Snail Programming Language

Nakamura Aoi

2020 年 11 月 28 日

1　はじめに

Snail は静的型付けの関数型プログラミング言語である.

主な特徴として,

- Bounded Linear Type によるリソースの制御
- Effect System / Coeffect System (未実装)
- 軽量な依存型 (indexed type) (未実装)

が挙げられる.

本文では Snail について Core 言語を定義し,Core 言語への脱糖規則,Core 言語の型付け規則, 操作的意味論を定義することにより Snail に定義を与える.

本文中ではメタ変数として以下のようなものを用いる.

- Γ, Δ, Θ $\cdots$ 型環境上を動くメタ変数.
- $A, B \cdots$ 型の上を動くメタ変数.
- $X \cdots$ 型変数上を動くメタ変数.
- $K \cdots$ コンストラクタ上を動くメタ変数.
- $x, y, z \cdots$ 変数上を動くメタ変数.
- $p \cdots$ パターン上を動くメタ変数.
- $r, s, t \cdots$ resource semiring 上を動くメタ変数.
- $e \cdots$ 項の上を動くメタ変数.
- n $\cdots$ 自然数上を動くメタ変数.
- l $\cdots$ バリアントにおけるラベル上を動くメタ変数.
- i, j $\cdots$ メタ自然数上を動くメタ変数.

2　Snail の構文定義

EBNF 記法を用いて Snail の具象構文を以下に示す.

$$
\begin{aligned}
\text{toplevel} ::=\ & \text{let [rec] } x\ \{y\ [\ :\ \langle\text{type}\rangle]\}\!: \langle\text{type}\rangle = \langle\text{term}\rangle\ \{\langle\text{mutual-recursion-let}\rangle\} \\
\mid\ & \text{typedef } A\ = [\ |\ ]\ \{\langle K \text{ [of } \langle\text{type}\rangle]\rangle \mid \}\ \langle K \text{ [of } \langle\text{type}\rangle]\rangle\ \{\langle\text{mutual-recursion-type}\rangle\}
\end{aligned}
$$

$$
\text{mutual-recursion-type} ::= \text{and } A\ = [\ |\ ]\ \{\langle K \text{ [of } \langle\text{type}\rangle]\rangle \mid \}\ \langle K \text{ [of } \langle\text{type}\rangle]\rangle
$$

$$
\begin{aligned}
\text{type} ::=\ & \langle\text{type}\rangle \to \langle\text{type}\rangle \\
\mid\ & !\ \text{'['}\ \langle\text{expmod}\rangle\ \text{']'}\ \text{'\{'}\ \langle\text{type}\rangle\ \text{'\}'} \\
\mid\ & \text{'('}\ \langle\text{type}\rangle\ \text{')'} \\
\mid\ & A
\end{aligned}
$$

$$
\text{expmod} ::= n \mid \infty
$$

$$
\begin{aligned}
\text{pattern} ::=\ & \langle\text{pattern}\rangle \text{ binop } \langle\text{pattern}\rangle \\
\mid\ & \text{'('}\ \langle\text{pattern}\rangle\ \text{')'} \\
\mid\ & x \mid K\ (\{\langle\text{pattern}\rangle\ ,\}\langle\text{pattern}\rangle) \\
\mid\ & \text{list} \qquad \text{(組み込みリストの構文糖衣)} \\
\mid\ & \_
\end{aligned}
$$

$$
\text{mutual-recursion-let} ::= \text{and } x\ \{y\ [\ :\ \langle\text{type}\rangle]\}\ :\ \langle\text{type}\rangle = \langle\text{term}\rangle
$$

