Abstract Parsing for Two-staged Language

ROPAS Weekly Show&Tell

Soonho Kong

9 February 2009

Motivation & Goal

Motivation

In multi-staged programming,

program generates another program.

Goal

Check syntax of generated program.

Contents

Two-staged Language

$$
\begin{array}{rcl}
e & ::= & \texttt{let}\; x\; e\; e \\
  & | & \texttt{re}\; x\; e\; e\; e \\
  & | & \texttt{or}\; e\; e \\
  & | & \texttt{`}f \\
  & | & x
\end{array}
\qquad
\begin{array}{rcl}
f & ::= & \texttt{let} \\
  & | & \texttt{re} \\
  & | & \texttt{or} \\
  & | & x \\
  & | & ,e \\
  & | & f.f
\end{array}
$$

Collecting Semantics

$$
\begin{array}{rcl}
\mathcal{V}_0 x \Sigma & = & \{\sigma(x) \mid \sigma \in \Sigma\} \\
\mathcal{V}_0(\texttt{or}\ e_1\ e_2)\Sigma & = & \bigcup_{\sigma\in\Sigma} (\mathcal{V}_0 e_1\{\sigma\} \cup \mathcal{V}_0 e_2\{\sigma\}) \\
\mathcal{V}_0(\texttt{let}\ x\ e_1\ e_2)\Sigma & = & \bigcup_{\sigma\in\Sigma} \bigcup_{c\in\mathcal{V}_0 e_1\{\sigma\}} \mathcal{V}_0 e_2\{\sigma[x \mapsto c]\} \\
\mathcal{V}_0(\texttt{re}\ x\ e_1\ e_2\ e_3)\Sigma & = & \bigcup_{\sigma\in\Sigma} \mathcal{V}_0 e_3\{\sigma[x \mapsto fix \lambda k.(\mathcal{V}_0 e_1\{\sigma\} \sqcup \mathcal{V}_0 e_2\{\sigma[x \mapsto k]\})]\} \\
\mathcal{V}_0(\grave{}\, f)\Sigma & = & \mathcal{V}_1 f \Sigma \\
\mathcal{V}_1(f_1.f_2)\Sigma & = & \bigcup_{\sigma\in\Sigma} \{xy \mid x \in \mathcal{V}_1 f_1\{\sigma\}, y \in \mathcal{V}_1 f_2\{\sigma\}\} \\
\mathcal{V}_1(,e)\Sigma & = & \mathcal{V}_0 e \Sigma \\
\mathcal{V}_1\, \texttt{or}\, \Sigma & = & \{\text{``}\texttt{or}\text{''}\} \\
\mathcal{V}_1\, \texttt{let}\, \Sigma & = & \{\text{``}\texttt{let}\text{''}\} \\
\mathcal{V}_1\, \texttt{re}\, \Sigma & = & \{\text{``}\texttt{re}\text{''}\} \\
\mathcal{V}_1 x \Sigma & = & \{\text{``}\texttt{x}\text{''}\}
\end{array}
$$

$$
\boxed{
\begin{array}{rcl}
V_C & : & Code \\
\sigma \in Env & : & Var \to V_C \\
\mathcal{V} & : & Pgm \to 2^{Env} \to 2^{V_C}
\end{array}
}
$$

Abstract Collecting Semantics

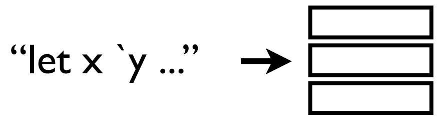
$$
\begin{array}{rcl}
\mathcal{V}_0 x \sigma & = & \sigma(x) \\
\mathcal{V}_0(\texttt{or}\; e_1\; e_2)\sigma & = & \mathcal{V}_0 e_1 \sigma \sqcup \mathcal{V}_0 e_2 \sigma \\
\mathcal{V}_0(\texttt{let}\; x\; e_1\; e_2)\sigma & = & \mathcal{V}_0 e_2(\sigma[x \mapsto \mathcal{V}_0 e_1 \sigma]) \\
\mathcal{V}_0(\texttt{re}\; x\; e_1\; e_2\; e_3)\sigma & = & \mathcal{V}_0 e_3(\sigma[x \mapsto fix \lambda k.((\mathcal{V}_0 e_1 \sigma) \sqcup (\mathcal{V}_0 e_2(\sigma[x \mapsto k])))]) \\
\mathcal{V}_0(\grave{}\, f)\sigma & = & \mathcal{V}_1 f \sigma \\
\mathcal{V}_1(f_1.f_2)\sigma & = & \{xy \mid x \in \mathcal{V}_1 f_1 \sigma, y \in \mathcal{V}_1 f_2 \sigma\} \\
\mathcal{V}_1(,e)\sigma & = & \mathcal{V}_0 e \sigma \\
\mathcal{V}_1\, \texttt{or}\, \sigma & = & \{\texttt{``or''}\} \\
\mathcal{V}_1\, \texttt{let}\, \sigma & = & \{\texttt{``let''}\} \\
\mathcal{V}_1\, \texttt{re}\, \sigma & = & \{\texttt{``re''}\} \\
\mathcal{V}_1 x \sigma & = & \{\texttt{``x''}\}
\end{array}
$$

