ハイパフォーマンス
コンピューティング 82−17
（２０００．８．４）

学振 MIRAI コンパイラにおける剰余区間伝播の設計

中西 恒夫　　福田 晃

奈良先端科学技術大学院大学情報科学研究科
{*tun, fukuda*}*@is.aist-nara.ac.jp*

概要

自動並列化コンパイラにおける依存解析ではループ指標や配列添字の値域情報が必要となる．一般に値域は区間，すなわち上限値と下限値とによって表現されるが，1より大きなストライドのループにおいては区間による値域表現は正確さを欠き，ひいては依存解析精度の低下につながる．本稿では，依存解析精度の向上を図り，スカラ変数の値域表現のために区間に法と剰余を追加した剰余区間を用い，単調データフローシステムによってスカラ変数の値域解析を行う剰余区間伝播のアルゴリズムを設計する．

Design of Modulo Interval Propagation in the JSPS MIRAI Compiler

Tsuneo Nakanishi　Akira Fukuda

Graduate School of Information Science
Nara Institute of Science and Technology

Abstract

Dependence analysis for parallelizing compilers requires range information of loop indices and array subscripts. Range information of a scalar variable is represented by the interval consisting of the upper and lower bounds of the variable in general. The interval is not accurate representation in loops with strides greater than one and can provide false dependence information which disables parallelization. In this paper we employ the modulo interval, the interval with the modulus and the residue, for range representation and develop an algorithm, called *modulo interval propagation*, to evaluate ranges of scalar variables with the monotone dataflow system.

1 はじめに

依存解析は科学技術計算向け自動並列化コンパイラの並列化性能を大きく左右する重要なプログラム解析である．多くの依存解析手法において重要な情報として，ループ指標や配列添字の採り得る値集合の情報，すなわち値域情報がある．変数の値域情報の解析，すなわち値域解析は，すでに70年代から行われており [2]，自動並列化コンパイラの分野においても依存解析や配列参照解析の必要から研究されている．また近年では，高位論理合成の分野においても，データパス幅削減によるハードウェアコストを図って，変数の値域解析が研究されている [3, 5]．

値域情報の最も簡便な表現法として，値域の上限値と下限値の対による表現，区間があり，区間に対する演算体系として区間演算が定義されている．区間演算はしばしば値域解析において用いられる．しかしながら区間は，上限値と下限値の間の連続する整数の集合しか表現し得ず，1より大きなストライ

ドのループにおけるループ指標や配列参照の添字式の値域を表現するには正確さを欠く．生成される並列プログラムの元の逐次プログラムに対する意味的等価性を保証するべく，依存解析は「控え目」に行われるため，不正確な値域の表現は不十分な並列化につながる．以上の考察に基づいて，本研究では従来の区間演算を拡張し，文献[8]において，実数上の区間に含まれる整数の集合に対する集合演算と算術演算からなる演算体系，剰余区間演算を提案および定義している．

剰余区間演算は記号 $[a,b]_{n(r)}$ で表され，区間 $[a,b]$ 内の整数のうち，整数 n で整除する場合の余りが r になるような整数の集合を意味する．従来の区間に対して追加された情報は法 n ならびに余り r のみであるが，ループのイタレーションの集合，あるいはそれらから参照される配列要素の添字の集合を表現する上で有利な表現であり，また整数論で裏付けられる興味深い数学的性質を多く有する．本稿ではこの剰余区間演算を用いて，プログラム中のスカラ変数，あるいはそれらを引数とする式が採り得る値集合を求める解析技術，剰余区間伝播の提案とその実装について述べる．ここでスカラ変数にはループ指標，またそれらを引数とする式にはループ中の配列参照の添字式も含まれることに注意されたい．

以下，本稿では第2節において剰余区間演算の基本性質について説き，第3節において剰余区間伝播のアルゴリズムについて述べる．剰余区間伝播では，ループ中の誘導変数の値域解析に多大な計算時間を要し得るが，その対策を第4節において述べる．最後に第5節でまとめと関連研究，今後の課題について述べる．

2 剰余区間演算

剰余区間は記号 $[a,b]_{n(r)}$ で表され，実数上の区間 $[a,b]$ 内の整数のうち，整数 n で整除する場合の余りが r になるような整数の集合を意味し，形式的には以下のように定義される集合である．

$$\{x \in \boldsymbol{Z} | a \le x \le b, x = nm + r, m \in \boldsymbol{Z}\}$$

四則演算，累乗など整数上に定義される任意の演算 $\odot$ は，剰余区間に対する演算として以下のように再定義される．但し，A，B は任意の剰余区間，z は任意の整数である．

$$\begin{aligned} A \odot B &= \{x = a \odot b | a \in A, b \in B\} \\ A \odot z &= \{x = a \odot z | a \in A\} \\ z \odot B &= \{x = z \odot b | b \in B\} \end{aligned}$$

剰余区間どうしの加法，減法，乗法については以下の性質が成り立つ．

性質 1 （加法・減法・乗法に関する性質）剰余区間どうしの加法，減法，乗法について次の性質が成り立つ．

$$\begin{aligned} &[a,b]_{m(r)} + [c,d]_{n(s)} \\ &\quad \subseteq \ [a+c, b+d]_{\gcd(m,n)(r+s)} \\ &[a,b]_{m(r)} - [c,d]_{n(s)} \\ &\quad \subseteq \ [a-d, b-c]_{\gcd(m,n)(r-s)} \\ &[a,b]_{m(r)} * [c,d]_{n(s)} \\ &\quad \subseteq \ [\min\{ac, ad, bc, bd\}, \\ &\qquad \max\{ac, ad, bc, bd\}]_{\gcd(m,n)(rs)} \end{aligned}$$

