Hat 言語の仕様

島 和之

2023 年 11 月 11 日

Hat は Scheme と同様に静的スコープを持つ動的型付けのプログラミング言語である．ただし，その評価戦略は Scheme とは異なり，名前呼びである．つまり，引数として与えられた式の評価が必要な場合，呼び出された関数で明示的に評価する必要がある．一方，使わない引数を評価する無駄を省くことができる．また，プログラマが遅延評価を用いた独自の制御構造を実現できる．

1 形式的構文

この節では，Hat の形式的構文を示す．記述を簡潔にするため，BNF を以下のように拡張する．

- ⟨thing⟩* は ⟨thing⟩ の 0 個以上の出現を意味する．
- ⟨thing⟩$^+$ は ⟨thing⟩ の 1 個以上の出現を意味する．

上記のように拡張した BNF で，Hat プログラムの構文 ⟨program⟩ の生成規則を以下に示す．

⟨program⟩ → ⟨includer⟩* ⟨definition⟩*
⟨includer⟩ → (include ⟨string⟩)
⟨definition⟩ →
  (define ⟨identifier⟩ ⟨hat expression⟩)
⟨hat expression⟩ → ^ ⟨head⟩ ⟨body⟩
⟨head⟩ → (⟨identifier⟩*)
⟨body⟩ → ⟨operator⟩ ⟨operands⟩
⟨operator⟩ → ⟨identifier⟩ | (⟨hat expression⟩)
  | (⟨operands⟩)
⟨operands⟩ → ⟨expression⟩*
  | ⟨expression⟩* ⟨hat expression⟩
⟨expression⟩ → ⟨simple datum⟩ | '(⟨datum⟩*) |
  (⟨hat expression⟩) | (⟨operands⟩)
⟨simple datum⟩ →
  ⟨number⟩ | ⟨string⟩ | ⟨identifier⟩
⟨datum⟩ → ⟨simple datum⟩ | (⟨datum⟩*)

上記で未定義の記号 ⟨variable⟩, ⟨literal⟩, ⟨lambda expression⟩ の定義は R5RS と同じである．⟨variable⟩ は変数を示す．⟨literal⟩ は真偽値，数値，文字列，quote を付けたシンボルやリストなどを示す．⟨lambda expression⟩ はラムダ式を示す．

2 形式的意味論

この節では Hat プログラムに対する形式的な表示的意味論を定める．以下のように記号を定義する．

I	$\in$	Ide	識別子（変数）
E	$\in$	Exp	式
ρ	$\in$	$U = \text{Ide} \to L$	環境
κ	$\in$	$K = E* \to C$	式の継続

$$
\begin{array}{rcl}
v, v_1, v_2, \ldots & \in & \langle\text{hat variable}\rangle \\
f, f_1, f_2, \ldots & \in & \langle\text{hat function}\rangle \\
c, c_1, c_2, \ldots & \in & \langle\text{hat call}\rangle \\
e, e_1, e_2, \ldots & \in & \langle\text{hat expression}\rangle \\
L, L_1, L_2, \ldots & \in & \langle\text{lambda expression}\rangle
\end{array}
$$

(defineCPS v f) は $(v, f) \in \rho$ を意味する．ここで，ρ は変数に対応する関数を示す環境であり，⟨hat variable⟩ と ⟨hat function⟩ からなる二項組の集合として定義される．以下，$\mathcal{H}[\![e]\!]$ は Hat の式 e に対する数学的意味を返す意味関数，$\mathcal{E}[\![e]\!]$ は Scheme の式 e に対する数学的意味を返す意味関数，$\mathcal{F}[\![e]\!]$ は式 e に含まれる自由変数の集合とする．

⟨hat variable⟩ の意味は次のように定義される．

$$
\mathcal{H}[\![v]\!] = \begin{cases} \mathcal{H}[\![\texttt{(}f\texttt{)}]\!] & \text{if } ((v, \exists f) \in \rho) \\ \text{wrong “undefined variable”} & \\ & \text{otherwise} \end{cases}
$$