$$
\begin{array}{rcl}
\widehat{V_C} & : & 2^{Code} \\
\widehat{\sigma} \in \widehat{Env} & : & Var \to \widehat{V_C} \\
\mathcal{V} & : & Pgm \to \widehat{Env} \to \widehat{V_C}
\end{array}
$$

Concrete Parsing

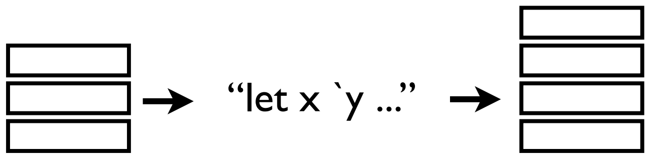
We usually think it of

$$Parse : Code \rightarrow P$$



More generally, we could think it of

$$Parse : P \rightarrow Code \rightarrow P$$



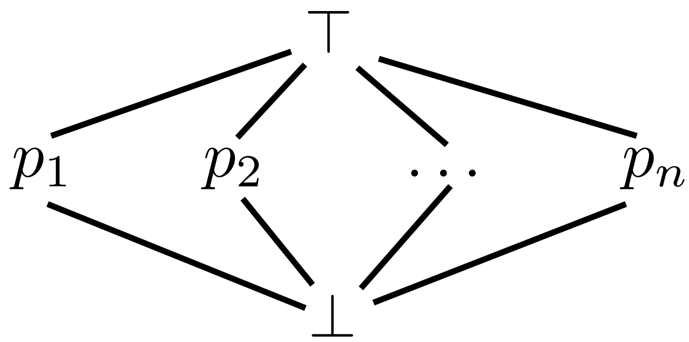
It is composition of

$$Parse\_Action : P \rightarrow Token \rightarrow P$$

Concrete Parsing : Domain

Concrete Parse Stack Domain

$$P \quad = \quad \Sigma^{+} \cup \{\bot, \top\}$$



Concrete Parsing Domain

$$V_P = \quad P \rightarrow 2^{P} \cup \{\bot, \top\}$$

$$\sqsubseteq \quad : \quad f_1 \sqsubseteq f_2 \qquad \qquad \forall p \in P : f_1(p) \sqsubseteq f_2(p)$$

$$\sqcup \quad : \quad f_1 \sqcup f_2 = \lambda p. f_1(p) \sqcup f_2(p)$$

Concrete Parsing Semantics

$$\begin{array}{rcl} \sigma \in Env_P & : & Var \to V_P \\ \mathcal{V}_P & : & Pgm \to Env_P \to V_P \end{array}$$

$$\begin{array}{rcl}
\mathcal{V}_P^0 x\sigma & = & \sigma(x) \\
\mathcal{V}_P^0(\texttt{or}\ e_1\ e_2)\sigma & = & \lambda p.(\mathcal{V}_P^0 e_1\sigma p \cup \mathcal{V}_P^0 e_2\sigma p) \\
\mathcal{V}_P^0(\texttt{let}\ x\ e_1\ e_2)\sigma & = & \mathcal{V}_P^0 e_2(\sigma[x \mapsto \mathcal{V}_P^0 e_1\sigma]) \\
\mathcal{V}_P^0(\texttt{re}\ x\ e_1\ e_2\ e_3)\sigma & = & \mathcal{V}_P^0 e_3(\sigma[x \mapsto \lambda p.fix\lambda k.((\mathcal{V}_P^0 e_1\sigma p) \sqcup (\mathcal{V}_P^0 e_2(\sigma[x \mapsto k])p))]) \\
\mathcal{V}_P^0 \grave{}\, f\sigma & = & \mathcal{V}_P^1 f\sigma \\
\mathcal{V}_P^1(f_1.f_2)\sigma & = & \lambda p. \displaystyle\bigcup_{p' \in \mathcal{V}_P^1 f_1\sigma p} \mathcal{V}_P^1 f_2\sigma p' \\
\mathcal{V}_P^1(,e)\sigma & = & {\mathcal{V}_P}^0 e\sigma \\
\mathcal{V}_P^1 a\sigma & = & \lambda p.\{parse_action(p,a)\}
\end{array}$$

