

ソフトウェア構成特論 第12回

大学院理工学研究科 電気電子情報工学専攻 篠埜 功

1 はじめに

単純型付きラムダ計算にさらに構文を追加していく。

2 組 (Tuple)

対 (pair) では2つの式の組み合わせだったが、これを n 個の組み合わせにしたものが組 (tuple) である。例えば、 $\{1, 2, \text{true}\}$ は3つ組であり、その型は $\{\text{Nat}, \text{Nat}, \text{Bool}\}$ と書かれる。

t_1 から t_n の n 個の組を $\{t_i^{i \in 1 \dots n}\}$ と書き、型は $\{T_i^{i \in 1 \dots n}\}$ と書く。 n は0でもよいとし、その場合は $\{t_i^{i \in 1 \dots n}\}$ は空の組 $\{\}$ である。また、1つの要素からなる組、例えば $\{5\}$ は、5とは異なる。

組をこれまでの計算体系に追加する。まず、term の定義に組と projection を追加する。

$$t ::= \dots \mid \{t_i^{i \in 1 \dots n}\} \mid t.i$$

値に、値の組を追加する。

$$v ::= \dots \mid \{v_i^{i \in 1 \dots n}\}$$

型に組型を追加する。

$$T ::= \dots \mid \{T_i^{i \in 1 \dots n}\}$$

評価規則に3つの規則を追加する。

$$\frac{}{\{v_i^{i \in 1 \dots n}\}.j \rightarrow v_j} \text{ (E-PROJTUPLE)} \quad \frac{t \rightarrow t'}{t.i \rightarrow t'.i} \text{ (E-PROJ)}$$
$$\frac{t_j \rightarrow t'_j}{\{v_i^{i \in 1 \dots j-1}, t_j, t_k^{k \in j+1 \dots n}\} \rightarrow \{v_i^{i \in 1 \dots j-1}, t'_j, t_k^{k \in j+1 \dots n}\}} \text{ (E-TUPLE)}$$

型付け規則については以下の2つの規則を追加する。

$$\frac{\text{for each } i \quad \Gamma \vdash t_i : T_i}{\Gamma \vdash \{t_i^{i \in 1 \dots n}\} : \{T_i^{i \in 1 \dots n}\}} \text{ (T-TUPLE)} \quad \frac{\Gamma \vdash t : \{T_i^{i \in 1 \dots n}\}}{\Gamma \vdash t.j : T_j} \text{ (T-PROJ)}$$

置換の定義に組と組の projection の場合を追加する。

$$[x \mapsto s]\{t_i^{i \in 1 \dots n}\} = \{([x \mapsto s]t_i)^{i \in 1 \dots n}\}$$
$$[x \mapsto s](t.i) = ([x \mapsto s]t).i$$

練習問題 1. 型判定 $\vdash \{\text{true}, \text{pred}(\text{succ } 0), \text{false}\}.2 : \text{Nat}$ の導出木を書け。

練習問題 2. term $\{\text{true}, \text{pred}(\text{succ } 0), \text{false}\}.2$ を評価せよ。各1ステップ評価の導出木も書け。

3 レコード (Record)

レコードは、組の各要素にラベルが付いたものである。例えば、 $\{x=5\}$ や $\{\text{partno}=5524, \text{cost}=30.27\}$ はレコードであり、これらの型は $\{x:\text{Nat}\}$ 、および $\{\text{partno}:\text{Nat}, \text{cost}:\text{Float}\}$ である。一つのレコード内のラベルは異ならなければならない。

レコードをこれまでの計算体系に追加する。まず、term にレコードとレコードの projection を追加する。

$$t ::= \dots \mid \{l_i = t_i^{i \in 1 \dots n}\} \mid t.l$$

値に、レコード値を追加する。

$$v ::= \dots \mid \{l_i = v_i^{i \in 1 \dots n}\}$$

型にレコード型を追加する。

$$T ::= \dots \mid \{l_i : T_i^{i \in 1 \dots n}\}$$

評価規則に3つの規則を追加する。

$$\frac{}{\{l_i = v_i^{i \in 1 \dots n}\}.l_j \rightarrow v_j} \text{ (E-PROJRCD)} \quad \frac{t \rightarrow t'}{t.l \rightarrow t'.l} \text{ (E-PROJ)}$$

$$\frac{t_j \rightarrow t'_j}{\{l_i = v_i^{i \in 1 \dots j-1}, l_j = t_j, l_k = t_k^{k \in j+1 \dots n}\} \rightarrow \{l_i = v_i^{i \in 1 \dots j-1}, l_j = t'_j, l_k = t_k^{k \in j+1 \dots n}\}} \text{ (E-RCD)}$$

型付け規則については以下の2つの規則を追加する。

$$\frac{\text{for each } i \quad \Gamma \vdash t_i : T_i}{\Gamma \vdash \{l_i = t_i^{i \in 1 \dots n}\} : \{l_i : T_i^{i \in 1 \dots n}\}} \text{ (T-RCD)} \quad \frac{\Gamma \vdash t : \{l_i : T_i^{i \in 1 \dots n}\}}{\Gamma \vdash t.l_j : T_j} \text{ (T-PROJ)}$$

置換の定義にレコード、レコードの projection の場合を追加する。

$$\begin{aligned} [x \mapsto s]\{l_i = t_i^{i \in 1 \dots n}\} &= \{l_i = [x \mapsto s]t_i^{i \in 1 \dots n}\} \\ [x \mapsto s](t.l) &= ([x \mapsto s]t).l \end{aligned}$$