⟨hat function⟩ の意味は以下のように定義される．

$$
\mathcal{H}[\![\texttt{(}f\texttt{)}]\!] =
$$

$$
\begin{cases}
\lambda x.x(\lambda v_1.\mathcal{H}[\![\texttt{(\^{}(}v_2\ \cdots\ v_n\texttt{)}\ c\texttt{)}]\!]) & \\
\quad \text{if } (f = \texttt{\^{}(}v_1\ v_2\ \cdots\ v_n\texttt{)}\ c) \land (n \geq 1) & \\
\mathcal{H}[\![\texttt{(}c\texttt{)}]\!] & \text{if } (f = \texttt{\^{}(\ )}\ c) \\
\lambda x.x(\lambda v_1.\mathcal{H}[\![\texttt{(\^{}(}v_2\ \cdots\ v_n\ \texttt{.}\ v_{n+1}\texttt{)}\ c\texttt{)}]\!]) & \\
\quad \text{if } (f = \texttt{\^{}(}v_1\ v_2\ \cdots\ v_n\ \texttt{.}\ v_{n+1}\texttt{)}\ c) & \\
\quad \land (n \geq 1) & \\
\lambda x.x(\lambda v_1.\mathcal{H}[\![\texttt{(\^{}}\ v_2\ c\texttt{)}]\!]) & \\
\quad \text{if } (f = \texttt{\^{}(}v_1\ \texttt{.}\ v_2\texttt{)}\ c) & \\
\lambda v.\mathcal{H}[\![\texttt{(}c\texttt{)}]\!]v & \text{if } (f = \texttt{\^{}}\ v\ c)
\end{cases}
$$

⟨hat call⟩ の意味は以下のように定義される．

$$
\mathcal{H}[\![\texttt{(}c\texttt{)}]\!] =
\begin{cases}
\lambda x.x(\mathcal{E}[\![L]\!]e_1 e_2 \cdots e_n) \\
\qquad \text{if } (c = L \;\; e_1 \;\; e_2 \;\; \cdots \;\; e_n) \\
\mathcal{H}[\![\texttt{((}f\texttt{)} \;\; e_1 \;\; e_2 \;\; \cdots e_n\texttt{)}]\!] \\
\qquad \text{if } (c = v \;\; e_1 \;\; e_2 \;\; \cdots \;\; e_n) \\
\qquad \wedge (n \geq 0) \wedge ((v, \exists f) \in \rho) \\
\mathcal{H}[\![\texttt{((}f\texttt{)} \;\; e_1 \;\; e_2 \;\; \cdots e_{n-1}\texttt{)}]\!](\lambda x.x e_n) \\
\qquad \text{if } (c = \texttt{(}f\texttt{)} \;\; e_1 \;\; e_2 \;\; \cdots \;\; e_n) \wedge (n \geq 1) \\
\mathcal{H}[\![\texttt{(}f\texttt{)}]\!] \qquad \text{if } (c = \texttt{(}f\texttt{)}) \\
\mathcal{H}[\![\texttt{(}c_1\texttt{\^{}(}v\texttt{)} \;\; v \;\; e_1 \;\; e_2 \;\; \cdots e_n\texttt{)}]\!] \\
\qquad \text{if } (c = \texttt{(}c_1\texttt{)} \;\; e_1 \;\; e_2 \;\; \cdots \;\; e_n) \\
\qquad \wedge (n \geq 1) \wedge v \notin \mathcal{F}[\![\texttt{(}c\texttt{)}]\!] \\
\mathcal{H}[\![\texttt{(}c_1\texttt{)}]\!] \qquad \text{if } (c = \texttt{(}c_1\texttt{)}) \\
\lambda x.\mathcal{H}[\![\texttt{(}\kappa_1 \;\; e_1 \;\; e_2 \;\; \cdots \;\; e_n\texttt{)}]\!] x_1 \\
\qquad \text{if } (c = \texttt{(F.C } \kappa_1 \;\; \texttt{.} \;\; x_1\texttt{)} \;\; e_1 \;\; e_2 \;\; \cdots \;\; e_n) \\
\mathcal{H}[\![\texttt{(}e_1 \;\; e_2 \;\; \cdots \;\; e_n \;\; \texttt{.} \;\; \texttt{(}f\texttt{))}]\!] \\
\qquad \text{if } (c = e_1 \;\; e_2 \;\; \cdots \;\; e_n \;\; f) \wedge (n \geq 1) \\
\lambda x.\mathcal{H}[\![\texttt{(}e_1 \;\; e_2 \;\; \cdots \;\; e_n\texttt{)}]\!]\texttt{(F.C } \kappa \;\; \texttt{.} \;\; x\texttt{)} \\
\qquad \text{if } (c = e_1 \;\; e_2 \;\; \cdots \;\; e_n \;\; \texttt{.} \;\; \kappa) \wedge (n \geq 1)
\end{cases}
$$