

卒業研究報告書

題目

PRAM シミュレータの構築

指導教員

石水隆助手

報告者

01-1-26-071

柳原大志

近畿大学理工学部電気工学科

平成 17 年 2 月 19 日提出

目次

| | | |
|----------|---------------------------------|--------|
| 第 1 章 | 序論 ¹⁾ | - 1 - |
| 1.1. | 並列処理の概要..... | - 1 - |
| 1.2. | 並列処理の目的 ²⁾ | - 1 - |
| 1.3. | 並列コンピュータの必要性..... | - 2 - |
| 1.4. | 本研究の目的..... | - 2 - |
| 第 2 章 | 原理..... | - 3 - |
| 2.1. | PRAM ³⁾ | - 3 - |
| 2.2. | PRAM アルゴリズム ⁴⁾ | - 4 - |
| 2.3. | PRAM アルゴリズムの正当性と計算量..... | - 5 - |
| 2.4. | PRAM シミュレータ..... | - 5 - |
| 第 3 章 | 実験方法..... | - 7 - |
| 3.1. | C 風並列言語プログラム..... | - 7 - |
| 3.2. | PRAM シミュレータ..... | - 8 - |
| 第 4 章 | 結果..... | - 10 - |
| 4.1. | PRAM シミュレータの性能..... | - 10 - |
| 第 5 章 | 考察..... | - 12 - |
| 5.1. | PRAM シミュレータの性能..... | - 12 - |
| 5.2. | C 風並列言語の性能..... | - 12 - |
| 第 6 章 | 結論..... | - 13 - |
| | 参考文献..... | - 15 - |
| 付録 1 | | - 16 - |
| 付録 2 | | - 46 - |
| 付録 2.1 | 足し算プログラム..... | - 46 - |
| 付録 2.1.1 | 変換前 C 風言語プログラム..... | - 46 - |
| 付録 2.1.2 | 変換後 C 言語プログラム修正前..... | - 46 - |
| 付録 2.1.3 | 変換後 C 言語プログラム修正後..... | - 48 - |

第1章 序論¹⁾

1.1. 並列処理の概要

より速いコンピュータの必要性はコンピュータが考案された当初からやむことはない。最も簡単なものから最も高性能なものまで、今日存在する大多数のコンピュータは概念的には互いに非常によく似ている。それらのアーキテクチャと演算は多かれ少なかれ J.von Neumann が考案し、1940 年代に定式化された設計原理に従っている。その操作の流れは単純であり、基本的には次のようである。制御装置は命令とオペランドを記憶装置からとってきて、それらを処理装置に送る。その結果がメモリに戻される。この動作系列が書く命令ごとに繰り返される。そこでは処理装置、制御装置、記憶装置がそれぞれただ一つあり、一度にただ一つだけの命令を実行する。

並列計算を用いると状況は一変する。並列コンピュータは複数の処理装置、またはプロセッサからなり、ある問題を解く際、その問題の異なる部分問題を同時に解き、全体の結果を導くのに協力し合う。使用されるプロセッサ数は数個から数百万にまでわたる。結果的にこれまでの単一プロセッサのコンピュータで問題を解くのにかかる時間よりも非常に少ない時間で解くことができる。このアプローチは次のような理由からも魅力である。

- (1) 多くの問題に対して並列的な解法のほうがごく自然である。
- (2) 近年コンピュータ素子の価格とサイズが急激に下がっており、多数のプロセッサを用いた並列コンピュータが作りやすくなっている。
- (3) 並列処理では問題または問題クラスを解くのに最も適した並列アーキテクチャを運ぶことが可能である。

並列性は人間の思考方法および、コンピュータの使いかたを確実に変化させる。それは、これまでに解くことのできなかつた問題を手の届く範囲にし、また以前には夢にも見なかつたような知識の新分野の研究も可能とする。豊富なアーキテクチャは新旧の問題に対して、貴重な解法より効果的な解法の発見のたすけとなる。

1.2. 並列処理の目的²⁾

並列処理システムの開発の目的は大きく分けて 2 つある。第 1 の目的はより小さいサイズの部分問題に分割した後、多数のプロセッサ上で各部分問題を同時に処理することにより処理を高速化することである。もう 1 つの目的は耐故障 (Fault Tolerant) 性や無待機 (Wait Free) 性を得ることである。システムを多重化することによりいくつかのプロセッサが故障してもシステム全体としては処理を続けることができる。本研究では第 1 の目的である処理の高速化を取り上げる。

1.3. 並列コンピュータの必要性

より速いコンピュータがさまざまな分野で求められている。具体的な例としては次のようなものがある。

- (1) 量子力学、統計力学、相対論的物理学
- (2) 宇宙論と宇宙物理学
- (3) 計算流体力学と乱流
- (4) 物質の設計と超流動
- (5) 生物学、薬理学、ゲノム配列、遺伝子工学、たんぱく質の合成、酵素の働き、細胞のモデル化
- (6) 医学、人間の臓器と骨格のモデル化
- (7) 大域的な気象と環境のモデル化

高速なコンピュータが必要となるその他の応用としては、物質探査、経済の計画、暗号解析、地震学、航空機のテスト、原始・核・プラズマ物理学などの数値シミュレーション、ロボティックス、大規模データベースの管理、実時間音声認識、人間とコンピュータの高度なインターフェースなどがある。

計算速度を向上させるための方法としては、性能の良い高速な素子を用いるか、あるいは並列処理を行なうかの2通りが考えられる。近年、計算素子は高度に集積され、計算速度の目覚ましい向上が成し遂げられてきた。しかし真空中の光の速度が 3×10^8 m/sec であるという物理学の法則が、この傾向が間もなく終わるということを予想している。一方、並列処理にはそのような限界は本質的には存在しない。このため並列処理に対して大きな期待がよせられている。

1.4. 本研究の目的

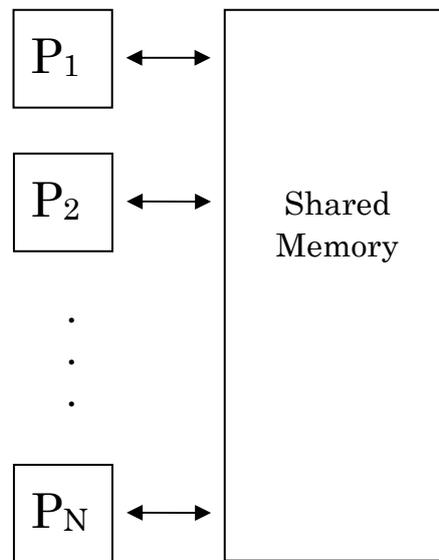
本研究では PRAM アルゴリズムの評価を容易にする為に PRAM シミュレータを作成する事を目的とする。

本研究では、PRAM アルゴリズムを記述できる C 風並列言語を作成する。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する。本研究で作成した PRAM シミュレータは C 風並列言語プログラムを C 言語プログラムに変換する。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。

第2章 原理

2.1. PRAM³⁾

第1図のように、複数のプロセッサがメモリを共有したコンピュータを共有メモリ並列コンピュータ(Shared Memory Parallel Computer)と呼ぶ。代表的な並列計算モデルであるPRAM(Parallel Random-Access Machine)は共有メモリ型並列計算機を抽象化したモデルである。



第1図 共有メモリコンピュータ

Fig.1 Shared Memory Parallel Computer

PRAMは共有メモリとそれに接続されたP個のプロセッサからなる。並列アルゴリズムの実行中、P個のプロセッサは入力データの読み出し、中間結果の読みだし、及び書き込み、最終結果の書き込みをするために、共有メモリにアクセスする。PRAM各プロセッサは、共有メモリ上の任意の位置にあるメモリセルに対して1単位時間でデータの読み書きができ、また全ての演算は1単位時間で行なうことができると仮定されている。また、1単位時間ごとに全てのプロセッサで同期がとられる完全同期型モデルである。