$$
\begin{aligned}
\text{term} ::=\ & \langle\text{term}\rangle\ \langle\text{term}\rangle \\
\mid\ & \text{let [rec] } x\ \{y\ [\ :\ \langle\text{type}\rangle]\}\!: \langle\text{type}\rangle = \langle\text{term}\rangle\ \{\langle\text{mutual-recursion-let}\rangle\} \text{ in } \langle\text{term}\rangle \\
\mid\ & \text{fun } \{x\ [\ :\ \langle\text{type}\rangle]\}\ \to \langle\text{term}\rangle \\
\mid\ & \text{case } \langle\text{term}\rangle \text{ of } [\ |\ ]\ \{\langle\text{pattern}\rangle \to \langle\text{term}\rangle \mid \}\ \langle\text{pattern}\rangle \to \langle\text{term}\rangle \\
\mid\ & \text{if } \langle\text{term}\rangle \text{ then } \langle\text{term}\rangle \text{ else } \langle\text{term}\rangle \\
\mid\ & \text{fix } x.\langle\text{term}\rangle \\
\mid\ & \text{fold } t \mid \text{unfold } t \\
\mid\ & \text{'('}\ \langle\text{term}\rangle\ [\ :\langle\text{type}\rangle]\ \text{')'} \\
\mid\ & !\ \langle\text{term}\rangle \\
\mid\ & K\ (\{\langle\text{term}\rangle,\}\langle\text{term}\rangle) \\
\mid\ & x \mid \text{list}
\end{aligned}
$$

3 Snail の Core 言語

Snail の Core 言語は Snail のプログラムを脱糖する事により得ることができる.
Core 言語は次のような性質を持つ.

- パターン p は重複せず全ての条件を網羅している.
- resource semiring として extended natural numbers semiring $(\mathbb{N} \cup \{\infty\}, +, 0, \star, 1, \leq)$ を用いる.
- コンストラクタ K が持つべき型は型環境に追加されている.

3.1 Core 言語の構文

Core 言語は次のような構文を持つ.

$$
\begin{aligned}
e ::=\ & \text{let } x = e_1 \text{ in } e_2 \\
\mid\ & (e_1, e_2, \dots, e_n) \mid x \mid K \\
\mid\ & \text{case } e \text{ of } \{p_i \to e_i\ \}_{i=1}^{n} \\
\mid\ & e_1\ e_2 \mid \lambda x.e \mid \text{fix } x.e \\
\mid\ & \text{fold } t \mid \text{unfold } t
\end{aligned}
$$

$$p ::= K\ (x_1, x_2, \dots, x_n)$$

$$\Gamma ::= \emptyset \mid \Gamma, x \colon A^r$$

$$A^r ::= \mu X.A^r \mid A \xrightarrow{r} A \mid (A \times A \times \cdots \times A)^r \mid \{l_1 \colon A_1 + \cdots + l_n \colon A_n\}^r \mid X^r \mid ()^r$$

A^r の定義に見られるように, 現れる型の resource semiring がどんなものでも良い場合,resource semiring を省略して表記することがある.

3.2 Core 言語の型システム

Core 言語の型付け規則を次に示す.

3.2.1 Context 間の演算の定義

Context 間の加算 $+$ を次のように定義する.

$$
\begin{aligned}
\emptyset + \Delta &= \Delta \\
(x : A^r, \Gamma) + (x : A^s, \Delta) &= x : A^{r+s}, (\Gamma + \Delta) \\
(x : A^r, \Gamma) + \Delta &= x : A^r, (\Gamma + \Delta) \qquad \text{if } x \notin \text{dom}(\Delta)
\end{aligned}
$$

同様に,Context と resource semiring の乗算 $\star$ を次のように定義する.

$$r \star \emptyset = \emptyset$$
$$r \star (x : A^s, \Gamma) = x : A^{r \star s}, r \star \Gamma$$

3.2.2 部分型付け規則

$$\frac{}{A <: A} \quad (\text{O-I}) \qquad \frac{}{\Gamma <: \Gamma} \quad (\text{O-IC})$$

$$\frac{A <: B \quad r \preceq s}{A^s <: B^r} \quad (\text{O-B}) \qquad \frac{\Gamma <: \Delta \quad A <: B}{\Gamma, x : A <: \Delta, x : B} \quad (\text{O-C})$$

$$\frac{A_1 <: A_2 \quad B_1 <: B_2}{A_2 \to B_1 <: A_1 \to B_2} \quad (\text{O-L}) \qquad \frac{A <: B}{\mu X.A <: \mu X.B} \quad (\text{O-R})$$

