

プログラム言語論

亀山幸義

筑波大学 情報科学類

No. 6b: 単一化についての補足

① 単一化アルゴリズム

型に関する方程式

型推論問題は、以下のような問題に帰着された:

問題 (例): 以下の等式すべてが成立する型 τ_1, \dots, τ_9 は存在するか?

$$\tau_1 = \tau_2 \rightarrow \tau_3$$

$$\tau_2 \rightarrow \text{int} = \text{int} \rightarrow \tau_3$$

$$\tau_4 = \tau_3 \rightarrow \tau_1$$

単一化問題

型に関する変数を含む等式の集合に対して、全ての等式が成立する「解」が存在するかどうか。

問題:

$$\tau_1 = \tau_2 \rightarrow \tau_3$$

$$\tau_2 \rightarrow \text{int} = \text{int} \rightarrow \tau_3$$

$$\tau_4 = \tau_3 \rightarrow \tau_1$$

解: 上記の等式集合は解が存在して、その1つは以下の通り。

$$\tau_1 := \text{int} \rightarrow \text{int}$$

$$\tau_2 := \text{int}$$

$$\tau_3 := \text{int}$$

$$\tau_4 := \text{int} \rightarrow (\text{int} \rightarrow \text{int})$$

解が存在しない問題もある。

▶ 問題: $\tau_1 = \text{int}$, $\tau_1 = \text{int} \rightarrow \tau_2$

単一化アルゴリズム

入力: 等式の集合 E

出力: 型変数に対する型の代入の集合 σ (初期値は空)

- ▶ $E = \{\}$ なら終了。解が存在して σ である。
- ▶ $E \neq \{\}$ なら等式 $S = T \in E$ を取りだす。
- ▶ S と T が同一の式なら、 $E - \{S = T\}$ を新しい E として、最初に戻る。
- ▶ S が型変数、 S が T に出現すれば、単一化失敗。
- ▶ S が型変数、 S が T に出現しなければ、 $\sigma[S := T]$ を新しい σ とし、 $E - \{S = T\}$ に $S := T$ を適用したものを新しい E として、最初に戻る。
- ▶ T が型変数、 S が型変数でないなら、 $E - \{S = T\} \cup \{T = S\}$ を新しい E として、最初に戻る。
- ▶ $S = \text{int}$ かつ T が int 以外の式なら、単一化失敗。
- ▶ $T = \text{int}$ かつ S が int 以外の式なら、単一化失敗。
- ▶ $S = S_1 \rightarrow S_2$ かつ $T = T_1 \rightarrow T_2$ なら、 $E - \{S = T\} \cup \{S_1 = T_1, S_2 = T_2\}$ を新しい E として、最初に戻る。

単一化アルゴリズムの実行例 (1)

$$E = \{\tau_1 \rightarrow \text{int} = \tau_2 \rightarrow \tau_1\}$$

- ▶ E から、等式 $\tau_1 \rightarrow \text{int} = \tau_2 \rightarrow \tau_1$ を取り出す。
- ▶ 等式の左右両辺が関数型なので、それを分解して、 $\{\tau_1 = \tau_2, \text{int} = \tau_1\}$ を新しい E とする。
- ▶ E から、等式 $\tau_1 = \tau_2$ を取り出す。
- ▶ 等式の左辺が型変数で、 τ_2 に τ_1 は出現しないので、 $[\tau_1 := \tau_2]$ を新しい σ とする。 $\{\text{int} = \tau_1\}$ に σ を適用すると、 $\{\text{int} = \tau_2\}$ となるので、これを新しい E とする。
- ▶ E から、等式 $\text{int} = \tau_2$ を取り出す。
- ▶ 等式の右辺が型変数で、 int に τ_2 は出現しないので、 $[\tau_1 := \tau_2]$ に $[\tau_2 := \text{int}]$ を加える。つまり、 $[\tau_1 := \text{int}, \tau_2 := \text{int}]$ を新しい σ とする。残っている等式はないので、 $E = \{\}$ となる。
- ▶ $E = \{\}$ なので、 $[\tau_1 := \text{int}, \tau_2 := \text{int}]$ が解の 1 つである。

単一化アルゴリズムの実行例 (2)

$E = \{\tau_1 \rightarrow \text{int} = (\tau_1 \rightarrow \text{int}) \rightarrow \tau_1\}$

- ▶ E から、等式 $\tau_1 \rightarrow \text{int} = (\tau_1 \rightarrow \text{int}) \rightarrow \tau_1$ を取り出す。
- ▶ 等式の左右両辺が関数型なので、それを分解して、 $\{\tau_1 = \tau_1 \rightarrow \text{int}, \text{int} = \tau_1\}$ を新しい E とする。
- ▶ E から、等式 $\tau_1 = \tau_1 \rightarrow \text{int}$ を取り出す。
- ▶ 等式の左辺が型変数で、 $\tau_1 \rightarrow \text{int}$ に τ_1 は出現するので、単一化に失敗する。

別の順番でアルゴリズムを走らせる。

- ▶ E から、等式 $\tau_1 \rightarrow \text{int} = (\tau_1 \rightarrow \text{int}) \rightarrow \tau_1$ を取り出す。
- ▶ 等式の左右両辺が関数型なので、それを分解して、 $\{\tau_1 = \tau_1 \rightarrow \text{int}, \text{int} = \tau_1\}$ を新しい E とする。
- ▶ E から、等式 $\text{int} = \tau_1$ を取り出す。
- ▶ 等式の右辺が型変数で、 int に τ_1 は出現しないので、 $[\tau_1 := \text{int}]$ を新しい σ とする。 $\{\tau_1 = \tau_1 \rightarrow \text{int}\}$ に σ を適用すると、 $\{\text{int} = \text{int} \rightarrow \text{int}\}$ となるので、これを新しい E とする。
- ▶ E から、等式 $\text{int} = \text{int} \rightarrow \text{int}$ を取り出す。
- ▶ 等式の左辺が int で右辺が int 以外なので単一化に失敗する。

単一化アルゴリズムの性質

- ▶ 単一化問題 E に解が存在することと、単一化アルゴリズムを E に適用した時に実行が成立するかどうかは、同値である。
- ▶ 単一化アルゴリズムを E に適用した時に得られる解 (代入) σ は、単一化問題 E の解となっている。
- ▶ 単一化アルゴリズムを E に適用した時に得られる解 (代入) σ は、単一化問題 E の解の中で、最も一般的 (most general) な解である。