Recursion and Dynamic Data-structures in Bounded Space: Towards Embedded ML Programming

John Hughes and Lars Pareto
ICFP'99

Presented by Oukseh Lee

Research On Program Analysis System, KAIST

May 20, 2000

Result

Well-typed with the stack-effect δ and the store effect ϕ,

$$\Downarrow$$

Use at most $\delta + \phi$ words of memory.

- Language: strict, first-order, explicit (sized) region.
- Semantics: the abstract machine $\langle V, S, E, C \rangle$.
  - V: temporary stack.
  - S: region stack.
- Type: (linear) sized type with stack-effect, put-effect, and store-effect.
- Type system: the effect system $\Gamma \models_{\mathrm{F}} e :: \tau \;!\; \delta; p; \phi$.

Example

$len :: \forall kar.\mathsf{List}_k\, a\, r \to \mathsf{Nat}_k$ **with** $\delta = 5k$
$len\ xs = \mathsf{case}\ xs\ \mathsf{of}\ \mathsf{nil} \to 0;\ \mathsf{cons}\ x\ s \to 1 + len\ s$

$fall :: \forall kr.\mathsf{Nat}_k \times r \to \mathsf{List}_{k+1}\ \mathsf{Nat}_k\ r$ **with** $\delta = 5k;\ r \mathrel{+}= 3k+1$
$fall\ n\ r = \mathsf{cons}\, n\, (\mathsf{case}\, n\ \mathsf{of}\ 0 \to \mathsf{nil}\ r;\ x+1 \to fall\ x\ r)\ r$

$\texttt{letreg}\, x \,\#\, \check{3}\check{*}\check{5}\check{+}\check{1}\ \texttt{in}\ len\ (fall\ 4\ x)$

Language & Type

$$
\begin{array}{rcl}
P & ::= & g \cdots g \; e \\
g & ::= & d \cdots d \\
d & ::= & f \; x \cdots x \; \texttt{=} \; e \\
e & ::= & n \mid \check{n} \mid x \mid c \; e \cdots e \; e \mid e \; op \; e \mid f \; e \cdots e \\
 & \mid & \texttt{let} \; x \,\texttt{=}\, e \; \texttt{in} \; e \mid \texttt{letreg} \; x \,\#\, e \; \texttt{in} \; e \mid (e, \cdots, e)_\epsilon \mid \pi_n \; e \\
 & \mid & \texttt{case} \; e \; \texttt{of} \; 0 \to e \; [] \; x\texttt{+1} \to e \mid \texttt{case} \; e \; \texttt{of} \; A \cdots A \\
op & ::= & \texttt{+} \mid \texttt{-} \mid \texttt{*} \mid \check{\texttt{+}} \mid \check{\texttt{-}} \mid \check{\texttt{*}} \\
A & ::= & c \; x \cdots x \to e \\
\sigma & ::= & \forall k.\sigma \mid \forall t.\sigma \mid \forall \rho.\sigma \mid \nu \\
\nu & ::= & \tau \times \cdots \times \tau \xrightarrow{\delta;\, p;\, \phi} \tau \mid \tau \times \cdots \times \tau \times \rho \to \kappa \\
\tau & ::= & t \mid \rho \mid [\tau \times \cdots \times \tau]_\rho \mid \mathsf{Nat}_s \mid \mathsf{Size}_s \mid \kappa \\
\kappa & ::= & T_{s \cdots s} \; \tau \cdots \tau \; \rho \\
p & ::= & \{\rho \mapsto n, \cdots \rho \mapsto n\} \\
s & ::= & \omega \mid i \\
i & ::= & k \mid n \mid n * i \mid i + i \\
 & & \delta, \phi, n \in \mathbf{N}
\end{array}
$$