3.2.3 型付け規則

$$\frac{}{x : A \vdash x : A} \quad (\text{VAR}) \qquad \frac{\Gamma \vdash e : A \xrightarrow{1} B \quad \Delta \vdash e' : A}{\Gamma + \Delta \vdash e\ e' : B} \quad (\text{APP})$$

$$\frac{}{K : A \vdash K : A} \quad (\text{CON}) \qquad \frac{\Gamma, x : A^r \vdash e : A^1 \quad 1 + r \star s \preceq s}{s \star \Gamma \vdash \text{fix } x.e : A^1} \quad (\text{FIX})$$

$$\frac{\Gamma \vdash e : B^1}{r \star \Gamma \vdash e : B^r} \quad (\text{PR}) \qquad \frac{\Gamma \vdash e : A^r \quad \Delta, x : A^r \vdash e' : B}{\Gamma + \Delta \vdash \text{let } x = e \text{ in } e' : B} \quad (\text{LET})$$

$$\frac{\Delta \vdash e : B \quad \Gamma <: \Delta}{\Gamma, \Theta \vdash e : B} \quad (\text{SUB}) \qquad \frac{A = \mu X.B \quad \Gamma \vdash t : [X \mapsto B]A}{\Gamma \vdash \text{fold } t : A} \quad (\text{FOLD})$$

$$\frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash \lambda x.e : A \xrightarrow{1} B} \quad (\text{ABS}) \qquad \frac{A = \mu X.B \quad \Gamma \vdash t : A}{\Gamma \vdash \text{unfold } t : [X \mapsto A]B} \quad (\text{UNFOLD})$$

$$\frac{\Gamma_1 \vdash e_1 : A_1 \quad \Gamma_2 \vdash e_2 : A_2 \ \cdots \ \Gamma_n \vdash e_n : A_n}{\Gamma_1 + \Gamma_2 + \cdots + \Gamma_n \vdash (e_1, e_2, \ldots, e_n) : (A_1 \times A_2 \times \cdots \times A_n)^1} \quad (\text{TUPLE})$$

$$\frac{\Gamma \vdash e : A \quad e \parallel p_i \Rightarrow \Gamma_i \quad \Delta + \Gamma_i \vdash e_i : B}{\Gamma + \Delta \vdash \text{case } e \text{ of } \{p_i \to e_i\}_{i=1}^n : B} \quad (\text{CASE})$$

パターンマッチについての型付け規則を定義するために新しく三項関係 $e \parallel p \Rightarrow \Gamma$ を導入する. これは, パターン p を用いて e を分解したとき, 新たに Context Γ が結果として導入されることを表している.

3.2.4 アルゴリズム的型付け規則

アルゴリズム的型付け規則を与えるために,context difference operator $\div$ を次のように定義する.

$$\Gamma_1, x : A^s, \Gamma_2 \div x = \Gamma_1, \Gamma_2$$

アルゴリズム的型付け規則は,context Γ_1, 項 e, 型 A, 未使用の context Γ_2,resource ring 及び型変数に関する制約 C の関係

$$\Gamma_1 \vdash e \uparrow A \rhd \Gamma_2 \; [C] \qquad\qquad \Gamma_1 \vdash e \downarrow A \rhd \Gamma_2 \; [C]$$

によって表される.

$\uparrow$ は bidirectional typechecking における synthesis mode を $\downarrow$ は check mode を表している.

