Oishi Type System に対する型制約「解消」アルゴリズム

2017/1/21

1　解きたい型制約

以下の構文を持つ classifier 式を考える。

$$e ::= d \mid \gamma_x \mid e \cup e$$

ここで、d は定数 (固有変数条件をみたす classifier 変数), γ_x は classifier をあらわす変数、e は classifier 式。(Oishi 論文では、classifier 式は γ と書いていたが、γ_x と紛らわしいので、ここでは e と書くことにする。) また、型推論の対象となった式の一番外側での classifier も定数と考えられるので、それを d_0 とする。

解きたい制約は、以下の形である。

$$\Delta \models C$$

ただし、 Δ は、$d \geq e$ の形の不等式を有限個「かつ」でつなげたものである。また、すべての式 d_i $(i > 0)$ に対して $d_i \geq d_o$ が Δ に含まれているとしてよい。また、C は、$e \geq e$ の形の不等式を「かつ」でつなげたものである。

なお、仮定 Δ は、(code-lambda ルール等で) 固有変数を導入する際の $d \geq e$ という形の不等式のみがはいっているので、左辺が必ず d であることに注意せよ。

制約解消問題 $\Delta \models C$ に対して、$[\gamma_x := e, \cdots, \gamma_x := e]$ という形の代入 θ で、$\Delta\theta \models C\theta$ が (順序に関する規則のみで) 導けるとき、この θ を解と呼ぶ。そのような θ が存在しないとき、解がないという。

2　代数的な準備

Upper Semilattice (上むき半束) と Join-Irreducible いう言葉を導入する。

まず、lattice(束) とは、順序集合で、$\cap$ と $\cup$ があるものである。今回の型推論では、$\cup$ はあるが $\cap$ はないので、latice ではない。「上の方向だけ (半分)、lattice の構造がある」という意味で、semilattice (半束) あるいは、もっと正確には upper semilattice(上向き半束) という言葉を使う。

要素 e が、join-irreducible (join について還元可能でない) であるとは、

ほかのどんな要素 e_1, e_2 に対しても、$e_1 \cup e_2 \geq e$ ならば、$e_1 \geq e$ または $e_2 \geq e$ であることを言う。[*1]

今回型推論をおこなう対象の構造は、Upper Semilattice であって、かつ、すべての clasifier 定数 d(もともとは固有変数だったもの) が join-irreducible としてよい。(理由： Δ は $d \geq e$ の形だけからなる。Δ から $e_1 \cup e_2 \geq d$ を導く証明のサイズに関する帰納法で、Δ から $e_1 \geq d$ または $e_2 \geq d$ が導けることが言える。

3　制約解消アルゴリズム

Δ と C と代入 θ に対して、以下の手順を適用する。代入 θ の初期値は、空代入とする。

- C が空であるならば、θ を返して終了する。

[*1] ただし、こまかいことだが、最小元である d_0 は、通常は、join-irreducible とは言わない。

- $(e \geq e) \in C$ (両辺が同じ不等式) ならば、C から、この不等式を除去して再帰する。
- $(d_1 \geq d_2) \in C$ (両辺が異なる classifier 定数) のとき、さらに、Δ のもとで、$d_1 \geq d_2$ が導けるのであれば、この不等式を除去して再帰する。導けないのであれば、「制約解消失敗」を返して終了する。
- $e_1 \geq e_2 \cup e_3) \in C$ ならば、C から $e_1 \geq e_2 \cup e_3$ を除去し、$e_1 \geq e_2$ と $e_1 \geq e_3$ を追加して再帰する。
- $(\cdots \cup \gamma_x \cup \cdots) \in \gamma_x$ ならば、この不等式を除去して再帰する。
- $(e_1 \cup e_2 \geq d) \in C$ ならば、join-irreducible の仮定 (*) を用いて、$e_1 \geq d$ または $e_2 \geq d$ である。よって、$C - \{(e_1 \cup e_2 \geq d)\} \cup \{(e_1 \geq d)\}$ に対して再帰し、さらに、$C - \{(e_1 \cup e_2 \geq d)\} \cup \{(e_2 \geq d)\}$ に対して再帰し、それらの解の両方を解として返す。(一意的には解けない。解は一般的には複数ある。)
- 上記のいずれのケースもあてはまらないが、C が空でないときは、残った classifier 変数 γ_x をひとつ取り出し、C の中から以下の形の不等式をすべて列挙する。

$$
\begin{aligned}
e_i &\geq \gamma_x & \text{for } i = 1, 2, , ...m \\
\gamma_x &\geq d_j & \text{for } j = 1, 2, , ...n
\end{aligned}
$$

  ただし m や n は 0 かもしれない。また、上記の変形が終了しているので、すべての e_i は γ_x は含まないことに注意。
  - $m = 0$ かつ $n > 0$ のとき、$\gamma_x := d_1 \cup \cdots \cup d_n$ という代入を θ に追加して、C からは γ_x に関する不等式をすべて取り除いて再帰する。
  - $m > 0$ かつ $n = 0$ のとき、$\gamma_x := d_0$ という代入を θ に追加して、C からは γ_x に関する不等式をすべて取り除いて再帰する。
  - $m > 0$ かつ $n > 0$ のとき、C から γ_x に関する不等式をすべて取り除いた上で、以下の不等式を追加する。

$$e_i \geq d_j \quad (\text{すべての } i, j \text{ に対して})$$

    さらに、$\gamma_x := d_1 \cup \cdots \cup d_n$ という代入を θ に追加した上で再帰する。

上記の手順を繰返すと (この繰返しは、かならず停止することがいえる)、最終的に、(途中で fail しないかぎり)γ_x 変数がすべて消去され、制約 C が空になり、制約を満たす解の有無が判定できる。