Abstract Machine

$$
\begin{array}{lcl}
M & ::= & \langle V, S, E, C \rangle \\
V & ::= & v\colon V \mid v \\
v & ::= & n \mid \check{n} \mid a \mid \rho \mid c \mid C_{ES} \\
a & ::= & o_\rho \\
S & ::= & \{\rho \mapsto R, \cdots, \rho \mapsto R\} \\
R & ::= & \{o \mapsto v, \cdots, o \mapsto v\}^n \\
E & ::= & \{x \mapsto v, \cdots, x \mapsto v\} \\
C & ::= & \iota\colon C \mid \mathbf{call}_f \mid \mathbf{ret}_n \mid \mathbf{sret}_n^{C_{ES}} \mid \mathbf{end} \\
\iota & ::= & e \mid op \mid e\,[]_x\,e \mid \ \not\rho \mid ()^n \mid \pi_n \mid c^n \\
 & \mid & A \cdots A \mid \lambda x.e \mid \{\}_x e \\
o & \in & Nat \quad \rho \in RegVar
\end{array}
$$

Dynamic and Static Semantics

- Numbers and Variables

$$\begin{array}{rcl}\langle V, S, E, n\colon C\rangle & \to & \langle n\colon V, S, E, C\rangle \\ \langle V, S, E, x\colon C\rangle & \to & \langle E(x)\colon V, S, E, C\rangle\end{array}$$

$$\begin{array}{l}\Gamma \models_{\mathbb{F}} n :: \mathsf{Nat}_{n+1} \,!\, 1; \{\}; 0 \\ \Gamma \models_{\mathbb{F}} \check{n} :: \mathsf{Size}_{n} \,!\, 1; \{\}; 0 \\ \Gamma, x :: \tau \models_{\mathbb{F}} x :: \tau \,!\, 1; \{\}; 0\end{array}$$

- Constructors

$$\begin{array}{rcl}\langle V, S, E, c\, e_1 \cdots e_n\colon C\rangle & \to & \langle V, S, E, e_1\colon \cdots : e_n\colon c^n\colon C\rangle \\ \langle \rho\colon v_n\colon \cdots : v_1\colon V, S, E, c^n\colon C\rangle & \to & \langle o_\rho\colon V, S', E, C\rangle;\ o = |S(\rho)_{\mathbf{D}}|\end{array}$$

$$S' \equiv S + \{o_\rho \mapsto c\} + \{(o+i)_\rho \mapsto v_i\}$$

$$\frac{\begin{array}{l}c :: \tau_1 \times \cdots \times \tau_n \times \rho \to \kappa \quad \tau_i' \rhd \tau_i \\ \Gamma \models_{\mathbb{F}} e_1 :: \tau_i' \,!\, \delta_i; p_i; \phi_i \quad \Gamma \models_{\mathbb{F}} e :: \rho \,!\, \delta_{n+1}; p_{n+1}; \phi_{n+1}\end{array}}{\Gamma \models_{\mathbb{F}} c\, e_1 \cdots e_n\, e :: \kappa \,!\, \sqcup_{i=1}^{n+1} (i - 1 + \delta_i); \Sigma_i p_i + \{\rho \mapsto 1 + n\}; \sqcup_i \phi}$$

- Arithmetic Operator

$$\begin{array}{rcl}\langle V, S, E, e_1 * e_2 \colon C\rangle & \rightarrow & \langle V, S, E, e_1 \colon e_2 \colon * \colon C\rangle \\ \langle n_2 \colon n_1 \colon V, S, E, * \colon C\rangle & \rightarrow & \langle n_1 * n_2 \colon V, S, E, C\rangle\end{array}$$

$$\frac{\Gamma \models_{\mathbb{F}} e_1 :: \mathsf{Nat}_{s_i} \,!\, \delta_i; p_i; \phi_i}{\Gamma \models_{\mathbb{F}} e_1 * e_2 :: \mathsf{Nat}_{\omega} \,!\, \delta_1 \cup (1 + \delta_2); p_1 + p_2; \phi_1 \cup \phi_2}$$

$$\frac{\Gamma \models_{\mathbb{F}} e_1 :: \mathsf{Size}_{n} \,!\, \delta_1; p_1; \phi_1 \quad \Gamma \models_{\mathbb{F}} e_2 :: \mathsf{Size}_{s} \,!\, \delta_2; p_2; \phi_2}{\Gamma \models_{\mathbb{F}} e_1 \check{*} e_2 :: \mathsf{Size}_{n*s} \,!\, \delta_1 \cup (1 + \delta_2); p_1 + p_2; \phi_1 \cup \phi_2}$$