また, アルゴリズム的型付け規則では resource semiring に結果が 0 以上になる範囲で除算を導入する.

$$\frac{\dfrac{1}{r} \star \Gamma_1 \vdash e_1 \uparrow A_1 \rhd \Gamma_2 \; [C_1] \quad \Gamma_2 \vdash e_2 \uparrow A_2 \rhd \Gamma_3 \; [C_2] \; \cdots \; \Gamma_n \vdash e_n \uparrow A_n \rhd \Gamma_{n+1} \; [C_n]}{\Gamma_1 \vdash (e_1, e_2, \ldots, e_n) \uparrow (A_1 \times A_2 \times \cdots \times A_n)^r \rhd \Gamma_{n+1} \; [C_1 \cup \cdots \cup C_n]} \quad (\textsc{tuple-}\uparrow)$$

$$\frac{\dfrac{1}{r} \star \Gamma_1 \vdash e_1 \downarrow A_1 \rhd \Gamma_2 \; [C_1] \quad \Gamma_2 \vdash e_2 \downarrow A_2 \rhd \Gamma_3 \; [C_2] \; \cdots \; \Gamma_n \vdash e_n \downarrow A_n \rhd \Gamma_{n+1} \; [C_n]}{\Gamma_1 \vdash (e_1, e_2, \ldots, e_n) \downarrow (A_1 \times A_2 \times \cdots \times A_n)^r \rhd \Gamma_{n+1} \; [C_1 \cup \cdots \cup C_n]} \quad (\textsc{tuple-}\downarrow)$$

$$\frac{}{\Gamma_1, x : A^r, \Gamma_2 \vdash x \uparrow A^s \rhd \Gamma_1, x : A^{r-s}, \Gamma_2 \; [\{1 \le s \le r\}]} \quad (\textsc{var-}\uparrow)$$

$$\frac{}{\Gamma_1, x : B^r, \Gamma_2 \vdash x \downarrow A^s \rhd \Gamma_1, x : B^{r-s}, \Gamma_2 \; [\{1 \le s \le r, B^r = A^s\}]} \quad (\textsc{var-}\downarrow)$$

$$\frac{}{\Gamma_1, K : A, \Gamma_2 \vdash K \uparrow A \rhd \Gamma_1, K : A, \Gamma_2 \; [\emptyset]} \quad (\textsc{con-}\uparrow)$$

$$\frac{}{\Gamma_1, K : A, \Gamma_2 \vdash K \downarrow A \rhd \Gamma_1, K : A, \Gamma_2 \; [\emptyset]} \quad (\textsc{con-}\downarrow)$$

$$\frac{\Gamma_1 \vdash e \uparrow A \xrightarrow{t} B \rhd \Gamma_2 \; [C_1] \qquad \Gamma_2 \vdash e' \downarrow A \rhd \Gamma_3 \; [C_2]}{\Gamma_1 \vdash e \; e' \uparrow B \rhd \Gamma_3 \; [C_1 \cup C_2 \cup \{1 \le t\}]} \quad (\textsc{app-}\uparrow)$$

$$\frac{\Gamma_1 \vdash e \ \uparrow\ A \xrightarrow{t} B \rhd \Gamma_2\ [C_1] \qquad \Gamma_2 \vdash e' \ \downarrow\ A \rhd \Gamma_3\ [C_2]}{\Gamma_1 \vdash e\ e' \ \downarrow\ B \rhd \Gamma_3\ [C_1 \cup C_2 \cup \{1 \leq t\}]} \quad (\text{APP-}\downarrow)$$

$$\frac{\begin{array}{c}\mathrm{fresh}(X, \Gamma, C) \qquad \sigma = \mathrm{unify}(C) \qquad \mathrm{NotContainTyvar}(\sigma(X)) \\ A = \sigma(X) \qquad \dfrac{1}{t} \star \Gamma_1, x : X \vdash e \ \uparrow\ B \rhd \Gamma_2\ [C]\end{array}}{\Gamma_1 \vdash \lambda x.e \ \uparrow\ A \xrightarrow{t} B \rhd \Gamma_2 \div x\ [C]} \quad (\text{ABS-}\uparrow)$$