複数のプロセッサによる異なる位置のメモリセルへのアクセスに対しては全てのプロセッサが自由に読み書きを行なうことができる。一方、複数のプロセッサによる同一セルへのアクセスについてはそれをどう処理するかによりPRAMが以下の4つに分類される。

- (1) 排他読み出し排他書き込み(EREW, Exclusive-Read Exclusive-Write)PRAM
メモリ位置へアクセスは排他的である。どの2つのプロセッサも同じメモリ位置から同時に読み出したり書き込んだりできない。

- (2) 同時読み出し排他書き込み(CREW, Concurrent-Read Exclusive-Write)PRAM
複数のプロセッサが同時に同じメモリ位置から読み出すことができるが、書き込みの権利は排他的であり、どの 2 つのプロセッサも同じメモリ位置に同時に書き込むことはできない。
- (3) 排他読み出し同時書き込み(ERCW, Exclusive-Read Concurrent-Write)PRAM
複数のプロセッサが同時に同じメモリ位置に書き込むことができるが、読み出しは排他的である。
- (4) 同時読み込み同時書き込み(CRCW, Concurrent-Read Concurrent-Write)PRAM
複数のプロセッサによる同じメモリ位置への同時読み出し、同時書き込みが認められている。

さらに、CRCWPRAM は同時書き込みが行われる際の処理方法で以下の 3 種に分類される。

- (i) 優先型(Priority)CRCWPRAM
同時書き込みが起こった時、最も優先順位が高いものが書き込みに成功する。
- (ii) 任意型(Arbitrary) CRCWPRAM
同時書き込みが起こった時、どれか一つが書き込みに成功する。
- (iii) 共通型(Common) CRCWPRAM
同時書き込みが起こった時、全てが同じ値書き込もうとした時に成功し、その他はエラーとする。

尚、本研究で作成した PRAM シミュレータは優先型 CRCW である。

2.2. PRAM アルゴリズム⁴⁾

アルゴリズム(Algorithm)は、問題を解くための手段を定めたものである。この手順はどのような操作をそのような手順で行うかを曖昧な点の残らないようきちんと定めたものでなければならない。

手順を明確に定めてあれば計算機をその手順どおり動かして問題を解かせることができる。アルゴリズムという概念自信は計算機と無関係に成立するが、普通は計算機に問題を解かせるための手順を示す。

PRAM アルゴリズムは PRAM 上でどのようなデータを分割し、それをどのプロセッサに割り当てるか記述しなければならない。そのため、一般的には逐次アルゴリズムをそのまま PRAM アルゴリズムにすることはできず、PRAM 用に新たにアルゴリズムを作成する必要がある。

2.3. PRAM アルゴリズムの正当性と計算量

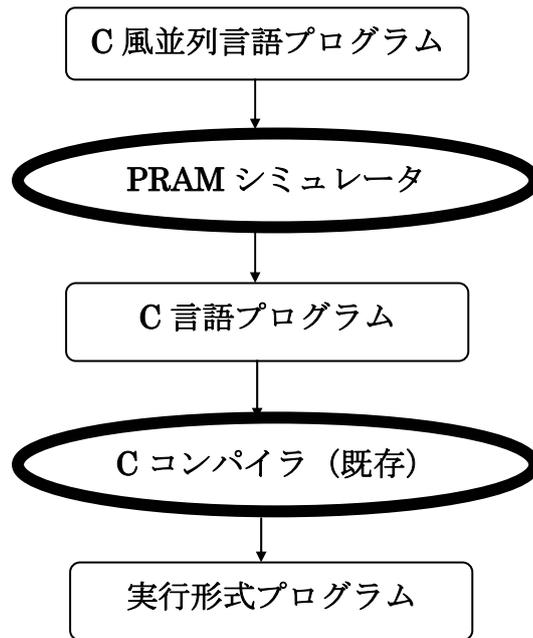
PRAM アルゴリズムは正当性を満たさなければならない。正当性とはアルゴリズムが有限の時間内に正しい解を出力することである。正当性の理論的な検証は考えられる全てのデータに対して正しいことを保証しなければならないため、一般的には難しい。また、PRAM アルゴリズムはその計算量も検証する必要がある。計算量はどれか 1 つのプロセッサが処理を開始してから全てのプロセッサが処理を終了するまでの時間である。計算量の検証も様々な例外的な場合について調べる必要があり、やはり一般的には難しい。

2.4. PRAM シミュレータ

本研究では、PRAM アルゴリズムの正当性とその計算量を実験的に評価するため、PRAM シミュレータを作成した。また、PRAM アルゴリズムを記述するため、C 風並列言語を作成した。

第 2 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 風並列言語プログラムを C 言語プログラムに変換する。また、C 言語プログラムに PRAM 上での実行時間を表示する機能を付加する。

PRAM シミュレータを用いた C 風言語プログラムに実行方法は以下の通りである。まず、対照となる問題を解くアルゴリズムを C 風並列言語で記述する。次に、作成した C 風並列言語プログラムのファイル名を `input.pc` とし、PRAM シミュレータと同じディレクトリに置く。その後、PRAM シミュレータを実行すれば `output.c` というファイルに C 言語プログラムが出力される。



第2図 PRAM シミュレータによる実行の流れ
Fig.2 Data Flow by PRAM Simulator

第3章 実験方法

PRAM シミュレータにより C 風並列言語プログラムは C 言語プログラムに変換される。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。

3.1. C 風並列言語プログラム

本研究で作成した C 風並列言語とは C 言語のサブセットを元に並列処理をするための命令 (`parallel`) を加えたものである。`parallel` は引数で指定したプロセッサの集合に以下に続く命令群を並列に実行させる命令である。本研究で作成した C 風並列言語の文法は以下の通りである。

名前 ::= 英字 { 英字 | 数字 | “_” } | “_p”

整数 ::= 数字 { 数字 }

文字列 ::= ダブルクォート { 任意の文字 } ダブルクォート

文字 ::= シングルクォート 任意の文字 シングルクォート

変数 ::= 名前 { “[” <expression> “]” }

<program> ::= [<sharp>] <main>

<sharp> ::= { # (“define” 名前 整数 | “include” “<” 名前 “.h” “>”) }

<main> ::= “main” “(” “)” <block>

<block> ::= “{” { <vdecl> } { <statement> } “}”

<vdecl> ::= (“int” | “char”) 名前 [“[” 整数 “]”] { “,” 名前 [“[” 整数 “]”] } “;”

<statement> ::= <printf> | <scanf> | <if> | <while> | <for> | <assign> | <parallel>
“{” { <pstatement> } “}” | “;”

<printf> ::= “rintf” “(” 文字列 { “,” <expression> } “)” “;”

<scanf> ::= “scanf” “(” 文字列 { “,” “&” 変数 } “)” “;”

<if> ::= “if” “(” <expression> “)” <statement> [“else” <statement>]

*ただし else 節は他の else 節と関係していない最も近い if 文と関係するものとする。

<while> ::= “while” “(” <expression> “)” <statement>

<for> ::= “for” “(” [<forassign>] “;” [<expression>] “;” “(” [<forassign>] “)”
<statement>

<assign> ::= 変数 (“=” <expression> | “++” | “--”) “;”

<forassign> ::= 変数 (“=” <expression> | “++” | “--”)

<parallel> ::= “(” <expression> “,” <expression> “)” <statement>

*ただし parallel 文の中には、さらに parallel は入れてはいけない。さらに、parallel の中に for を入れた (ループパラメタを使用) 場合は誤作動を起こす場合があるので修正が必要な場合がある。

<leftside> ::= 変数

<expression> ::= <logicalterm> { “|” <logicalterm> }

<logicalterm> ::= <logicalfactor> { “&&” <logicalfactor> }

<logicalfactor> ::= <arithmeticexpression>

| <arithmeticexpression> “==” <arithmeticexpression>

| <arithmeticexpression> “!=” <arithmeticexpression>

| <arithmeticexpression> “<” <arithmeticexpression>

| <arithmeticexpression> “<=” <arithmeticexpression>

| <arithmeticexpression> “>” <arithmeticexpression>

| <arithmeticexpression> “>=” <arithmeticexpression>

<arithmeticexpression> ::= <arithmeticterm> { (“+” | “-”) <arithmeticterm> }

<arithmeticterm> ::= <arithmetictfactor> { (“*” | “/” | “%”) <arithmetictfactor> }