- Function Calls

$$\begin{array}{rcl}\langle V, S, E, f\ e_1 \cdots e_n \colon C\rangle & \rightarrow & \langle C_{ES} \colon V, S, E, e_1 \colon \cdots \colon e_n \colon \mathbf{call}_f\rangle \\ \langle v_n \colon \cdots \colon v_1 \colon C_{ES'} \colon V, S, E', \mathbf{call}_f\rangle & \rightarrow & \\ & & \langle v_n \colon \cdots \colon v_1 \colon C_{ES'} \colon V, S, \{x_i \mapsto v_i\}, e_f \colon \mathbf{ret}_n\rangle \\ \langle v \colon v_n \colon \cdots \colon v_1 \colon C_{ES'} \colon V, S, E', \mathbf{ret}_n\rangle & \rightarrow & \langle v \colon V, S, E, C\rangle;\ S' \subseteq S\end{array}$$

$$\frac{\begin{array}{l}\Gamma \vdash_F f :: \tau_1 \times \cdots \times \tau_n \overset{\delta_{n+1}; p_{n+1}; \phi_{n+1}}{\longrightarrow} \tau \\ \Gamma \models_{\mathbb{F}} e_i :: \tau'_i \,!\, \delta_i; p_i; \phi_i \quad \tau'_i \rhd \tau_i\end{array}}{\Gamma \models_{\mathbb{F}} f\ e_1 \cdots e_n :: \tau \,!\, 1 + \sqcup_{i=1}^{n+1}(i - 1 + \delta_i); \Sigma_i p_i; \sqcup_i \phi_i}$$

- Let-Bindings

$$\langle V, S, E, \texttt{let}\; x \,\texttt{=}\, e \;\texttt{in}\; e' \colon C\rangle \;\rightarrow\; \langle V, S, E, e \colon \lambda x.e' \colon C\rangle$$

$$\langle v \colon V, S, E, \lambda x.e \colon C\rangle \;\rightarrow\; \langle v \colon V, S, E + \{x \mapsto V\}, e \colon \mathbf{sret}_1^{C_{ES}}\rangle$$

$$\langle v \colon v_n \colon \cdots \colon v_1 \colon V, S, E', \mathbf{sret}_n^{C_{ES'}}\rangle \;\rightarrow\; \langle v \colon V, S, E, C\rangle;\;\; S' \subseteq S$$

$$\frac{\Gamma \models_{\mathbb{F}} e :: \tau \mathbin{!} \delta; p; \phi \quad \Gamma, x :: \tau \models_{\mathbb{F}} e' :: \tau' \mathbin{!} \delta'; p'; \phi'}{\Gamma \models_{\mathbb{F}} \texttt{let}\; x \,\texttt{=}\, e \;\texttt{in}\; e' :: \tau' \mathbin{!} \delta \sqcup (1 + \delta'); p + p'; \phi \sqcup \phi'}$$

- Let Region

$$\langle V, S, E, \texttt{letreg}\; x \,\#\, e \;\texttt{in}\; e' \colon C\rangle \;\rightarrow\; \langle V, S, E, e \colon \{\}_x e' \colon C\rangle$$

$$\langle \check{n} \colon V, S, E, \{\}_x e \colon C\rangle \;\rightarrow$$

$$\langle \rho \colon V, S + \{\rho \mapsto \{\}^{n+1}\}, E + \{x \mapsto \rho\}, e \colon \not\rho \colon \mathbf{sret}_1^{C_{ES}}\rangle;\;\; \rho \notin S_{\mathbf{D}}$$

$$\langle v \colon \rho \colon V, S, E, \not\rho \colon \mathbf{sret}_1^{C_{E'S'}}\rangle \;\rightarrow\; \langle v \colon \rho \colon V, S \backslash \rho, \emptyset, \mathbf{sret}_1^{C_{E'S'}}\rangle$$