Abstract Parsing

Abstract parse stack represents infinite number of concrete parse stacks

$$expand(s_0 s_1^+) = \{s_0 s_1, s_0 s_1 s_1, s_0 s_1 s_1 s_1, \ldots\}$$

$$\begin{array}{rll} expand(s) = \{s\} & & s \in \Sigma \\ expand(ws) = \{xy \mid x \in expand(w), y \in expand(v)\} & & w \in \Sigma^+ \wedge s \in \Sigma \\ expand(w^+) = expand(w) \cup expand(ww) \cup expand(www) \ldots & & w \in \Sigma^+ \end{array}$$

Domain $\widehat{P} = L(A) \cup \{\perp, \top\}$ is defined by

$$\begin{array}{lll|lll} A & ::= & \Sigma & \sqsubseteq & : & \widehat{p} \sqsubseteq \top \\ & \mid & A\ A & & & \perp \sqsubseteq \widehat{p} \\ & \mid & A^+ & & & \widehat{p}_1 \sqsubseteq \widehat{p}_2 \quad \text{iff } RE\_Inclusion(\widehat{p}_1, \widehat{p}_2) = T \\ & & & \sqcup & : & \widehat{p}_1 \sqcup \widehat{p}_2 = \widehat{p}_2 \quad (\widehat{p}_1 \sqsubseteq \widehat{p}_2) \\ & & & & & \widehat{p}_1 \sqcup \widehat{p}_2 = \widehat{p}_1 \quad (\widehat{p}_2 \sqsubseteq \widehat{p}_1) \\ & & & & & \widehat{p}_1 \sqcup \widehat{p}_2 = \top \quad (otherwise) \end{array}$$

Abstract Parsing

Powerset Domain : $2^{\widehat{P}}$

$$\sqsubseteq \quad : \quad \widehat{S}_1 \sqsubseteq \widehat{S}_2 \qquad\qquad \forall \widehat{s}_1 \in \widehat{S}_1 \; \exists \widehat{s}_2 \in \widehat{S}_2 : RE\_Inclusion(\widehat{s}_1, \widehat{s}_2)$$

$$\sqcup \quad : \quad \widehat{S}_1 \sqcup \widehat{S}_2 = \{ s \in \widehat{S}_1 \cup \widehat{S}_2 \mid \forall s' \in (\widehat{S}_1 \cup \widehat{S}_2)/\{s\} : s \notin expand(s') \}$$

Abstract Parsing Domain : $V_{\widehat{P}}$

$$V_{\widehat{P}} \quad = \quad \widehat{P} \rightarrow 2^{\widehat{P}} \cup \{\bot, \top\}$$

$$\sqsubseteq \quad : \quad \widehat{f}_1 \sqsubseteq \widehat{f}_2 \qquad\qquad \forall \widehat{p} \in \widehat{P} : \widehat{f}_1(\widehat{p}) \sqsubseteq \widehat{f}_2(\widehat{p})$$

$$\sqcup \quad : \quad \widehat{f}_1 \sqcup \widehat{f}_2 = \lambda \widehat{p}. \widehat{f}_1(\widehat{p}) \sqcup \widehat{f}_2(\widehat{p})$$

Abstract Parsing Semantics

$$\begin{array}{rcl} \widehat{\sigma} \in Env_{\widehat{P}} & : & Var \rightarrow V_{\widehat{P}} \\ \mathcal{V}_{\widehat{P}} & : & Pgm \rightarrow Env_{\widehat{P}} \rightarrow V_{\widehat{P}} \end{array}$$

