# From Greek to Clojure

Nada Amin and William Byrd

Clojure/conj, Alexandria, VA

November 14, 2013

# Peano Numbers: Syntax

- a number $n$ is either
  - zero $z$
  - or the successor of a number $(s\ n)$
- $n := z \mid (s\ n)$

# Peano Numbers: Plus Relation

$$z + n = n \qquad \text{(PLUS-Z)}$$

$$\frac{n_1 + n_2 = n_3}{(s\ n_1) + n_2 = (s\ n_3)} \qquad \text{(PLUS-S)}$$

# Syntax

- variable $x, y, z$
- term $e :=$
  - variable $x$
  - abstraction $\lambda x.e$
  - application $(e_1\ e_2)$

# Typing

$$\frac{(x : T) \in \Gamma}{\Gamma \vdash x : T} \quad (\textsc{var})$$

$$\frac{\Gamma, (x : T_1) \vdash e : T_2}{\Gamma \vdash \lambda x.e : T_1 \to T_2} \quad (\textsc{abs})$$

$$\frac{\Gamma \vdash e_1 : T_1 \to T \qquad \Gamma \vdash e_2 : T_1}{\Gamma \vdash (e_1\ e_2) : T} \quad (\textsc{app})$$

# Reduction

- value $v := \lambda x.e$

$$((\lambda x.e)\ v) \implies [v/x]e \qquad \text{(APP-BETA)}$$

$$\frac{e_1 \implies e_1'}{(e_1\ e_2) \implies (e_1'\ e_2)} \qquad \text{(APP-1)}$$

$$\frac{e_2 \implies e_2'}{(v_1\ e_2) \implies (v_1\ e_2')} \qquad \text{(APP-2)}$$

# CPS: Syntax of Target

- atom expression $a :=$
  - variable $x$
  - abstraction $\lambda[x*].a$
- complex expresion $c :=$
  - application $(a\ a*)$

# CPS: Rules

$$x \Rightarrow^{M} x \qquad \text{(M-VAR)}$$

$$\frac{e|y_k \Rightarrow^{T} c}{\lambda x.e \Rightarrow^{M} \lambda[x\ y_k].c} \qquad \text{(M-ABS)}$$

$$\frac{x \Rightarrow^{M} c_x}{x|a_k \Rightarrow^{T} (a_k\ c_x)} \qquad \text{(T-VAR)}$$

$$\frac{\lambda x.e \Rightarrow^{M} c}{\lambda x.e|a_k \Rightarrow^{T} (a_k\ c)} \qquad \text{(T-ABS)}$$

$$\frac{\begin{array}{c} e_1|\lambda x_1.c_2 \Rightarrow^{T} c_1 \\ e_2|\lambda x_2.(x_1\ x_2\ a_k) \Rightarrow^{T} c_2 \end{array}}{(e_1\ e_2)|a_k \Rightarrow^{T} c_1} \qquad \text{(T-APP)}$$

# Evaluation

- value $v := < \lambda x.e \textbf{ in } \rho >$

$$\frac{(x : v) \in \rho}{\rho \vdash x \Downarrow v} \quad (\textsc{var})$$

$$\rho \vdash \lambda x.e \Downarrow < \lambda x.e \textbf{ in } \rho > \quad (\textsc{abs})$$

$$\frac{\begin{array}{c}\rho \vdash e_1 \Downarrow < \lambda x.e \textbf{ in } \rho_1 > \\ \rho \vdash e_2 \Downarrow v_2 \\ \rho_1, (x : v_2) \vdash e \Downarrow v\end{array}}{\rho \vdash (e_1\ e_2) \Downarrow v} \quad (\textsc{app})$$