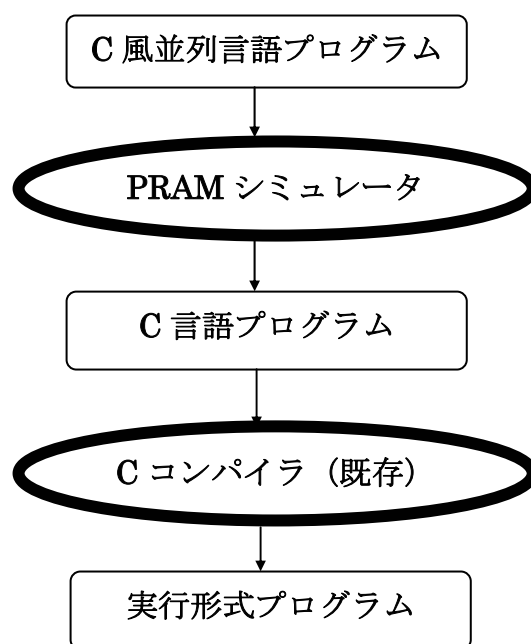


1. **目的** 本研究では PRAM アルゴリズムの評価を容易にする為に PRAM シミュレータを作成する事を目的とする。

2. **原理・方法** PRAM 上で動く C 風並列言語を作成した。また、それを PRAM 上で実行させた場合の動作をシミュレートするシミュレータを作成した。第一図に本研究で作成したシミュレータを用いた C 風並列言語の実行方法を示す。本研究で作成した PRAM シミュレータは C 風並列言語を C 言語に変換する。本研究で定義した C 風並列言語とは C 言語に新たに `parallel` という並列計算用の命令を追加したもので、PRAM シミュレータの限定する範囲で C 言語と同じ記述ができ、PRAM アルゴリズムを扱う事ができるものである。PRAM シミュレータにより C 風並列言語の `parallel` 文は変数のコピーをとった後に `for` 文を実行するという C 言語に変換される。この時、変換されたプログラムは PRAM 上の実行時間も出力するようになっている。これを用いて、いくつかの問題に対する PRAM アルゴリズムをシミュレートし、それが正しい解を出力する事、またその実行時間が理論的な実行時間に等しい事を検証する。



第一図 実行の流れ

3. **結果** データ数 n に対する足し算、及びポインタジャンプを解く C 風並列言語プログラムを作り、PRAM シミュレータを用いて変換する事によってシミュレータが正確に動作しているのかを検討した。検討の結果、変換後の C 言語プログラムは正しい解を出力していると判明した。また、その実行結果は理論値と比較した所、ほぼ等しい値だった。

4. **検討・考察** PRAM シミュレータとしての最低限の能力は備えたシミュレータは作成できた。しかし、本研究で作成した C 風並列言語は C 言語と比べて関数が扱えない事を含め、文法の制限が多く、適応範囲が狭い言語となっている。また、一部のプログラムは正しく変換する事ができなかった。

5. **結論** 本研究で作成したシミュレータは C 風並列言語を C 言語に変換して、さらに C 風並列言語の PRAM 上の実行時間を計測することができる。今後の課題は C 風並列言語をより拡張し、その拡張した C 風並列言語に対応できるシミュレータを作成することである。