<arithmetictfactor> ::= <unsignedfactor> | “!” <unsignedfactor>

| “-” <unsignedfactor>

<unsignedfactor> ::= 変数 | 整数 | 文字 | “(” <expression> “)”

以上が C 風並列言語の定義である。命令 parallel は以下のように使用する。

parallel(式 1,式 2)文

上記のように書いた場合、プロセッサ番号式 1 から式 2 までのプロセッサが後ろに続く文を並列に実行する。また、この時 int 型変数_p には現在処理中のプロセッサ番号が格納される。

3.2. PRAM シミュレータ

本研究で開発した PRAM シミュレータは C 風並列言語を C 言語に変換するコンパイラである。具体的には parallel という命令があった場合、全ての変数を作業用変数にコピーする。さらに parallel(式 1,式 2) 文 1 とあれば

for(_p=式 1;_p<=式 2;_p++) 文 2 に変換する。

ただし、文 2 とは文 1 中の代入文で出てくる右辺にでてくる変数を全てコピー用変数に置き換えたものである。また_p とは実行中のプロセッサ番号を表す。

例として以下のようなプログラムが与えられたとする。

```
int a[10];  
parallel(0,8) a[_p]=a[_p+1];
```

この時、PRAM シミュレータは以下のように出力する。

```
int a[10],__a[10],_i;  
for (_i=0;_i<10;_i+) __a[_i]=a[_i];  
for(_p=0;_p<=8;_p++) a[_p]=__a[_p+1];
```

また変換される際に時間を計測する_t を加える。この_t は C 風並列言語の各命令が実行されるたびに増加し、C 風並列言語の出力が終了すると_t の値が出力されることになっている。

第4章 結果

4.1. PRAM シミュレータの性能

本研究で作成した PRAM シミュレータでは C 風言語の中で `parallel` の中に `for` を入れた場合、正しく動作しない。

例として付録 2. 1. 1 に C 風並列言語プログラムを、付録 2. 1. 2 にそれを PRAM シミュレータにより変換した C 言語プログラムを示す。プログラム 2. 1. 1 は n 個のデータを与えられたとき n 台のプロセッサを用いてその和を求めるプログラムである。以下に示すように `for` 文中のループパラメータを使用している部分を修正する必要がある。

38 行目の `a[i]=_a[2*_i]+_a[2*_i+1];` を `a[i]=_a[2*i]+_a[2*i+1];` に修正
45 行目 `a[i]=_a[2*_i];` を `a[i]=_a[2*i];` に修正。

修正後のプログラムは付録 2. 1. 3 に示す。

この修正後のプログラムを実行した所、正しい解が得られた。またデータ数 n を変えた時の実行時間を計った。この結果を第 1 表に示す。

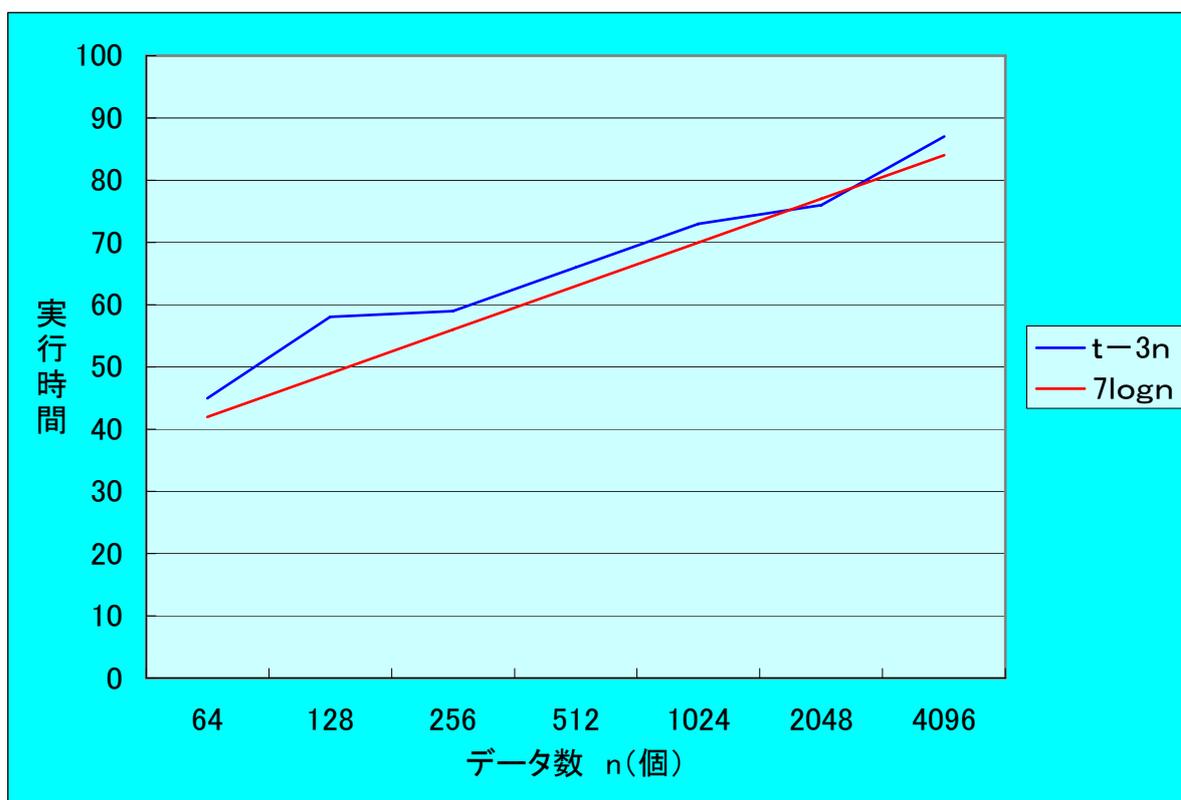
第 1 表 足し算プログラムの実行時間

Table1. Running Time of Sum Program by Pram Simulator

| データ数 n | 実行時間 t | $t - 3n$ | $7 \log n$ |
|----------|----------|----------|------------|
| 64 | 237 | 45 | 42 |
| 128 | 436 | 58 | 49 |
| 256 | 827 | 59 | 56 |
| 512 | 1602 | 66 | 63 |
| 1024 | 3145 | 73 | 70 |
| 2048 | 6224 | 76 | 77 |
| 4096 | 12375 | 87 | 84 |

第 1 表の n は計算の対象の個数を表し、 t は命令実行数を表す。 n 個のデータに対し初期値の入力は時間 $3n$ で行なわれるため実質的な実行時間は $t - 3n$ である。

足し算プログラムの並列アルゴリズム上での実行時間は $\log n$ に比例する。ここで本研究で得られた実行結果（第 1 表参照）からデータ数と実行時間の関係を示すため、データ数を x 軸に、初期値入力時の実行時間を考慮した実行時間 $t - 3n$ と $7 \log n$ を y 軸にして、その関係を示したグラフを第 2 図に示す。



第3図 PRAM シミュレータによる足し算プログラムの実行時間
 Fig.3 Running Time of Sum Program by Pram Simulator

第3図より $t-3n \approx 7 \log n$ である。従って、実行時間はデータ数の対数に比例し、理論値と一致する。

第5章 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。

5.1. PRAM シミュレータの性能

本研究で作成した PRAM シミュレータは一部の C 風並列言語プログラムは正しく変換することができなかった。考えられる原因としては以下のようなことが考えられる。