$$\begin{array}{rcl}
\mathcal{V}^0_{\widehat{P}} x \widehat{\sigma} & = & \widehat{\sigma}(x) \\
\mathcal{V}^0_{\widehat{P}} (\texttt{or}\ e_1\ e_2) \widehat{\sigma} & = & \lambda \widehat{p}.(\mathcal{V}^0_{\widehat{P}} e_1 \widehat{\sigma} \widehat{p} \cup \mathcal{V}^0_{\widehat{P}} e_2 \widehat{\sigma} \widehat{p}) \\
\mathcal{V}^0_{\widehat{P}} (\texttt{let}\ x\ e_1\ e_2) \widehat{\sigma} & = & \mathcal{V}^0_{\widehat{P}} e_2 (\widehat{\sigma}[x \mapsto \mathcal{V}^0_{\widehat{P}} e_1 \widehat{\sigma}]) \\
\boxed{\mathcal{V}^0_{\widehat{P}} (\texttt{re}\ x\ e_1\ e_2\ e_3) \widehat{\sigma} \quad = \quad \mathcal{V}^0_{\widehat{P}} e_3 (\widehat{\sigma}[x \mapsto \lambda \widehat{p}.fix \lambda k.((\mathcal{V}^0_{\widehat{P}} e_1 \widehat{\sigma} \widehat{p}) \sqcup (\mathcal{V}^0_{\widehat{P}} e_2 (\widehat{\sigma}[x \mapsto k]) \widehat{p}))])} \\
\mathcal{V}^0_{\widehat{P}} \grave{}\, f \widehat{\sigma} & = & \mathcal{V}^1_{\widehat{P}} f \widehat{\sigma} \\
\mathcal{V}^1_{\widehat{P}} (f_1 . f_2) \widehat{\sigma} & = & \lambda \widehat{p}. \displaystyle\bigcup_{\widehat{p}' \in \mathcal{V}^1_{\widehat{P}} f_1 \widehat{\sigma} \widehat{p}} \mathcal{V}^1_{\widehat{P}} f_2 \widehat{\sigma} \widehat{p}' \\
\mathcal{V}^1_{\widehat{P}} (, e) \widehat{\sigma} & = & \mathcal{V}^0_{P} e \widehat{\sigma} \\
\boxed{\mathcal{V}^1_{\widehat{P}} a \widehat{\sigma} \quad = \quad \lambda \widehat{p}.\{parse_action(\widehat{p}, a)\}}
\end{array}$$

Abstract Parsing Semantics : Folding

$$
\begin{array}{rcl}
fold & : & \widehat{P} \rightarrow 2^{\widehat{P}} \rightarrow 2^{\widehat{P}} \\
fold(\widehat{p}, \widehat{S}) & = & \widehat{S}_1 \cup \widehat{S}_2 \\
\widehat{S}_1 & = & \{\widehat{a}\widehat{b}^+ \mid \exists \widehat{a}, \widehat{b} \in \widehat{P} : \widehat{p} = \widehat{a}\widehat{b} \land \widehat{a}\widehat{b}\widehat{b} \in \widehat{S}\}\} \\
\widehat{S}_2 & = & \widehat{S}/expand(\widehat{S}_1)
\end{array}
$$

$$
fold(s_0 s_1, \{\ldots, s_0 s_1 s_1, \ldots\}) = \{\ldots, s_0 s_1^+ \ldots\}
$$

Abstract Parsing Semantics : re

$$
\begin{aligned}
F &: 2^{\widehat{P}} \to 2^{\widehat{P}} \\
&= \lambda k.((\mathcal{V}^0_{\widehat{P}} e_1 \widehat{\sigma}\widehat{p}) \sqcup (\mathcal{V}^0_{\widehat{P}} e_2 (\widehat{\sigma}[x \mapsto k])\widehat{p}))])
\end{aligned}
$$

$$
\begin{aligned}
C &: (2^{\widehat{P}} \to 2^{\widehat{P}}) \to 2^{\widehat{P}} \to 2^{\widehat{P}} \\
&= \lambda F.\lambda \widehat{S}. \bigsqcup_{\widehat{p} \in \widehat{S}} fold(\widehat{p}, F(\widehat{p})
\end{aligned}
$$

$$
\mathcal{V}^0_{\widehat{P}}(\texttt{re}\ x\ e_1\ e_2\ e_3)\widehat{\sigma} \quad = \quad \mathcal{V}^0_{\widehat{P}} e_3 (\widehat{\sigma}[x \mapsto \lambda \widehat{p}.fixC(F)])
$$

Abstract Parsing Semantics :

$$\widehat{parse\_action} : \widehat{P} \rightarrow Token \rightarrow 2^{\widehat{P}}$$

If

1) the action is REDUCE

2) $Top(\widehat{p}) = \widehat{p'}^{+}$

Then

$$\begin{aligned} \widehat{parse\_action}(\widehat{p}, a) &= \widehat{parse\_action}(Rest(\widehat{p})\widehat{p'}, a) \cup \\ &\quad \widehat{parse\_action}(Rest(\widehat{p})\widehat{p'}^{+}\widehat{p'}, a) \end{aligned}$$

$$\mathcal{V}^{1}_{\widehat{P}} a \widehat{\sigma} = \lambda \widehat{p}.\widehat{parse\_action}(\widehat{p}, a)$$

