha)} \to \tau_{\mathtt{list}(\alpha)}$

Erweitere λ-Kalkül um Listen

- `Cons,Nil,head,tail,null`
- `null` $(\mathtt{Cons}\ t_1\ t_2) \Rightarrow \mathtt{False}$ $\quad \mathtt{null}\ \mathtt{Nil} \Rightarrow \mathtt{True}$
  `head` $(\mathtt{Cons}\ t_1\ t_2) \Rightarrow t_1$ $\quad \mathtt{tail}\ (\mathtt{Cons}\ t_1\ t_2) \Rightarrow t_2$
- `head Nil` $\not\Rightarrow$ (Laufzeitfehler) $\quad$ `head 42` (Typfehler)

Typregeln für Listen

Typ von `Cons`, `Nil`, `head`, `tail`, `null` abhängig von Elementtyp:

- erweiterte Typsyntax

$$\tau = \underbrace{\textit{Basistyp}}_{\text{int, bool}} \mid \underbrace{\textit{Var}}_{\alpha, \alpha_2, \beta, \gamma, \ldots} \mid \tau_1 \underbrace{\rightarrow}_{\text{2-stelliger Typkonstruktor}} \tau_2 \mid \underbrace{\text{list}\,(\tau)}_{\text{1-stelliger Typkonstruktor}}$$

- keine fixen Typen $\tau_{\texttt{Cons}}, \tau_{\texttt{head}}, \ldots \Rightarrow$ neue *Typregeln*

Typregeln für Listen

$$\textsc{Cons}\ \frac{}{\Gamma \vdash \texttt{Cons}\ t_1\ t_2 : \text{list}\,(\tau)} \qquad \textsc{Nil}\ \frac{}{\Gamma \vdash \texttt{Nil} : \text{list}\,(\tau)}$$

$$\textsc{head}\ \frac{}{\Gamma \vdash \texttt{head}\ t : \text{list}\,(\tau)} \qquad \textsc{tail}\ \frac{}{\Gamma \vdash \texttt{tail}\ t : \text{list}\,(\tau)}$$

$$\textsc{null}\ \frac{}{\Gamma \vdash \texttt{null}\ t : \text{bool}}$$

Typregeln für Listen

Typ von `Cons`, `Nil`, `head`, `tail`, `null` abhängig von Elementtyp:

- erweiterte Typsyntax

$$\tau = \underbrace{\mathit{Basistyp}}_{\text{int, bool}} \mid \underbrace{\mathit{Var}}_{\alpha, \alpha_2, \beta, \gamma, \ldots} \mid \underbrace{\tau_1 \to \tau_2}_{2\text{-stelliger Typkonstruktor}} \mid \underbrace{\text{list}\,(\tau)}_{1\text{-stelliger Typkonstruktor}}$$

- keine fixen Typen $\tau_{\texttt{Cons}}, \tau_{\texttt{head}}, \ldots \Rightarrow$ neue *Typregeln*

Typregeln für Listen

$$\textsc{Cons}\ \frac{\Gamma \vdash t_1 : \tau \qquad \Gamma \vdash t_2 : \text{list}\,(\tau)}{\Gamma \vdash \texttt{Cons}\ t_1\ t_2 : \text{list}\,(\tau)} \qquad \textsc{Nil}\ \frac{}{\Gamma \vdash \texttt{Nil} : \text{list}\,(\tau)}$$

$$\textsc{Head}\ \frac{\Gamma \vdash t : \text{list}\,(\tau)}{\Gamma \vdash \texttt{head}\ t : \text{list}\,(\tau)} \qquad \textsc{Tail}\ \frac{\Gamma \vdash t : \text{list}\,(\tau)}{\Gamma \vdash \texttt{tail}\ t : \text{list}\,(\tau)}$$

$$\textsc{Null}\ \frac{\Gamma \vdash t : \text{list}\,(\tau)}{\Gamma \vdash \texttt{null}\ t : \text{bool}}$$

Typregeln für Listen

Typ von `Cons,Nil,head,tail,null` abhängig von Elementtyp:

- erweiterte Typsyntax

$$\tau = \underbrace{\textit{Basistyp}}_{\text{int,bool}} \mid \underbrace{\textit{Var}}_{\alpha,\alpha_2,\beta,\gamma,\ldots} \mid \tau_1 \underbrace{\rightarrow}_{\text{2-stelliger Typkonstruktor}} \tau_2 \mid \underbrace{\mathsf{list}\,(\tau)}_{\text{1-stelliger Typkonstruktor}}$$

- keine fixen Typen $\tau_{\texttt{Cons}}, \tau_{\texttt{head}}, \ldots \Rightarrow$ erweiterte Typannahmen

$$\begin{aligned}\Gamma_{\texttt{list}} = \quad & \texttt{Cons} : \forall\alpha.\alpha \rightarrow \mathsf{list}\,(\alpha) \rightarrow \mathsf{list}\,(\alpha),\\ & \texttt{Nil} : \forall\alpha.\mathsf{list}\,(\alpha)\,,\\ & \texttt{head} : \forall\alpha.\mathsf{list}\,(\alpha) \rightarrow \mathsf{list}\,(\alpha),\\ & \texttt{tail} : \forall\alpha.\mathsf{list}\,(\alpha) \rightarrow \mathsf{list}\,(\alpha),\\ & \texttt{null} : \forall\alpha.\mathsf{list}\,(\alpha) \rightarrow \mathsf{bool}\end{aligned}$$