`parallel` の中に `for` を入れた文がある C 風並列言語プログラムを変換した場合、まず変数を作業用変数にコピーした後、`parallel` 文は `for` 文に変換され、また、`for` 文中の代入文の右辺に使われる全ての変数はこのコピーした変数を使用するような C 言語プログラムに変換される。この時 `for` 文のパラメータとして使用されている変数はループするごとに増加するようになっているのに対し、コピーした変数は最初に初期値が設定されてから値は増加していかない。よって `for` 文中内でループパラメータを使用した代入文がある場合ループパラメータの値とそのコピーとの間で値の不一致が起こり、正しく動作しなくなる。また、本研究で作成したシミュレータは入力ファイルとしては同じディレクトリにあるファイル名 `input.pc` のみ、出力ファイルとしてはファイル名 `output.c` のみ認めるため、使用者が好むファイル名を自ら指定したり、入力や出力の際に他のディレクトリから参照したりすることはできない。

5.2. C 風並列言語の性能

本研究で作成した C 風並列言語とは C 言語のサブセットを元に並列処理をするための命令 (`parallel`) を加えたものであり、基本的な命令はほぼ C 言語と同一のものである。しかし、本研究で作成した C 風並列言語は適応範囲が C 言語と比べて低いため、関数に対応していないなどの使い勝手の悪さが残る。この問題を解決するためには本研究で作成した C 風並列言語の適応範囲を広めるために、仕様をさらに改善していく必要がある。

第6章 結論

本研究では、PRAM アルゴリズムを記述できる C 風並列言語の提案、及び C 風並列言語プログラムの PRAM 上での動作をシミュレートできるシミュレータを作成した。

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、PRAM シミュレータとしての最低限の能力は備えている。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、本研究で作成した PRAM シミュレータは `parallel` の中に `for` を入れた場合など、一部の C 風並列言語プログラムは正しく変換することができなかった。また、本研究で作成したシミュレータは入力ファイルとしては同じディレクトリにあるファイル名 `input.pc` のみ、出力ファイルとしてはファイル名 `output.c` のみ認めるため、使用者が好むファイル名を自ら指定したり、入力や出力の際に他のディレクトリから参照したりすることはできない。

今後の課題としては C 風並列言語の適応範囲をさらに広げるために、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が挙げられる。

さらに、入力ファイル名、出力ファイル名を指定した名前にすることができるようにする事や、入力や出力の際に別のディレクトリから参照できるようにすることも挙げられる。

謝辞

本研究を進めるにあたり、多種多様にわたる数え切れない様々なご指導や御助言を頂いた石水隆助手には大変尽力して頂き、感謝申し上げます。

さらには、情報論理工学研究室の方々には、相談等乗っていただいたことに、大変感謝申し上げます。

参考文献

- 1) 渋沢 進：“並列分散処理入門”、培風館、東京、1~5、(1998)
- 2) 渡辺勝正：“並列処理概説”、コロナ社、東京。195、(1991)
- 3) 渋沢 進：“並列分散処理入門”、培風館、東京、39、46、48~50、(1998)
- 4) 渋沢 進：“並列分散処理入門”、培風館、東京、13、14、(1998)
- 5) 加藤 暢、樋口昌宏：“情報・コンピュータシステムプロジェクト I 指導書”、近畿大学工学部情報学科、(2004)
- 6) 疋田輝雄・石畑 清：“コンパイラの理論と実現”、共立出版、東京、(1992)
- 7) 辻野嘉宏：“コンパイラ”、昭晃堂、東京、(2000)

付録 1

本研究で用いた PRAM シミュレータプログラムを以下に示す。

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define HSIZEMAX 100
#define HNMAX 255
#define TEQEQ 1
#define TNULL 0
#define TEQ 2
#define TMOREEQ 3
#define TMORE 4
#define TLESSEQ 5 /* <= */
#define TLESS 6
#define TINC 7
#define TADD 8
#define TDEC 9
#define TSUB 10/* - */
#define TMUL 11
#define TDIV 12
#define TMOD 13
#define TAND 14
#define TLAND 15
#define TOR 16
#define TLOR 17
#define TNEQ 18
#define TNOT 19
#define TKO 50
#define TKC 51
#define TMKO 52
#define TMKC 53
#define TKKO 54
#define TKKC 55
#define TSC 56
#define TSHP 57
#define TCMA 58
#define TDOT 59
#define TDOTH 60
#define TINTEGER 20/* 数字 */
#define THNAME 21 /* 変数名 */
#define TSTRING 22 /* 文字列 */
```

```

#define TCHARACTER 23/* 文字 */
#define TMAIN 24
#define TVOID 25
#define TINT 26
#define TCHAR 27
#define TINCLUDE 28
#define TDEFINE 29
#define TIF 30
#define TELSE 31
#define TWHILE 32
#define TFOR 33
#define TPRINTF 34
#define TSCANF 35
#define TPARALLEL 36

char c; /* 今読み込んでる文字 */
char nc; /* 今読み込んでる文字の次の文字 */
char H[50]; /* 字句解析で記憶する文字列 */
char HTABLE[HNMAX][HSIZEMAX];
int V; /* 字句解析で記憶する値 */
int HN=0;
int HSIZE[HNMAX];
FILE *FPI;
FILE *FPO;
int getkey();
int key;
int nkey;
int HTYPE[255];
int indent;
void newline();
void Serror();
int Nline=0;
int Nchar=0;
int pflag=0;
int aflag=0;

void main() {
    void pprogram();
    FPI=fopen("input.pc","r");
    FPO=fopen("output.c","w");
    c='¥0'; /* ダミーの¥をcに読み込む */
    nc=getc(FPI); /* ファイルの中の最初の文字をncに読み込む */
    pprogram();
}

```

```
/* 構文解析 */
```

```
void newline() {  
    int i;  
    fprintf(FP0, "\n");  
    if(nkey==TMKC) {  
        for(i=0; i<indent-1; i++) {  
            fprintf(FP0, "  ");  
        }  
    }  
    else {  
        for(i=0; i<indent; i++) {  
            fprintf(FP0, "  ");  
        }  
    }  
}
```

```
void program() {          /* program */  
    int psharp();  
    void pmain();  
    int l;  
    key=getkey();  
    nkey=getkey();  
    l=psarp(0);  
    pmain(l);  
}
```

```
int psharp(l) int l; {  
    fprintf(FP0, "#define _P 100");  
    newline();  
    while(key==TSHP) {  
        fprintf(FP0, "#");  
        key=nkey;  
        nkey=getkey();  
        if(key==TDEFINE) {  
            fprintf(FP0, "define ");  
            key=nkey;  
            nkey=getkey();  
            if(key==THNAME) {  
                strcpy(HTABLE[l], H); /* HTYPE=9は#define */  
                HTYPE[l]=9;  
                l++;  
                HN++;  
                fprintf(FP0, "%s ", H);  
                key=nkey;
```

```

        nkey=getkey();
    }
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
if(key==TINTEGER) {
    fprintf(FP0, "%d¥n", V);
    key=nkey;
    nkey=getkey();
}
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
}
if(key==TINCLUDE) {
fprintf(FP0, "include");
key=nkey;
    nkey=getkey();
    if(key==TLESS) {
fprintf(FP0, "<");
key=nkey;
    nkey=getkey();
    }
    else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
    if(key==THNAME) {
        fprintf(FP0, "%s", H);
    key=nkey;
    nkey=getkey();
    }
    else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
if(key==TDOTH) {
    fprintf(FP0, ".h");
    key=nkey;
    nkey=getkey();
}
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
if(key==TMORE) {
    fprintf(FP0, ">¥n");
    key=nkey;
    nkey=getkey();
}
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
}
}
fprintf(FP0, "int _t=0;");
newline();
fprintf(FP0, "int _tp[_P]:");
newline();
fprintf(FP0, "int _p;");

```