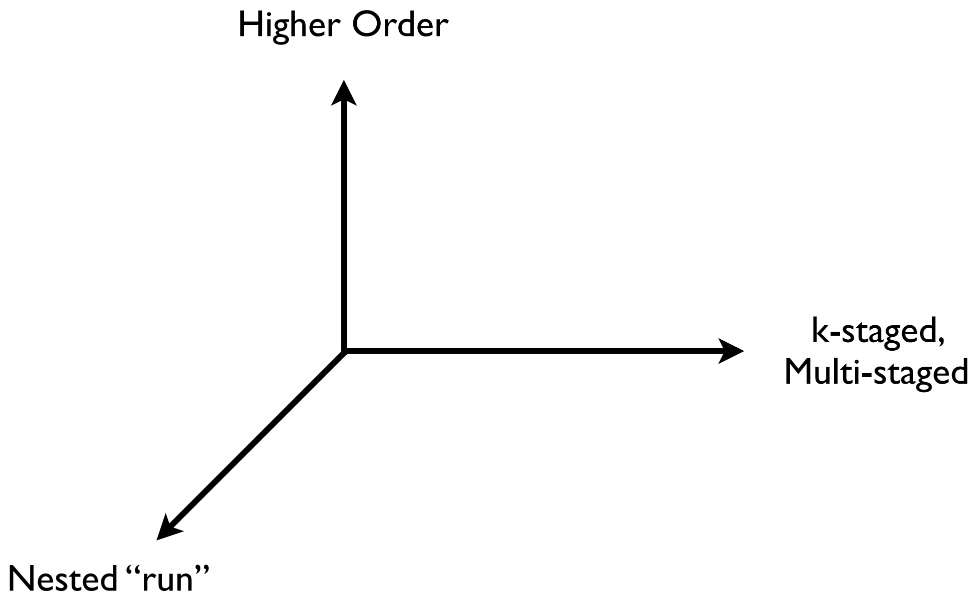
Abstract Parsing Semantics

$$\begin{array}{rcl} \widehat{\sigma} \in Env_{\widehat{P}} & : & Var \to V_{\widehat{P}} \\ \mathcal{V}_{\widehat{P}} & : & Pgm \to Env_{\widehat{P}} \to V_{\widehat{P}} \end{array}$$

$$\begin{array}{rcl}
\mathcal{V}^0_{\widehat{P}} x \widehat{\sigma} & = & \widehat{\sigma}(x) \\
\mathcal{V}^0_{\widehat{P}} (\texttt{or}\ e_1\ e_2) \widehat{\sigma} & = & \lambda \widehat{p}.(\mathcal{V}^0_{\widehat{P}} e_1 \widehat{\sigma} \widehat{p} \cup \mathcal{V}^0_{\widehat{P}} e_2 \widehat{\sigma} \widehat{p}) \\
\mathcal{V}^0_{\widehat{P}} (\texttt{let}\ x\ e_1\ e_2) \widehat{\sigma} & = & \mathcal{V}^0_{\widehat{P}} e_2 (\widehat{\sigma}[x \mapsto \mathcal{V}^0_{\widehat{P}} e_1 \widehat{\sigma}]) \\
\boxed{\mathcal{V}^0_{\widehat{P}} (\texttt{re}\ x\ e_1\ e_2\ e_3) \widehat{\sigma} \quad = \quad \mathcal{V}^0_{\widehat{P}} e_3 (\widehat{\sigma}[x \mapsto \lambda \widehat{p}.fixC(F)])} \\
\mathcal{V}^0_{\widehat{P}} \grave{}\, f \widehat{\sigma} & = & \mathcal{V}^1_{\widehat{P}} f \widehat{\sigma} \\
\mathcal{V}^1_{\widehat{P}} (f_1 . f_2) \widehat{\sigma} & = & \lambda \widehat{p}. \displaystyle\bigcup_{\widehat{p}' \in \mathcal{V}^1_{\widehat{P}} f_1 \widehat{\sigma} \widehat{p}} \mathcal{V}^1_{\widehat{P}} f_2 \widehat{\sigma} \widehat{p}' \\
\mathcal{V}^1_{\widehat{P}} (, e) \widehat{\sigma} & = & \mathcal{V}^0_{P} e \widehat{\sigma} \\
\boxed{\mathcal{V}^1_{\widehat{P}} a \widehat{\sigma} \quad = \quad \lambda \widehat{p}.\widehat{parse_action}(\widehat{p}, a)}
\end{array}$$

Next Step

- k-bound Abstract Parsing
- Introduce non-nested “run”



Thank you