練習問題 3. 型判定 $\vdash \{a=\text{true}, b=\text{pred}(\text{succ } 0)\}.b : \text{Nat}$ の導出木を書け。

練習問題 4. term $\{a=\text{true}, b=\text{pred}(\text{succ } 0)\}.b$ を評価せよ。各1ステップ評価の導出木も書け。

レコード内の要素の順番については、言語によって扱いが異なる。ある言語では、順番は意味に影響しない、つまり、例えば $\{\text{partno}=5524, \text{cost}=30.27\}$ と $\{\text{cost}=30.27, \text{partno}=5524\}$ は同じレコードであり、同じ型 $\{\text{partno}:\text{Nat}, \text{cost}:\text{Float}\}$ を持つ。また、順番が意味に影響する言語もある。この講義では、順番が異なるレコードは値も型も異なるものとする。

4 直和 (Sum)

直和型 (sum type) は、2つの型の値を合わせた集合である。例えば、住所録の各データの型として以下の2つを使っているとする。

```
PhysicalAddr = {firstlast:String, addr:String}
VirtualAddr = {name:String, email:String}
```

両方の型のデータを含むリストを作りたいような場合、以下のような直和型を用いることができる。

```
Addr = PhysicalAddr + VirtualAddr
```

この直和型 Addr の要素は、PhysicalAddr 型と VirtualAddr 型の要素に inl, inr というタグをつけたものである。例えば、{firstlast:IsaoSasano, addr:Tokyo} は PhysicalAddr 型であり、それに inl というタグをつけた inl {firstlast:IsaoSasano, addr:Tokyo} は Addr 型である。また、{name:Sasano, email:sasano@tokyo} は VirtualAddr 型であり、それに inr というタグをつけた inr {name:Sasano, email:sasano@tokyo} は Addr 型である。

一般に、 T_1+T_2 型の要素は T_1 型の要素に inl というタグを付けたものと T_2 型の要素に inr というタグを付けたものから成る。直和型の要素を使うときには、case 式を用いる。例えば、Addr 型の要素から名前の部分を抽出したい場合は以下のような関数を用いればよい。

```
getName = λa:Addr.
  case a of
    inl x ⇒ x.firstlast
  | inr y ⇒ y.name
```

以上のような直和型をこれまでの計算体系に追加する。まず、term の定義にタグ付けと case 式を追加する。

$$t ::= \dots \mid \text{inl } t \mid \text{inr } t \mid \text{case } t \text{ of inl } x \Rightarrow t \mid \text{inr } x \Rightarrow t$$

値に、タグ付きの値を追加する。

$$v ::= \dots \mid \text{inl } v \mid \text{inr } v$$

型に直和型を追加する。

$$T ::= \dots \mid T + T$$

評価規則に 5 つの規則を追加する。

$$\frac{}{\text{case (inl } v) \text{ of inl } x_1 \Rightarrow t_1 \mid \text{inr } x_2 \Rightarrow t_2 \rightarrow [x_1 \mapsto v]t_1} \text{ (E-CASEINL)}$$

$$\frac{}{\text{case (inr } v) \text{ of inl } x_1 \Rightarrow t_1 \mid \text{inr } x_2 \Rightarrow t_2 \rightarrow [x_2 \mapsto v]t_2} \text{ (E-CASEINR)}$$

$$\frac{t_0 \rightarrow t'_0}{\text{case } t_0 \text{ of inl } x_1 \Rightarrow t_1 \mid \text{inr } x_2 \Rightarrow t_2 \rightarrow \text{case } t'_0 \text{ of inl } x_1 \Rightarrow t_1 \mid \text{inr } x_2 \Rightarrow t_2} \text{ (E-CASE)}$$

$$\frac{t \rightarrow t'}{\text{inl } t \rightarrow \text{inl } t'} \text{ (E-INL)} \quad \frac{t \rightarrow t'}{\text{inr } t \rightarrow \text{inr } t'} \text{ (E-INR)}$$

型付け規則については以下の 3 つの規則を追加する。

$$\frac{\Gamma \vdash t : T_1}{\Gamma \vdash \text{inl } t : T_1+T_2} \text{ (T-INL)} \quad \frac{\Gamma \vdash t : T_2}{\Gamma \vdash \text{inr } t : T_1+T_2} \text{ (T-INR)}$$

$$\frac{\Gamma \vdash t_0 : T_1+T_2 \quad \Gamma, x_1 : T_1 \vdash t_1 : T \quad \Gamma, x_2 : T_2 \vdash t_2 : T}{\Gamma \vdash \text{case } t_0 \text{ of inl } x_1 \Rightarrow t_1 \mid \text{inr } x_2 \Rightarrow t_2 : T} \text{ (T-CASE)}$$

練習問題 5. 型判定 $\vdash \text{let } x = \text{inl true in case } x \text{ of inl } y \Rightarrow y \mid \text{inr } z \Rightarrow z : \text{Bool}$ の導出木を書け。

練習問題 6. term $\text{let } x = \text{inl true in case } x \text{ of inl } y \Rightarrow y \mid \text{inr } z \Rightarrow z$ を評価せよ。各 1 ステップ評価の導出木も書け。

注意 置換の定義は term の定義が拡張されたら定義しなおす必要がある (が省略している)。case 式の置換において、 $\text{inl } x \Rightarrow t$ や $\text{inl } x \Rightarrow t$ の部分は、ラムダ抽象 $\lambda x.t$ と同じようにする必要がある。case 式の $\text{inl } x \Rightarrow t$ の部分において x はここで導入される変数であり、 t の中でのみ有効である (inr も同様)。よって、ラムダ抽象の場合と同様、必要なら変数名の付け替えを行うものとする。