```

        newline();
    fprintf(FP0, "int _t1:");
        newline();
    fprintf(FP0, "int _t2:");
        newline();
    fprintf(FP0, "int _pmax():");
        newline();
        fprintf(FP0, "int _j:");
        newline();
    return l;
}

void pmain(l)int l;{
    /* main */
    void block();
    if(key==TVOID||key==TINT){
        if(key==TVOID) fprintf(FP0, "void ");
        else if(key==TINT) fprintf(FP0, "int ");
        key=nkey;
        nkey=getkey();
    }
    if(key==TMAIN){
        fprintf(FP0, "main");
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! ! エラーを出力 ! ! ! ! ! */
    if(key==TK0){
        fprintf(FP0, "(");
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! ! エラーを出力 ! ! ! ! ! */
    if(key==TKC){
        fprintf(FP0, ")");
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! ! エラーを出力 ! ! ! ! ! */
    if(key==TMKO){
        indent++;
        fprintf(FP0, "{");
        newline();
        key=nkey;
        nkey=getkey();
    }
}

```

```

else Serror() /* ! ! ! ! エラーを出力 ! ! ! ! */

block(l);

if(key==TMKC) {
    indent--;

    fprintf(FP0, "printf(¥"Time is %c%c¥", _t);", 37, 100);
    newline();

    fprintf(FP0, "}");
    newline();
    key=nkey;
    nkey=getkey();
}
else Serror() /* ! ! ! ! エラーを出力 ! ! ! ! */
    if(key!=TNULL) Serror() /* ! ! ! ! エラーを出力 ! ! ! ! */

    fprintf(FP0, "int _pmax(_t1, _t2) int _t1; int _t2; {");
    indent++;
    newline();
    fprintf(FP0, "int max=_t1, i;");
    newline();
    fprintf(FP0, "for (i=_t1+1; i<=_t2; i++) if(_tp[max]<_tp[i]) max=i;");
    newline();
    fprintf(FP0, "return(_tp[max]);");
    indent--;
    newline();
    fprintf(FP0, "}");
    newline();
}

void block(l) int l; {
    int pvdecl();
    void pstatement();
    while(key==TINT || key==TVOID || key==TCHAR) {
        l=pvdecl(l);
    }
    while(key!=TMKC) {
        pstatement();
    }
}

int pvdecl(l) int l; {

```

```

int f;
if(key==TINT) {
fprintf(FP0, "int ");
    key=nkey;
    nkey=getkey();
    do{
        if(key!=THNAME) Serror();/* エラ----- */
        else {
fprintf(FP0, "%s", H);
            strcpy(HTABLE[1], H);
                HN++;
        }
    }
key=nkey;
    nkey=getkey();
    if(key==TKKO) {
fprintf(FP0, "[");
key=nkey;
    nkey=getkey();
        if(key==TINTEGER) {
fprintf(FP0, "%d", V);
                HSIZE[1]=V;
key=nkey;
    nkey=getkey();
        if(key==TKKC) {
fprintf(FP0, "]");
                key=nkey;
nkey=getkey();

                if(key==TKO) {
fprintf(FP0, "(");
                    key=nkey;
nkey=getkey();
                        if(key==TKC) {
fprintf(FP0, ")");
                                HTYPE[1]=3; /* HTYPE=3はint配列型関数 */
                                    key=nkey;
nkey=getkey();
                                        }
                                            else Serror();/* !!!!!エラーを
出力!!!! */
                                                }
                                                    else {
HTYPE[1]=1; /* HTYPE=1はint型配列
*/

```

```

fprintf(FPO, " __%s[%d]", HTABLE[I], V);
    }
    }
    else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */

}
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */

}
else if(key==TK0) {
    fprintf(FPO, "(");
        key=nkey;
        nkey=getkey();
        if(key==TKC) {
            fprintf(FPO, ")");
                HTYPE[I]=2; /* HTYPE=2はint型関数 */
                key=nkey;
                nkey=getkey();
            }
            else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */

        }
        else {
            HTYPE[I]=0; /* HTYPE=0はint型 */
            fprintf(FPO, "=0, __%s", HTABLE[I]);
        }
        I++;
        if(key==TCMA) {
            fprintf(FPO, ",");
                key=nkey;
                nkey=getkey();
                f=1;
            }
            else f=0;

        }
        while(f);
        if(key==TSC) {
            fprintf(FPO, ";");
                newline();
                key=nkey;
            nkey=getkey();

```

```

    }
    else Serror ();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */

}

else if(key==TCHAR) {
fprintf(FPO, "char ");
    key=nkey;
    nkey=getkey();
    do{
        if(key!=THNAME) Serror ();/* エラー----- */
        else {
fprintf(FPO, "%s", H);
            strcpy(HTABLE[1], H);
                HN++;
        }
    }
    key=nkey;
    nkey=getkey();
    if(key==TKK0) {
fprintf(FPO, "(");
    key=nkey;
    nkey=getkey();
    if(key==TINTEGER) {
fprintf(FPO, "%d", V);
    key=nkey;
    nkey=getkey();
        if(key==TKKC) {
fprintf(FPO, "]"");
            key=nkey;
    nkey=getkey();

                if(key==TK0) {
fprintf(FPO, "(");
                    key=nkey;
    nkey=getkey();
                if(key==TKC) {
fprintf(FPO, "]"");
                    HTYPE[1]=7; /* HTYPE=7はchar配列型関数
*/
                    key=nkey;
                    nkey=getkey();
                }
                else Serror ();/* ! ! ! ! !エラーを
出力 ! ! ! ! ! */
            }
        }
    }
}

```

```

else {
    HTYPE[1]=5; /* HTYPE=5はchar型配列
*/
    fprintf(FPO, " __%s[%d]", HTABLE[1], V);
    HSIZE[1]=V;
}
}
else Serror();/* !!!!!エラーを出力!!!! */

}
else Serror();/* !!!!!エラーを出力!!!! */

}
else if(key==TK0) {
    fprintf(FPO, "(");
    key=nkey;
    nkey=getkey();
    if(key==TKC) {
        fprintf(FPO, ")");
        HTYPE[1]=6; /* HTYPE=6はchar型関数 */
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* !!!!!エラーを出力!!!! */

}
else {
    HTYPE[1]=4; /* HTYPE=4はchar型 */
    fprintf(FPO, "=0, __%s", HTABLE[1]);
}
l++;
if(key==TCMA) {
fprintf(FPO, ",");
    key=nkey;
    nkey=getkey();
    f=1;
}
else f=0;
}
while(f);
if(key==TSC) {

```

```

    fprintf(FPO, "");
        newline();
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */

}

else if(key==TVOID) {
    fprintf(FPO, "void ");
        key=nkey;
        nkey=getkey();
        do{
            if(key!=THNAME) Serror();/* エラー----- */
            else {
                fprintf(FPO, "%s", H);
                    strcpy(HTABLE[1], H);
                    HN++;
            }
        }
        key=nkey;
        nkey=getkey();
            if(key==TK0) {
                fprintf(FPO, "(");
                    key=nkey;
                    nkey=getkey();
                        if(key==TKC) {
                            fprintf(FPO, "");
                                key=nkey;
                                nkey=getkey();
                                    HTYPE[1]=8; /* HTYPE=8はvoid型関数 */
                                }
                            else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
                        }
                    else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
                l++;
            }
        if(key==TCMA) {
            fprintf(FPO, ",");
                key=nkey;
                nkey=getkey();
                f=1;
            }
            else f=0;
        }
    while(f);

```

```

        if(key==TSC) {
            fprintf(FPO, "");
                newline();
                key=nkey;
            nkey=getkey();
        }
        else Serror();/* !!!!!エラーを出力!!!! */
    }
return l;
}

```

```

void pstatement() {          /* statement */
    void pprintf();
    void pscanf();
    void pif();
    void pwhile();
    void pfor();
    void passign();
    void pmko();
    void psc();
    void parallel();

    switch(key) {
case TPRINTF:
        pprintf();
        break;
case TSCANF:
        pscanf();
        break;
    case TIF:
        pif();
        break;
    case TWHILE:
        pwhile();
        break;
case TFOR:
        pfor();
        break;
case THNAME:
        passign();
        break;
case TMKO:
        indent++;
        fprintf(FPO, "{");

```