$$\frac{\dfrac{1}{t} \star \Gamma_1, x : A \vdash e \ \downarrow\ B \rhd \Gamma_2\ [C]}{\Gamma_1 \vdash \lambda x.e \ \downarrow\ A \xrightarrow{t} B \rhd \Gamma_2 \div x\ [C]} \quad (\text{ABS-}\downarrow)$$

$$\frac{\Gamma_1 \vdash e \ \uparrow\ A \rhd \Gamma_2\ [C_1] \qquad \Gamma_2, x : A \vdash e' \ \uparrow\ B \rhd \Gamma_3\ [C_2]}{\Gamma_1 \vdash \mathrm{let}\ x = e\ \mathrm{in}\ e' \ \uparrow\ B \rhd \Gamma_3 \div x\ [C_1 \cup C_2]} \quad (\text{LET-}\uparrow)$$

$$\frac{\Gamma_1 \vdash e \ \uparrow\ A \rhd \Gamma_2\ [C_1] \qquad \Gamma_2, x : A \vdash e' \ \downarrow\ B^s \rhd \Gamma_3\ [C_2]}{\Gamma_1 \vdash \mathrm{let}\ x = e\ \mathrm{in}\ e' \ \downarrow\ B \rhd \Gamma_3 \div x\ [C_1 \cup C_2]} \quad (\text{LET-}\downarrow)$$

$$\frac{\dfrac{1}{s} \star \Gamma_1, x : A^r \vdash e \ \downarrow\ A^1 \rhd \Gamma_2\ [C \cup \{\frac{1-s}{s} \leq r\}]}{\Gamma_1 \vdash \mathrm{fix}\ x.e \ \downarrow\ A^s \rhd \Gamma_2\ [C \cup \{\frac{1-s}{s} \leq r\}]} \quad (\text{FIX-}\downarrow)$$

アルゴリズム的型付け規則を与えるために導入した述語及び関数の定義を示す.

- $\mathrm{fresh}(X, \Gamma, C) \cdots X$ はΓにも C にも出現しない.
- $\mathrm{NotContainTyvar}(A) \cdots A$ には型変数が含まれない.
- $\mathrm{unify}(C) \cdots\ C$ を単一化し, 得られた代入を返す.

3.2.5 アルゴリズム的型付けの例

Cons は型 K についての値コンストラクタであり, 型 $A^1 \times A^1 \to K^1$ を持つとする.

- $\lambda x.\mathrm{Cons}(x, x)$

$$
\cfrac{\cfrac{\mathrm{TC}(\mathrm{Cons}) = A^1 \times A^1 \to K^1 \qquad \cfrac{\begin{array}{c} x : X_1^{\frac{r_1}{s*t}} \vdash x \downarrow A^1 \rhd x : X_1^{\frac{r_1}{s*t}-1}\ [\{1 \leq \frac{r_1}{s*t}, X_1 = A\}] \\ x : X_1^{\frac{r_1}{s*t}-1} \vdash x \downarrow A^1 \rhd x : X_1^{\frac{r_1}{s*t}-2}\ [\{1 \leq \frac{r_1}{s*t} - 1, X_1 = A\}] \end{array}}{x : X_1^{\frac{r_1}{s*t}} \vdash (x, x) \downarrow A^1 \times A^1 \rhd x : X_1^{\frac{r_1}{s*t}-2}\ [\{1 \leq \frac{r_1}{s*t}, 1 \leq \frac{r_1}{s*t} - 1, X_1 = A\}]}}{x : X_1^{r_1} \vdash \mathrm{Cons}(x, x) \uparrow K^s \rhd x : X_1^{\frac{r_1}{s*t}-2}\ [\{1 \leq \frac{r_1}{s*t}, 1 \leq \frac{r_1}{s*t} - 1, X_1 = A\}]}}{\vdash \lambda x.\mathrm{Cons}(x, x) \uparrow (A^{r_1} \to K^s)^t \rhd \emptyset\ [\{1 \leq \frac{r_1}{s*t}, 1 \leq \frac{r_1}{s*t} - 1, X_1 = A\}]}
$$