# Natural Semantics

## Hauptseminar Programmiersprachentheorie

Andrea Langner

15.05.2007

# Inhalt

- Motivation
- Formalismus
- Mini-ML
- Categorial Abstract Machine (CAM)
- Übersetzung Mini-ML nach CAM
- Zusammenfassung

# Motivation

- Probleme mit rein denotationeller Definition
  - Prüfung auf Wohlgetyptheit nicht zufriedenstellend
    $[\![ true + 0 ]\!] = WRONG$
  - Overloadingresolution bei Typisierung
  - Rekursion nicht abstrakt
  - Backpatching
  - Darstellung von Nichtdeterminismus
  - Zu geringe Abstraktion

# Formalismus

- Judgment: Aussage die wahr oder falsch sein kann
  - Expr ⇒ Value
  - Expr : Type
  - Hyps ⊢ Prop
- Inferenzregel: 0..n Prämissen und eine Folgerung
- Notation

$$\frac{Prämisse[1] \ldots Prämisse[n]}{Folgerung}$$

Beispiele

$$\frac{x \Rightarrow 1 \quad 3 \Rightarrow 3}{x + 3 \Rightarrow 4}$$

$$\frac{\Gamma \vdash A \quad \Gamma, A \vdash B}{\Gamma \vdash B}$$

# Mini-ML

- λ-Kalkül
  mit Ganzzahlen, Wahrheitswerten und Rekursion
- Ausdrücke:
  number, true, false, ident, lambda, if, mlpair, apply, let, letrec
- Beispiele :
  - **let** *id* = λx.x **in** id 1
  - **letrec** *fact* = λx.**if** x = 0
    **then** 1 **else** x * fact(x - 1)
    **in** fact 4

# Mini-ML - Dynamische Semantik I

$$\rho \vdash \mathrm{E} \Rightarrow \alpha$$

- ρ = Umgebung
  - Liste von Paaren P → α
  - P Pattern
  - α Wert
  - Bsp.: (x,y) → (true,5) • x → 1 • id → $[\![\lambda x.x\ ,\ \rho]\!]$
- E = Expression
- α = Ergebnis der Auswertung
  - Integer, true/false, Closures $[\![\lambda P.E\ ,\rho]\!]$ , opaque Closures, Paare (α,β)

# Mini-ML - Dynamische Semantik II

- Regeln $\rho \vdash \mathrm{E} \Rightarrow \alpha$

(1) $\rho \vdash \text{number N} \Rightarrow N$

(2) $\rho \vdash \text{true} \Rightarrow true$

(3) $\rho \vdash \text{false} \Rightarrow false$

(4) $\rho \vdash \lambda P.E \Rightarrow [\![ \lambda P.E , \rho ]\!]$

(5) $$\frac{\rho \overset{\text{val\_of}}{\vdash} \text{ident } I \rightarrow \alpha}{\rho \text{ ident } I \Rightarrow \alpha}$$

(6) $$\frac{\rho \vdash E_1 \Rightarrow true \qquad \rho \vdash E_2 \Rightarrow \alpha}{\rho \vdash if\ E_1 then\ E_2 else\ E_3 \Rightarrow \alpha}$$

(7) $$\frac{\rho \vdash E_1 \Rightarrow false \qquad \rho \vdash E_3 \Rightarrow \alpha}{\rho \vdash if\ E_1 then\ E_2 else\ E_3 \Rightarrow \alpha}$$

(8) $$\frac{\rho \vdash E_1 \Rightarrow \alpha \qquad \rho \vdash E_2 \Rightarrow \beta}{\rho \vdash (E_1, E_2) \Rightarrow (\alpha, \beta)}$$

(9) $$\frac{\rho \vdash E_1 \Rightarrow [\![ \lambda P.E , \rho_1 ]\!] \qquad \rho \vdash E_2 \Rightarrow \alpha \qquad \rho_1 \bullet (P \rightarrow \alpha) \vdash E \Rightarrow \beta}{\rho \vdash E_1 E_2 \Rightarrow \beta}$$

(10) $$\frac{\rho \vdash E_2 \Rightarrow \alpha \qquad \rho \bullet (P \rightarrow \alpha) \vdash E_1 \Rightarrow \beta}{\rho \vdash let\ P = E_2 \text{ in } E_1 \Rightarrow \beta}$$

(11) $$\frac{\rho \bullet (P \rightarrow \alpha) \vdash E_2 \Rightarrow \alpha \qquad \rho \bullet (P \rightarrow \alpha) \vdash E_1 \Rightarrow \beta}{\rho \vdash letrec\ P = E_2 \text{ in } E_1 \Rightarrow \beta}$$