```

        newline();
        key=nkey;
        nkey=getkey();
        while(key!=TMKC) {
            pstatement();
        }
        indent--;
        fprintf(FP0, "}") ;
        newline();
        key=nkey;
        nkey=getkey();
        break;
case TSC:
    fprintf(FP0, ":",");
        newline();
        key=nkey;
        nkey=getkey();
        break;
case TPARALLEL:
    parallel();
        break;
    default:
        Serror();/* ! ! ! ! ! エラーを出力 ! ! ! ! ! */
    }
}

void pprintf() {
    void pexpression();
    fprintf(FP0, "printf");
    key=nkey;
    nkey=getkey();
    if(key==TK0) {
        fprintf(FP0, "(");
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! ! エラーを出力 ! ! ! ! ! */
    if(key==TSTRING) {
        fprintf(FP0, "¥"¥s¥""", H);
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! ! エラーを出力 ! ! ! ! ! */
    while(key==TCMA) {
        fprintf(FP0, ",",");

```

```

        key=nkey;
        nkey=getkey();
            pexpression();
        }
    if(key==TKC) {
        fprintf(FP0, "(");
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! エラーを出力 ! ! ! ! */
    if(key==TSC) {
        fprintf(FP0, ";"");
            newline();

        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! エラーを出力 ! ! ! ! */
}

```

```

void pscanf() {
    void pexpression();
    fprintf(FP0, "scanf");
    key=nkey;
    nkey=getkey();
    if(key==TK0) {
        fprintf(FP0, "(");
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! エラーを出力 ! ! ! ! */
    if(key==TSTRING) {
        fprintf(FP0, "%s", H);
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! エラーを出力 ! ! ! ! */
    while(key==TCMA) {
        fprintf(FP0, ",");
        key=nkey;
        nkey=getkey();
            if(key==TAND) {
                fprintf(FP0, "&");
                key=nkey;
                nkey=getkey();
            }
    }
}

```

```

        else Serror ();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
        if(key==THNAME) {
            fprintf(FP0, "%s", H);
            key=nkey;
            nkey=getkey();
        }
        else Serror ();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
    }
if(key==TKC) {
    fprintf(FP0, ")");
    key=nkey;
    nkey=getkey();
}
else Serror ();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
if(key==TSC) {
    fprintf(FP0, ";");
    newline();
    key=nkey;
    nkey=getkey();
}
else Serror ();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
}

void pif() {                /* pif */
    void pexpression();
    void pstatement();
    if(key==TIF) {
        fprintf(FP0, "if");
        key=nkey;
        nkey=getkey();
    }
    else Serror ();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
    if(key==TK0) {
        fprintf(FP0, "(");
        key=nkey;
        nkey=getkey();
    }
    else Serror ();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
pexpression();
if(key==TKC) {
    fprintf(FP0, ")");
    key=nkey;
    nkey=getkey();
}
else Serror ();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */

```

```

        fprintf(FPO, "{");
        newline();
    if(pflag==0) fprintf(FPO, "_t++:");
        else fprintf(FPO, "_tp[_p]++:");
    newline();
    pstatement();
    fprintf(FPO, "}");
        newline();
        if(key==TELSE) {
    fprintf(FPO, "else ");
    key=nkey;
        nkey=getkey();
        if(key==TMKO) {
            pstatement();
        }
    else{
            indent++;
            fprintf(FPO, "{");
            newline();
            if(pflag==0) fprintf(FPO, "_t++:");
            else fprintf(FPO, "_tp[_p]++:");
            newline();
            pstatement();
            indent--;
            fprintf(FPO, "}");
            newline();
        }
    }
}

void pwhile() {          /* pwhile */
    void pexpression();
    void pstatement();
    if(key==TWHILE) {
        fprintf(FPO, "while");
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! エラーを出力 ! ! ! ! */
    if(key==TKO) {
        fprintf(FPO, "(");
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* ! ! ! ! エラーを出力 ! ! ! ! */
}

```

```

pexpression();
if(key==TKC) {
    fprintf(FPO, "(");
    key=nkey;
    nkey=getkey();
}
else Serror();/* !!!!!エラーを出力!!!! */
fprintf(FPO, "{");
newline();
if(pflag==0) fprintf(FPO, "_t++");
else fprintf(FPO, "_tp[_p]++");
newline();
pstatement();
fprintf(FPO, "}");
newline();
}

```

```

void pforassign() { /* for専用assign */
    void pleftside();
    void pexpression();
    pleftside();
    if(key==TEQ) {
        fprintf(FPO, "=");
        key=nkey;
        nkey=getkey();
        pexpression();
    }
    else if(key==TINC) {
        fprintf(FPO, "++");
        key=nkey;
        nkey=getkey();
    }
    else if(key==TDEC) {
        fprintf(FPO, "--");
        key=nkey;
        nkey=getkey();
    }
    else Serror();/* !!!!!エラーを出力!!!! */
}

```

```

void pfor() { /* pfor */
    void pexpression();
    void pstatement();
    if(key==TFOR) {
        if(pflag==0) fprintf(FPO, "_t++");
    }
}

```

```

else fprintf(FP0, "_tp[_p]++;");
        newline();
fprintf(FP0, "for");
        key=nkey;
        nkey=getkey();
    }
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
    if(key==TK0) {
fprintf(FP0, "(");
        key=nkey;
        nkey=getkey();
    }
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
pforassign();
    if(key==TSC) {
fprintf(FP0, ";");
        key=nkey;
        nkey=getkey();
    }
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
pexpression();
    if(key==TSC) {
fprintf(FP0, ";");
        key=nkey;
        nkey=getkey();
    }
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
pforassign();
if(key==TKC) {
    fprintf(FP0, ") ");
        key=nkey;
        nkey=getkey();
    }
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
    indent++;
fprintf(FP0, "{");
    newline();
if(pflag==0) fprintf(FP0, "_t=_t+2;");
    else fprintf(FP0, "_tp[_p]=_tp[_p]+2;");
newline();
pstatement();
    indent--;
fprintf(FP0, "}");
    newline();
}

```

```

void passign() {                                /* assign */
    void pleftside();
    void pexpression();
    pleftside();
    if(key==TEQ) {
        fprintf(FPO, "=");
        key=nkey;
        nkey=getkey();
        aflag=1;
    }
    pexpression();
    aflag=0;
}
else if(key==TINC) {
    fprintf(FPO, "+");
    key=nkey;
    nkey=getkey();
}
else if(key==TDEC) {
    fprintf(FPO, "--");
    key=nkey;
    nkey=getkey();
}
else Serror();/* ! ! ! ! エラーを出力 ! ! ! ! */
if(key==TSC) {
    fprintf(FPO, ";");
    newline();
    key=nkey;
    nkey=getkey();
}
else Serror();/* ! ! ! ! エラーを出力 ! ! ! ! */
if(pflag==0) fprintf(FPO, "_t++");
else fprintf(FPO, "_tp[_p]++");
newline();
}

```

```

void pleftside() {                            /* leftside */
    void pexpression();
    int i;
    if(key==THNAME) {
        if(pflag==1&&aflag==1&&strcmp(H, "_p") !=0) {
            fprintf(FPO, "__%s", H);
        }
        else fprintf(FPO, "%s", H);
        for(i=0; i<HN; i++) {

```