Funktionen in Haskell: polymorphe Definition erweitert Γ um Typschema

```
length :: [a] -> Int
length list = ...
```

$$\Gamma' = \Gamma, \texttt{length} : \forall\alpha.\mathsf{list}\,(\alpha) \rightarrow \mathsf{int}$$

Typinferenz für Listen

Wie zuvor: $[t_1, t_2, \ldots, t_n]$ für $\texttt{Cons}\ t_1\ (\ldots(\texttt{Cons}\ t_n\ \texttt{Nil})\ldots)$

$$\dfrac{\dfrac{\Gamma_{\texttt{list}} \vdash \texttt{\&\&} : \alpha_4 \qquad \dfrac{\Gamma_{\texttt{list}} \vdash \texttt{null} : \alpha_6 \qquad \Gamma_{\texttt{list}} \vdash [] : \alpha_7}{\Gamma_{\texttt{list}} \vdash \texttt{null}\ [] : \alpha_5}}{\Gamma_{\texttt{list}} \vdash \texttt{\&\&}\ (\texttt{null}\ []) : \alpha_3} \qquad \dfrac{\Gamma_{\texttt{list}} \vdash \texttt{null} : \alpha_8 \qquad \Gamma_{\texttt{list}} \vdash [1,2,3] : \alpha_9}{\Gamma_{\texttt{list}} \vdash \texttt{null}\ [1,2,3] : \alpha_4}}{\Gamma_{\texttt{list}} \vdash (\texttt{null}\ []) \texttt{\&\&} (\texttt{null}\ [1,2,3]) : \alpha_1}$$

$$\begin{aligned} C = \quad & \{\alpha_9 = \text{list}\,(\text{int}), \alpha_3 = \alpha_4 \to \alpha_1, \alpha_4 = \alpha_5 \to \alpha_3, \\ & \alpha_4 = \text{bool} \to (\text{bool} \to \text{bool}), \alpha_6 = \alpha_7 \to \alpha_5, \alpha_8 = \alpha_9 \to \alpha_4 \\ & \alpha_6 = ???? \\ & \alpha_8 = ???? \\ & \alpha_7 = ????\} \end{aligned}$$

$$\begin{aligned} \Gamma_{\texttt{list}} = \quad & \ldots \\ & \texttt{null} : \forall\alpha.\text{list}\,(\alpha) \to \text{bool}, \\ & \texttt{Nil} : \forall\alpha.\text{list}\,(\alpha) \\ & \ldots \end{aligned}$$

Typinferenz für Listen

Wie zuvor: $[t_1, t_2, \ldots, t_n]$ für $\texttt{Cons}\ t_1\ (\ldots(\texttt{Cons}\ t_n\ \texttt{Nil})\ldots)$

$$\dfrac{\dfrac{\Gamma_{\texttt{list}} \vdash \texttt{\&\&} : \alpha_4 \qquad \dfrac{\Gamma_{\texttt{list}} \vdash \texttt{null} : \alpha_6 \qquad \Gamma_{\texttt{list}} \vdash [] : \alpha_7}{\Gamma_{\texttt{list}} \vdash \texttt{null}\ [] : \alpha_5}}{\Gamma_{\texttt{list}} \vdash \texttt{\&\&}\ (\texttt{null}\ []) : \alpha_3} \qquad \dfrac{\Gamma_{\texttt{list}} \vdash \texttt{null} : \alpha_8 \qquad \Gamma_{\texttt{list}} \vdash [1,2,3] : \alpha_9}{\Gamma_{\texttt{list}} \vdash \texttt{null}\ [1,2,3] : \alpha_4}}{\Gamma_{\texttt{list}} \vdash (\texttt{null}\ []) \texttt{\&\&} (\texttt{null}\ [1,2,3]) : \alpha_1}$$

$$\begin{aligned} C = \quad & \{\alpha_9 = \text{list}\,(\text{int}), \alpha_3 = \alpha_4 \to \alpha_1, \alpha_4 = \alpha_5 \to \alpha_3, \\ & \alpha_4 = \text{bool} \to (\text{bool} \to \text{bool}), \alpha_6 = \alpha_7 \to \alpha_5, \alpha_8 = \alpha_9 \to \alpha_4 \\ & \alpha_6 = \text{list}\,(\beta_1) \to \text{bool} \\ & \alpha_8 = \text{list}\,(\beta_2) \to \text{bool} \\ & \alpha_7 = \text{list}\,(\beta_3)\} \end{aligned}$$

$$\begin{aligned} \Gamma_{\texttt{list}} = \quad & \ldots \\ & \texttt{null} : \forall\alpha.\text{list}\,(\alpha) \to \text{bool}, \\ & \texttt{Nil} : \forall\alpha.\text{list}\,(\alpha) \\ & \ldots \end{aligned}$$