# Beispiel - Mini-ML

- Identitätsfunktion: let id = λx.x in id 1

$$\dfrac{\dfrac{}{\emptyset \vdash \lambda x.x \Rightarrow [\![\lambda x.x, \emptyset]\!]} \qquad \dfrac{\dfrac{\dfrac{}{id \to [\![\lambda x.x, \emptyset]\!] \vdash id \to [\![\lambda x.x, \emptyset]\!]}\,\text{val\_of}}{id \to [\![\lambda x.x, \emptyset]\!] \vdash id \Rightarrow [\![\lambda x.x, \emptyset]\!]} \qquad \dfrac{}{id \to [\![\lambda x.x, \emptyset]\!] \vdash 1 \Rightarrow 1} \qquad \dfrac{\dfrac{}{x \to 1 \vdash x \to 1}\,\text{val\_of}}{x \to 1 \vdash x \Rightarrow 1}}{id \to [\![\lambda x.x, \emptyset]\!] \vdash id\ 1 \Rightarrow 1}}{\emptyset \vdash let\ \mathrm{id} = \lambda x.x\ \mathrm{in}\ \mathrm{id}\ 1 \Rightarrow 1}$$

# Categorial Abstract Machine (CAM)

- Stackmaschine $s \cdot \alpha$
- $s \vdash c \Rightarrow \alpha$

  s State
  c Programmcode
  α Ergebnis
- Anweisungen:

  quote, car, cdr, cons, push, swap, op, branch, cur, app, rec
- Werte:

  Integer, true/false, Closures $[\![ c , \rho ]\!]_{cam}$, opaque Closures, Paare, Umgebungen (= geschachtelte Paare)

# CAM - Definition

- Regeln $\quad s \vdash c \Rightarrow \alpha$

(1) $\dfrac{\text{init\_stack} \vdash COMS \Rightarrow s \bullet \alpha}{\vdash program(COMS) \Rightarrow \alpha}$

(2) $s \vdash \emptyset \Rightarrow s$

(3) $\dfrac{s \vdash COM \Rightarrow s_1 \qquad s_1 \vdash COMS \Rightarrow s_2}{s \vdash COM\ ;COMS \Rightarrow s_2}$

(4) $s \bullet \alpha \vdash quote(v) \Rightarrow s \bullet v$

(5) $s \bullet (\alpha, \beta) \vdash car \Rightarrow s \bullet \alpha$

(6) $s \bullet (\alpha, \beta) \vdash cdr \Rightarrow s \bullet \beta$

(7) $s \bullet \alpha \bullet \beta \vdash cons \Rightarrow s \bullet (\alpha, \beta)$

(8) $s \bullet \alpha \vdash push \Rightarrow s \bullet \alpha \bullet \alpha$

(9) $s \bullet \alpha \bullet \beta \vdash swap \Rightarrow s \bullet \beta \bullet \alpha$

(10) $\dfrac{\vdash OP, \alpha \overset{eval}{\Rightarrow} \beta}{s \bullet \alpha \vdash op\,OP \Rightarrow s \bullet \beta}$

(11) $\dfrac{s \vdash c_1 \Rightarrow s_1}{s \bullet true \vdash branch(c_1, c_2) \Rightarrow s_1}$

(12) $\dfrac{s \vdash c_2 \Rightarrow s_1}{s \bullet false \vdash branch(c_1, c_2) \Rightarrow s_1}$

(13) $s \bullet \rho \vdash cur(c) \Rightarrow s \bullet [\![c, \rho]\!]_{cam}$

(14) $\dfrac{s \bullet (\rho, \alpha) \vdash c \Rightarrow s_1}{s \bullet ([\![c, \rho]\!]_{cam}, \alpha) \vdash app \Rightarrow s_1}$

(15) $\dfrac{s \bullet (\rho, \rho_1) \vdash c \Rightarrow s \bullet \rho_1}{s \bullet \rho \vdash rec(c) \Rightarrow s \bullet \rho_1}$

# Beispiel - Ausführung

- Identitätsfunktion: let id = λx.x in id 1

$$push\,;cur\left[cdr\right];cons\,;\,push\,;cdr\,;swap\,;quote\left(1\right);cons\,;app$$

- 

| Stack | CAM |
|---|---|
| $S \bullet \alpha$ | $push\,;$ |
| $S \bullet \alpha \bullet \alpha$ | $cur\left[cdr\right];$ |
| $S \bullet \alpha \bullet \left[\!\left[ cdr\,,\alpha \right]\!\right]_{cam}$ | $cons\,;$ |
| $S \bullet (\alpha, \left[\!\left[ cdr\,,\alpha \right]\!\right]_{cam})$ | $push\,;$ |
| $S \bullet (\alpha, \left[\!\left[ cdr\,,\alpha \right]\!\right]_{cam}) \bullet (\alpha, \left[\!\left[ cdr\,,\alpha \right]\!\right]_{cam})$ | $cdr\,;$ |
| $S \bullet (\alpha, \left[\!\left[ cdr\,,\alpha \right]\!\right]_{cam}) \bullet \left[\!\left[ cdr\,,\alpha \right]\!\right]_{cam}$ | $swap\,;$ |
| $S \bullet \left[\!\left[ cdr\,,\alpha \right]\!\right]_{cam} \bullet (\alpha, \left[\!\left[ cdr\,,\alpha \right]\!\right]_{cam})$ | $quote\left(1\right);$ |
| $S \bullet \left[\!\left[ cdr\,,\alpha \right]\!\right]_{cam} \bullet 1$ | $cons\,;$ |
| $S \bullet (\left[\!\left[ cdr\,,\alpha \right]\!\right]_{cam}, 1)$ | $app\,;$ |
| $S \bullet (\alpha, 1)$ | $cdr\,;$ |
| $S \bullet 1$ | |