```

        if(strcmp(H, HTABLE[i])==0) break;
    }
    if(i==HN&&strcmp(H, "_p")!=0) Serror();/* !!!!!エラーを出力!!!! */
    key=nkey;
nkey=getkey();
    if(key==TKK0) {
        fprintf(FP0, "[");
            if(HTYPE[i]!=1&&HTYPE[i]!=5) Serror();/* !!!!!エラーを出力!!!! */
力!!!! */
            key=nkey;
nkey=getkey();
            pexpression();
            if(key!=TKKC) Serror();/* !!!!!エラーを出力!!!! */
            fprintf(FP0, "]");
            key=nkey;
nkey=getkey();
        }
        else if(HTYPE[i]==1||HTYPE[i]==5) Serror();/* !!!!!エラーを出力!!!! */
力!!!! */
    }
    else Serror();/* !!!!!エラーを出力!!!! */
}

void pexpression() { /* expression */
    void plogicalterm();
    plogicalterm();
    while(key==TLOR) {
        fprintf(FP0, "|");
        key=nkey;
        nkey=getkey();
            plogicalterm();
        }
    }

void plogicalterm() { /* logicalterm */
    void plogicalfactor();
    plogicalfactor();
    while(key==TLAND) {
        fprintf(FP0, "&&");
        key=nkey;
        nkey=getkey();
            plogicalfactor();
        }
    }
}

```

```

void plogicalfactor() {                               /* logicalfactor */
    void parithmeticexpression();
    parithmeticexpression();
    if (key==TEQEQ || key==TNEQ || key==TLESS || key==TLESSEQ || key==TMORE || key==TMOREEQ) {
        switch(key) {
            case TEQEQ:
                fprintf(FPO, "==" );
                break;
            case TNEQ:
                fprintf(FPO, "!=" );
                break;
            case TLESS:
                fprintf(FPO, "<" );
                break;
            case TLESSEQ:
                fprintf(FPO, "<=" );
                break;
            case TMORE:
                fprintf(FPO, ">" );
                break;
            case TMOREEQ:
                fprintf(FPO, ">=" );
                break;
        }
        key=nkey;
        nkey=getkey();
        parithmeticexpression();
    }
}

```

```

void parithmeticexpression() {                       /* arithmeticexpression */
    void parithmeticterm();
    parithmeticterm();
    while (key==TADD || key==TSUB) {
        switch(key) {
            case TADD:
                fprintf(FPO, "+" );
                break;
            case TSUB:
                fprintf(FPO, "-" );
                break;
        }
        key=nkey;
        nkey=getkey();
        parithmeticterm();
    }
}

```

```

    }
}

void parithmeticterm() { /* arithmeticterm */
    void parithmeticfactor();
    parithmeticfactor();
    while(key==TMUL||key==TDIV||key==TMOD) {
        switch(key) {
            case TMUL:
                fprintf(FPO, "*");
                break;
            case TDIV:
                fprintf(FPO, "/");
                break;
            case TMOD:
                fprintf(FPO, "%");
                break;
        }
        key=nkey;
        nkey=getkey();
        parithmeticfactor();
    }
}

void parithmeticfactor() { /* arithmeticfactor */
    void parithmeticfactor();
    void punsignedfactor();
    if(key==TNOT) {
        fprintf(FPO, "!");
        key=nkey;
        nkey=getkey();
        parithmeticfactor();
    }
    else if(key==TSUB) {
        fprintf(FPO, "-");
        key=nkey;
        nkey=getkey();
        parithmeticfactor();
    }
    else{
        punsignedfactor();
    }
}

void punsignedfactor() { /* unsignedfactor */

```

```

void pexpression();
void pleftside();
if(key==THNAME) {
pleftside();
}
else if(key==TINTEGER) {
fprintf(FPO, "%d", V);
key=nkey;
nkey=getkey();
}
else if(key==TCHARACTER) {
fprintf(FPO, "%s", V);
key=nkey;
nkey=getkey();
}
else if(key==TK0) {
fprintf(FPO, "(");
key=nkey;
nkey=getkey();
pexpression();
if(key==TKC) {
fprintf(FPO, ")");
key=nkey;
nkey=getkey();
}
else Serror();
}
}

void parallel() { /* parallel */
void pexpression();
void pstatement();
void pmas();
int i;
for(i=0; i<HN; i++) {
if(HTYPE[i]==0||HTYPE[i]==4) {
fprintf(FPO, "__%s=%s;", HTABLE[i], HTABLE[i]);
newline();
}
else if(HTYPE[i]==1||HTYPE[i]==5) {
fprintf(FPO, "for (_j=0; _j<%d; _j++)
__%s[_j]=%s[_j];", HSIZE[i], HTABLE[i], HTABLE[i]);
newline();
}
}
}

```

```

if (key==TPARALLEL) {
    fprintf(FP0, "for");
    key=nkey;
    nkey=getkey();
}
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
if (key==TK0) {
    fprintf(FP0, "(_p<(_t1=");
    key=nkey;
    nkey=getkey();
}
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
pexpression();
if (key==TCMA) {
    fprintf(FP0, ");_p<(_t2=");
    key=nkey;
    nkey=getkey();
}
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
pexpression();
    if (key==TKC) {
        fprintf(FP0, ");_p++)");
        key=nkey;
        nkey=getkey();
    }
else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
    pflag=1;
if (key==TMK0) {
    indent++;
    fprintf(FP0, "{");
    newline();
    key=nkey;
    nkey=getkey();
    fprintf(FP0, "_tp[_p]=0;");
    newline();
    while (key!=TMKC) {
        pstatement();
    }
    fprintf(FP0, "}");
    newline();
    key=nkey;
    nkey=getkey();
}
else {
    indent++;

```

```

        fprintf(FP0, "{");
        newline();
    fprintf(FP0, "_tp[_p]=0:");
    newline();
    pstatement();
    indent--;
    fprintf(FP0, "}");
        newline();
    }
    pflag=0;
    fprintf(FP0, "_t=_t+_pmax(_t1,_t2):");
    newline();
}

```

/* 構文解析ここまで */

```

int getkey() {
    int key;    /* 字句解析結果 */
    int l;
    void getC();
    do{
        getC();
    }
    while(c==' ' || c=='\t' || c=='\n');
    if(c>='a' && c<='z' || c>='A' && c<='Z' || c=='_') /* cは文字ですか? */
        l=0;
        H[l]=c;
        while(nc>='a' && nc<='z' || nc>='A' && nc<='Z' || nc=='_' || nc>='0' && nc<='9') /* cが文字か'_'か
数字なら16文字まで読み込み続ける */
            if (l<HSIZEMAX) {
                l++;
                H[l]=nc;
                getC();
            }
            else Serror();/* ! ! ! ! !エラーを出力 ! ! ! ! ! */
        }
        l++;
        H[l]='\0';
        if(strcmp(H, "main")==0)/* libraryの命令と比較して一致すればその命令と認識し、key
に代入 */
            key=TMAIN;
        else if(strcmp(H, "void")==0)
            key=TVOID;
        else if(strcmp(H, "int")==0)
            key=TINT;

```

```

else if(strcmp(H, "char")==0)
    key=TCHAR;
else if(strcmp(H, "if")==0)
    key=TIF;
else if(strcmp(H, "else")==0)
    key=TELSE;
else if(strcmp(H, "while")==0)
    key=TWHILE;
else if(strcmp(H, "for")==0)
    key=TFOR;
else if(strcmp(H, "include")==0)
    key=TINCLUDE;
else if(strcmp(H, "define")==0)
    key=TDEFINE;
else if(strcmp(H, "printf")==0)
    key=TPRINTF;
else if(strcmp(H, "scanf")==0)
    key=TSCANF;
else if(strcmp(H, "parallel")==0)
    key=TPARALLEL;
else
    key=THNAME;
}

else if(c>='0' && c<='9') { /* cが数字なら次の文字が数字である限り読みつづける */
    V=c-'0';
    while(nc>='0' && nc<='9') {
        getC();
        V=V*10+c-'0';
    }
    key=TINTEGER;
}
else if(c=='¥') {
    if(nc!='¥')
        V=nc;
    else Serror(); /* ! ! ! ! ! エラー ! ! ! ! ! */
    getC();
    if(nc=='¥') {
        printf("%c ", V);
        key=TCHARACTER;
        nc=getc(FPI);
        Nchar++;
    }
    else Serror(); /* ! ! ! ! ! エラー ! ! ! ! ! */
}
}

```

