

## 平成 15 年度 情報工学コース卒業研究報告要旨

坂部・酒井 研究室	氏 名	佐伯 昌樹
卒業研究題目	Join 算法のスコープルールに対応した JoinJAVA コンパイラ	
<p>マルチスレッドプログラミングが可能な言語として JAVA 言語がある。しかしこの JAVA 言語において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おらず、この記述を簡潔にするために同期や通信などの記述に優れた Join 算法と JAVA 言語との融合言語である JoinJAVA 言語が提案されている。これは同期や通信といった部分の記述を Join 算法を用い、それ以外の部分の記述を JAVA 言語を用いて行うものである。またこの言語を実装し実行するために JoinJAVA コンパイラが作成されている。このコンパイラは非同期チャンネルを用いた通信しか行うことができない。そのため値を返すという働きができず、関数呼び出しや実行結果の返却という処理が複雑になってしまう。この値を返すという働きを持つ同期チャンネルを含んだプログラムを非同期チャンネルのみからなるプログラムに変換するプリプロセッサとその変換効率を上げるための変換規則が提案されている。しかしこのコンパイラ、プリプロセッサはともにプログラムで使用できるチャンネルの名前がすべてグローバルであり、名前の重複やローカルなチャンネルが許されていない。よって比較的規模の小さなプログラムにおいてはあまり問題はないが、マルチスレッドプログラミングにおいては規模の大きなプログラムも当然扱うため、その際にはチャンネル名の重複により意図しない働きをする可能性があるため、チャンネル名の重複の有無に常に注意しなければならない。</p> <p>本研究では、Join 算法で用いられているチャンネルのスコープルールを JoinJAVA 言語にも適用し、その実装である JoinJAVA コンパイラを改良することで JoinJAVA 言語でローカルなチャンネルを扱えるように拡張した。Join 算法では、チャンネルの定義以前にチャンネルを使用することができる。Join 算法のスコープルールを JoinJAVA に適用する上で、このことが原因でチャンネルに適応すべき定義が一意に決定できなくなった。これを解決するため入力を 2 度パスする方法を用いて構文解析を行った。また、この拡張によって同期チャンネルを含むプログラムにおいてもプリプロセッサで非同期チャンネルのみのプログラムに変換した後、この拡張したコンパイラでコンパイルすることにより、ローカルなチャンネルを扱うことができるようになった。また、この拡張したコンパイラと拡張前のコンパイラとでコンパイルに要した時間の比較評価を行い、拡張されたコンパイラにおいてもほぼ変わらない時間でコンパイルできることを確認した。</p>		