# Übersetzung - Mini-ML nach CAM

- Regeln $\rho \vdash \mathrm{E} \Rightarrow \mathrm{c}$

(1) $$\frac{\mathrm{init\_pat} \vdash E \rightarrow c}{\vdash E \rightarrow program(c)}$$

(2) $$\rho \vdash \mathrm{number\ N} \rightarrow \mathrm{quote}(N)$$

(3) $$\rho \vdash \mathrm{true} \rightarrow \mathrm{quote}(true)$$

(4) $$\rho \vdash \mathrm{false} \rightarrow \mathrm{quote}(false)$$

(5) $$\frac{\rho \overset{access}{\vdash} \mathrm{ident}\, I : c}{\rho \vdash \mathrm{ident}\, I \rightarrow c}$$

(6) $$\frac{(\rho, P) \vdash E \rightarrow c}{\rho \vdash \lambda P.E \rightarrow cur(c)}$$

(7) $$\frac{\rho \vdash E_1 \rightarrow c_1 \qquad \rho \vdash E_2 \rightarrow c_2 \qquad \rho \vdash E_3 \rightarrow c_3}{\rho \vdash if\ E_1\ then\ E_2\ else\ E_3 \rightarrow push\ ;\ c_1\ ;\ branch(c_2, c_3)}$$

(8) $$\frac{\rho \vdash E_1 \rightarrow c_1 \qquad \rho \vdash E_2 \rightarrow c_2}{\rho \vdash (E_1, E_2) \rightarrow push\ ;\ c_1\ ;\ swap\ ;\ c_2\ ;\ cons}$$

(9) $$\frac{\rho \vdash E_1 \rightarrow c_1 \qquad \rho \vdash E_2 \rightarrow c_2}{\rho \vdash E_1\, E_2 \rightarrow push\ ;\ c_1\ ;\ swap\ ;\ c_2\ ;\ cons\ ;\ app}$$

(10) $$\frac{\rho \vdash E_1 \rightarrow c_1 \qquad (\rho, P) \vdash E_2 \rightarrow c_2}{\rho \vdash let\ P = E_1\ \mathrm{in}\ E_2 \rightarrow push\ ;\ c_1\ ;\ cons\ ;\ c_2}$$

(11) $$\frac{(\rho, P) \vdash E_1 \rightarrow c_1 \qquad (\rho, P) \vdash E_2 \rightarrow c_2}{\rho \vdash letrec\ P = E_1\ \mathrm{in}\ E_2 \rightarrow push\ ;\ rec(c_1)\ ;\ cons\ ;\ c_2}$$

# Beispiel - Übersetzung

- Identitätsfunktion: let id = λx.x in id 1

  10)push; λx.x; cons; id 1;
  6) push; cur( x ); cons; id 1;
  9) push; cur( x ); cons; push; id; swap; 1; cons; app;
  2) push; cur( x ); cons; push; id; swap; quote(1); cons; app;
  access)
  => push; cur(cdr); cons; push; cdr; swap; quote(1); cons; app;

$$\dfrac{\dfrac{\dfrac{(\emptyset, x) \vdash x : cdr}{(\emptyset, x) \vdash x \to cdr}}{\emptyset \vdash \lambda x.x \to cur[cdr]} \qquad \dfrac{\dfrac{(\emptyset, id) \vdash id : cdr}{(\emptyset, id) \vdash id \to cdr} \qquad \dfrac{}{(\emptyset, id) \vdash 1 \to quote(1)}}{(\emptyset, id) \vdash id\ 1 \to push; cdr; swap; quote(1); cons; app}}{\emptyset \vdash let\ id = \lambda x.x\ \text{in}\ id\ 1 \to push; cur[cdr]; cons; push; cdr; swap; quote(1); cons; app}$$

# Zusammenfassung

- Jugments für Typregeln $\rho \vdash \mathrm{E} : \tau$
- Dynamische Semantik ermöglicht einfaches Backpatching bei Übersetzungen
- Hohe Abstraktion ermöglicht einfaches Overloading
- Einfache mathematisch Methode eine semantische Beschreibung zu erstellen
- Mittlerweile Standard

E N D E