```

else if(c=='¥'){
    l=0;
    getC();
    while(c!='¥'){
        if (l<HSIZEMAX){
            H[l]=c;
            l++;
        }
        else Serror();/*!!!!エラー!!!!*/
        if(c==EOF) Serror();/*!!!!エラー!!!!*/
    }
    getC();
    }
    H[l]='¥0';
    key=TSTRING;
}
else if(c=='='){ /* == or = */
    if(nc=='='){
        key=TEQEQ;
        getC();
    }
    else
        key=TEQ;
}
else if(c=='>'){ /* >= or > */
    if(nc=='='){
        key=TMOREEQ;
        getC();
    }
    else
        key=TMORE;
}
else if(c=='<'){ /* <= or < */
    if(nc=='='){
        key=TLESSEQ;
        getC();
    }
    else
        key=TLESS;
}
else if(c=='+'){ /* ++ or + */
    if(nc=='+') {
        key=TINC;
        getC();
    }
}

```

```

        else
            key=TADD;
    }
    else if(c=='-'){ /* -- or - */
        if(nc=='-'){
            key=TDEC;
            getC();
        }
        else
            key=TSUB;
    }
    else if(c=='*') /* * */
        key=TMUL;

    else if(c=='%') /* % */
        key=TMOD;

    else if(c=='&'){ /* && or & */
        if(nc=='&'){
            key=TLAND;
            getC();
        }
        else
            key=TAND;
    }
    else if(c=='|'){ /* || or | */
        if(nc=='|'){
            key=TLOR;
            getC();
        }
        else
            key=TOR;
    }
    else if(c=='!'){ /* != or ! */
        if(nc=='='){
            key=TNEQ;
            getC();
        }
        else
            key=TNOT;
    }
    else if(c=='(') /* ( */
        key=TKO;

    else if(c==')') /* ) */

```

```

    key=TKC;

else if (c=='{') /* { */
    key=TMKO;

else if (c=='}') /* } */
    key=TMKC;

else if (c=='[') /* [ */
    key=TKKO;

else if (c==']') /* ] */
    key=TKKC;

else if (c==';') /* ; */
    key=TSC;

else if (c=='#') /* # */
    key=TSHP;

else if (c==',') /* , */
    key=TCMA;

else if (c=='/') { /* /* or / */
    if (nc=='*') {
        while (c!='*' || nc!='/') {
            nc=getc(FPI);
            Nchar++;
        }
        getC();
        if (nc==-1) { /* ncがファイルの終わりなら終了 */
            key=TNULL;
            break;
        }
    }
    nc=getc(FPI);
    Nchar++;
    return getkey();
}
else
    key=TDIV;
}

else if (c=='.') { /* . */
    if (nc=='h') {
        key=TDOTH;
        getC();
    }
}

```

```

        }
    else
        key=TDOT;
    }
else if(nc==-1)    /* ncがファイルの終わりなら終了 */
    key=TNULL;
else {
    exit(-1);
}
return key;    /*getkeyにkeyを返す*/
}

void Serror() {
    printf("%d行目の%d文字目を読んでいる時にエラーになりました%c", Nline, Nchar, c);
    exit(-1);
}

void getC() {
    c=nc;
    nc=getc(FPI);
    if(c=='\n') {    /* 何行目の何文字目を読んでいるかカウント */
        Nline++;
        Nchar=0;
    }
    else Nchar++;
}

```

付録 2

付録 2.1 足し算プログラム

本研究で用いた足し算プログラムを以下に示す。

付録 2.1.1 変換前 C 風言語プログラム

```
#include<stdio.h>
#include<stdlib.h>
#define n 100
void main() {
    int p, a[128], j, i;
    printf("Please input number of processor");
    scanf("%d", &p);
    for(i=0; i<n; i++) a[i]=1;
    for(j=1; j<n; j=j*2) {
        parallel(0, p-1) {
            for(i=n*_p/(j*p); i<n*(p+1)/(j*p); i++) {
                if(2*i+1<n) a[i]=a[2*i]+a[2*i+1];
                else if(2*i<n) a[i]=a[2*i];
                else a[i]=0;
            }
        }
    }
    printf("結果は%d", a[0]);
}
```

付録 2.1.2 変換後 C 言語プログラム修正前

```
#define _P 100
#include<stdio.h>
#include<stdlib.h>
#define n 100
int _t=0;
int _tp[_P];
int _p;
int _t1;
int _t2;
int _pmax();
int _j;
void main(){
```

```

int p=0,_p,a[128],_a[128],j=0,_j,i=0,_i;
printf("Please input number of processor");
scanf("%d",&p);
_t++;
for(i=0;i<n;i++) {
    _t=_t+2;
    a[i]=1;
    _t++;
}
_t++;
for(j=1;j<n;j=j*2) {
    _t=_t+2;
    {
        _p=p;
        for(_j=0;_j<128;_j++) _a[_j]=a[_j];
        _j=j;
        _i=i;
        for(_p=(_t1=0);_p<=(_t2=p-1);_p++){
            _tp[_p]=0;
            _tp[_p]++;
            for(i=n*_p/(j*p);i<n*(p+1)/(j*p);i++) {
                _tp[_p]=_tp[_p]+2;
                {
                    if(2*i+1<n) {
                        _tp[_p]++;
                        a[i]=_a[2*_i]+_a[2*_i+1];
                        _tp[_p]++;
                    }
                    else {
                        _tp[_p]++;
                        if(2*i<n) {
                            _tp[_p]++;
                            a[i]=_a[2*_i];
                            _tp[_p]++;
                        }
                        else {
                            _tp[_p]++;
                            a[i]=0;
                            _tp[_p]++;
                        }
                    }
                }
            }
        }
    }
}
_t=_t+_pmax(_t1,_t2);

```

```

        }
    }
    printf("結果は%d",a[0]);
    printf("Time is %d",_t);
}
int _pmax(_t1,_t2) int _t1;int _t2; {
    int max=_t1,i;
    for(i=_t1+1;i<=_t2;i++) if(_tp[max]<_tp[i]) max=i;
    return(_tp[max]);
}

```

付録 2.1.3 変換後 C 言語プログラム修正後

```

#define _P 100
#include<stdio.h>
#include<stdlib.h>
#define n 100
int _t=0;
int _tp[_P];
int _p;
int _t1;
int _t2;
int _pmax();
int _j;
void main() {
    int p=0, __p, a[128], __a[128], j=0, __j, i=0, __i;
    printf("Please input number of processer");
    scanf("%d", &p);
    _t++;
    for(i=0;i<n;i++) {
        _t=_t+2;
        a[i]=1;
        _t++;
    }
    _t++;
    for(j=1;j<n;j=j*2) {
        _t=_t+2;
        {
            __p=p;
            for(_j=0;_j<128;_j++) __a[_j]=a[_j];
            __j=j;
            __i=i;
            for(_p=(_t1=0);_p<=(_t2=p-1);_p++) {
                _tp[_p]=0;
                _tp[_p]++;
            }
        }
    }
}

```

```

for(i=n*_p/(j*p);i<n*(p+1)/(j*p);i++) {
    _tp[_p]=_tp[_p]+2;
    {
        if(2*i+1<n) {
            _tp[_p]++;
            a[i]=__a[2*i]+__a[2*i+1];
            _tp[_p]++;
        }
        else {
            _tp[_p]++;
            if(2*i<n) {
                _tp[_p]++;
                a[i]=__a[2*i];
                _tp[_p]++;
            }
            else {
                _tp[_p]++;
                a[i]=0;
                _tp[_p]++;
            }
        }
    }
}
}
}
_t=_t+_pmax(_t1,_t2);
}
}

printf("結果は%d",a[0]);
printf("Time is %d",_t);
}

int _pmax(_t1,_t2) int _t1;int _t2; {
    int max=_t1,i;
    for(i=_t1+1;i<=_t2;i++) if(_tp[max]<_tp[i]) max=i;
    return(_tp[max]);
}
}

```