$$\frac{\Gamma \models_{\mathbb{F}} e :: \mathsf{Size}_s \mathbin{!} \delta; p; \phi \quad \Gamma, x :: \rho \models_{\mathbb{F}} e' :: \tau \mathbin{!} \delta'; p'; \phi' \quad \rho \notin FV(\Gamma, \tau) \quad p'(\rho) \leq s}{\Gamma \models_{\mathbb{F}} \texttt{letreg}\; x \,\#\, e \;\texttt{in}\; e' :: \tau \mathbin{!} \delta \sqcup (1 + \delta'); p + p' \backslash \rho; \phi \sqcup (s + 1 + \phi')}$$

- Case Expressions

$$\langle V, S, E, \texttt{case}\; e\; \texttt{of}\; \texttt{0} \rightarrow e_1 \;[]\; x\texttt{+1} \rightarrow e_2 \colon C\rangle \;\rightarrow\; \langle V, S, E, e \colon e_1 \;[]_x\; e_2 \colon C\rangle$$
$$\langle 0 \colon V, S, E, e_1 \;[]_x\; e_2 \colon C\rangle \;\rightarrow\; \langle V, S, E, e_1 \colon C\rangle$$
$$\langle v \colon V, S, E, e_1 \;[]_x\; e_2 \colon C\rangle \;\rightarrow\; \langle v-1 \colon V, S, E + \{x \mapsto v-1\}, e_2 \colon \mathbf{sret}_1^{C_{ES}}\rangle$$

$$\frac{\begin{array}{c}\Gamma \models_{\mathrm{F}} e :: \mathsf{Nat}_{s+1} \,!\, \delta_0; p_0; \phi_0 \quad \Gamma \models_{\mathrm{F}} e_1 :: \tau_1 \,!\, \delta_1; p_1; \phi_1 \\ \Gamma, x :: \mathsf{Nat}_s \models_{\mathrm{F}} e_2 :: \tau_2 \,!\, \delta_2; p_2; \phi_2 \quad \tau_i \rhd \tau\end{array}}{\Gamma \models_{\mathrm{F}} \texttt{case}\; e\; \texttt{of}\; \texttt{0} \rightarrow e_1 \;[]\; x\texttt{+1} \rightarrow e_2 :: \tau \,!\, \delta_0 \sqcup \delta_1 \sqcup (1+\delta_2); p_0 + (p_1 \sqcup p_2); \sqcup_i \phi_i}$$

- Function Definitions

$$\frac{x_1 :: \tau_1, \cdots, x_n :: \tau_n \models_{\mathrm{F}} e :: \tau \,!\, \delta; p; \phi}{\mathrm{F} \vdash f\; x_1 \cdots x_n \texttt{=} e :: \tau_1 \times \cdots \times \tau_n \xrightarrow{\delta;p;\phi} \tau}$$

$$\frac{\begin{array}{c}\sigma_{i[k]} \equiv \forall \bar{k}_i \bar{t}_i \bar{\rho}_i . \nu_{i[k]} \quad FV(\forall k . \sigma_{i[k]}) = \emptyset \quad \nu'_i \rhd \nu_{i[k+1]} \\ \mathrm{F}, f_1 :: \sigma_{1[k]}, \cdots, f_n :: \sigma_{n[k]} \vdash f_i\; \bar{x}_i \texttt{=} e_i :: \nu'_i \quad \sigma_{i[0]} = \mathbf{U}\end{array}}{\mathrm{F} \vdash \{f_i\; \bar{x}_i \texttt{=} e_i\} \Rightarrow \mathrm{F}, f_1 :: \forall k . \sigma_{1[k]}, \cdots f_n :: \forall k . \sigma_{n[k]}}$$

Discussion

- Checking is feasible. They believe the inference of effects is feasible.
- Higher-order functions.
  - Problem: the size of function values
  - Proposed solution: only allowing partially applied function.