# 分散 JoinJAVA プログラムの正常実行性判定のための型システム

佐伯昌樹 *，坂部俊樹，酒井正彦，草刈圭一朗，西田直樹 （名古屋大学）

Type Judgement System for Assuring Distributed JoinJAVA Programs Run Normally

Saeki Masaki, Sakabe Toshiki, Sakai Masahiko, Kusakari Keiichirou, Nishida Naoki （Nagoya University）

## 1. はじめに

マルチスレッドプログラミングが可能な言語である JAVA 言語では，マルチスレッドプログラミングを行うために必要な，同期や通信に用いる命令は単純なものしか用意されていない．この問題を解決するため，同期や通信などの記述に優れた Join 算法 [1] と JAVA 言語を融合した言語である JoinJAVA[4] が提案されている．

本稿では，複数の計算機間でプロセスのやり取りをする仕組み [2] を JoinJAVA に導入するとともに，型および型判定システムを [3] に基づき与える．そして，その型判定システムを用いることでプログラムが異常停止しないと保証できることを示す．

## 2. 分散 JoinJAVA

JoinJAVA[4] は Join 算法の構文，意味論等の枠組みに基づき，JAVA 言語のプログラムを記述できるように拡張が行われた言語である．したがって，Join 算法の枠組みに従う形で JoinJAVA に分散環境を導入する．分散 JoinJAVA の構文を以下のように与える．

**定義 1 (構文)**

$$
\begin{array}{lcl@{\qquad}lcl}
S & \stackrel{\text{def}}{=} & \texttt{loc}\ \alpha[D{:}P]^{\Delta,I,F} & D & \stackrel{\text{def}}{=} & D\ \texttt{and}\ D' \\
 & | & S||S' & & | & J{=}P \\
P & \stackrel{\text{def}}{=} & P|P' & & | & \texttt{loc}\ a[D{:}P]^{\Delta,I} \\
 & | & n\langle\tilde{\text{n}}\rangle;P & J & \stackrel{\text{def}}{=} & n\langle\tilde{y}\rangle \\
 & | & \texttt{go}\ a;P & & | & J|J' \\
 & | & \texttt{def}\ D\ \texttt{in}\ P & \text{n} & \stackrel{\text{def}}{=} & n \\
 & | & \texttt{newch}\ n.P & & | & a \\
 & | & \text{JAVAterm};P & & | & y
\end{array}
$$

分散 JoinJAVA のプログラムは複数のロケーションの並列構造 $S$ からなり，いくつかのロケーションで 1 つの計算機を表す．各ロケーションは，ロケーション名の系列 $\alpha$，プロセス $P$，反応規則 $D$ 等で構成される．プロセスのアクションとしては，ロケーションを移動するための命令 **go** やローカルなチャネルを生成するための命令 **newch** 等が含まれる．また，JAVA のプログラムが JAVAterm として記述され，spawn 命令により，JAVA のプログラム中にも Join 算法のメッセージを記述出来る．反応規則 $D$ はロケーション定義，Join パターン ($J = P$) を **and** で結んだ並列構造を表す．Join パターンはチャネル $n$ を用いてプロセス同士で $n\langle\tilde{\text{n}}\rangle$ のようなメッセージの通信を行う為のマッチング規則のことである．

また，Join 算法に基づいたプログラムの動作を表す形式的な意味論を与える．この意味論では，プログラムの各部分項に対する解釈が定義されている．その 1 つを次に示す．

$$
\frac{dom(\sigma_{rn}) = rn(J)}{\mathbf{loc}\ \alpha : a[\mathcal{D}\ \mathbf{and}\ J = P : \mathcal{P} \mid a.J\sigma_{rn}]^{\Delta,I,F} \\ \rightarrow\ \mathbf{loc}\ \alpha : a[\mathcal{D}\ \mathbf{and}\ J = P : \mathcal{P} \mid P\sigma_{rn}]^{\Delta,I,F}}\ [\text{JOIN}]
$$

[JOIN] は Join パターンとプロセスのマッチにより，プロセスが置き換えれることを表す．

## 3. 分散 JoinJAVA における型と型判定システム

分散 JoinJAVA プログラム中の名前がチャネルであるのか，ロケーションであるのかという情報を型として表現する．この名前と型の対の集合を型環境 $\Gamma$ という．そして，この $\Gamma$ に従って，プログラムの一部の項 $\mathcal{S}$ が $\Gamma$ で示されている型と合っているかを判定するための型判定規則を与える．この規則は定義 1 で示した JoinJAVA の構文ごとに定義され，以下にその 1 つを示す．ある $\Gamma$ の下で $\Gamma \vdash \mathcal{S}$ と書かれた場合，項 $\mathcal{S}$ が規則に従って正しく型判定されたことを表すが，これは規則を再帰的に用いることにより判定される．

$$
\frac{\Gamma \vdash \text{JAVAterm} : jsta \quad Q = Spawned(\text{JAVAterm}) \qquad \Delta;\mathbf{I};\Gamma \vdash Q \quad \Delta;\mathbf{I};\Gamma \vdash P}{\Delta;\mathbf{I};\Gamma \vdash \text{JAVAterm};P}\ [\text{JAVA}]
$$

## 4. 分散 JoinJAVA プログラムの異常停止性

プログラムを実行するとき，予定外の状況で停止することがある．その原因となりうる状況を以下のように定義する．

**定義 2** プログラム $\mathcal{S}$ が以下の条件のいずれかを満たすとき，プログラム $\mathcal{S}$ は異常停止するという．

1. チャネルでない名前がチャネルとして扱われている．
2. ある名前の型がチャネル，ロケーション，JAVA のいずれの型でもない．
3. 命令 **go** $n$ において，$n$ がロケーションでない．
4. 同じ名前のチャネルの引数の数が異なる．
5. すべてのチャネルには対応するチャネル定義が存在する．

プログラム $\mathcal{S}$ が異常停止しないとき，正常実行性を持つという．

**予想 1** $\mathcal{S}$ をプログラムとし，$\Gamma$ を型環境とする．このとき，$\Gamma \vdash \mathcal{S}$ であるならば，$\mathcal{S}$ は正常実効性を持つ．

この予想の証明は一部未完成であり，完全な証明を与えることは今後の課題である．

文　献


[1] C.Fournet, G.Gonthier : The reflexive CHAM and the join-calculus, Principles of Programing Language, 1995.

[2] C.Fournet G.Gonthier J.J.Levy L.Maranget D.Remy : A Calculus of Mobile Agents, CONCUR, 1996.

[3] A.Schmitt : Safe Dynamic Binding in the Join Calculus, IFIP TCS, pp.563--575, 2002.

[4] 尾関 嘉一郎: Join 算法による並行計算の簡潔な記述が可能な JAVA 言語, 名古屋大学大学院 工学研究科 修士学位論文, 2000.