（証明は文献[8]参照.） □

性質1において一般的には等号は成立しないが，次の特別な場合においては成立する．

性質 2 （法の等しい剰余区間の加法・減法）法の等しい正規形の剰余区間どうしの加法，減法について次の性質が成り立つ．

$$\begin{aligned} [a,b]_{n(r)} + [c,d]_{n(s)} &= [a+c, b+d]_{n(r+s)} \\ [a,b]_{n(r)} - [c,d]_{n(s)} &= [a-d, b-c]_{n(r-s)} \end{aligned}$$

（証明は文献[8]参照.） □

さらに剰余区間と整数との演算については次の性質が成り立つ．性質 1 の場合とは異なり，これらの性質は等式として表される．

性質 3 （**剰余区間と整数との演算に関する性質**）剰余区間と整数との加法，減法，乗法について次の性質が成り立つ．

$$
\begin{aligned}
[a,b]_{n(r)} + z &= [a+z, b+z]_{n(r+z)} \\
z + [a,b]_{n(r)} &= [z+a, z+b]_{n(z+r)} \\
[a,b]_{n(r)} - z &= [a-z, b-z]_{n(r-z)} \\
z - [a,b]_{n(r)} &= [z-b, z-a]_{n(z-r)} \\
[a,b]_{n(r)} * z &= [a*z, b*z]_{nz(rz)} \\
z * [a,b]_{n(r)} &= [a*z, b*z]_{nz(rz)}
\end{aligned}
$$

（証明は文献 [8] 参照.） □

また剰余区間を単に整数の集合とみなせば，剰余区間に対して集合演算を定義することができる．剰余区間の共通部分に関しては次に挙げる性質 4 が成り立つ．

性質 4 （**剰余区間の共通部分**）任意の 2 つの剰余区間 $[a,b]_{m(r)}$，$[c,d]_{n(s)}$ の共通部分は

$$
\begin{aligned}
&[a,b]_{m(r)} \cap [c,d]_{n(s)} \\
&= \{x \in \boldsymbol{Z} \,|\, \max\{a,c\} \le x \le \min\{b,d\}, \\
&\qquad x \equiv r \pmod m, x \equiv s \pmod n\}
\end{aligned}
$$

で与えられる.（証明は文献 [8] 参照.） □

性質 4 で与えられる剰余区間の共通部分を求めるには，連立合同方程式：

$$
\left\{
\begin{array}{lcll}
x & \equiv & r & (\mathrm{mod}\ m) \\
x & \equiv & s & (\mathrm{mod}\ n)
\end{array}
\right.
$$

を解く必要がある．このような連立合同方程式の解は中華剰余定理（Chinese Remainder Theorem）により求められる．法の等しい剰余区間の共通部分については次の性質 5 により簡単に求めることができる．

性質 5 （**法の等しい剰余区間の共通部分**）任意の 2 つの法の等しい剰余区間 $[a,b]_{n(r)}$，$[c,d]_{n(s)}$ の共通部分は次式で与えられる．

$$
\begin{aligned}
&[a,b]_{n(r)} \cap [c,d]_{n(s)} \\
&= \left\{
\begin{array}{ll}
[\max\{a,c\}, \min\{b,d\}]_{n(r)} & \text{if } r = s \\
\emptyset & \text{if } r \neq s
\end{array}
\right.
\end{aligned}
$$

（証明は文献 [8] 参照.） □

3 剰余区間伝播

3.1 単調データフローシステム

本稿で議論する変数の値域解析は単調データフローシステム（monotone data flow system）[6] 上の問題としてモデル化される．単調データフローシステムは，制御フローグラフ（CFG: Control Flow Graph）上にデータフロー情報を順次伝播させることで，定数伝播，変数の生存解析，到達定義式の解析などコンパイラにおける多くのデータフロー問題を解決する．伝播するデータフロー情報の表現は解決するデータフロー問題に応じて定義され，本稿で議論する変数の値域解析の場合は剰余区間で表される各変数の値域情報の集合である．各節点において，自身の直前に位置する節点の保有するデータフロー情報をその節点に対応する演算に応じて加工したものを集めて自身のデータフロー情報とすることにより，データフロー情報を CFG 上に伝播させる．

単調データフローシステム上の問題は全てアルゴリズム 1 で解決できる．但し，V は CFG の節点集合，任意の節点 $v \in V$ について定義される $\mathrm{INF}(v)$ は v の保有するデータフロー情報，$\mathrm{pred}(v)$ は v の直前に位置する節点の集合，f_v は v におけるデータフロー情報の加工を表す関数である．

アルゴリズム 1

1. 全ての $v \in V$ について $\mathrm{INF}(v) := \emptyset$.
2. $stable := true$.

3. 任意の $v \in V$ について,

(a) $tmp = \bigcup_{u \in \mathrm{pred}(v)} f_u(\mathrm{INF}(u))$.

(b) $tmp \neq \mathrm{INF}(v)$ ならば $\mathrm{INF}(v) := tmp$, $stable := false$.

4. $stable = false$ ならば 2 へ飛ぶ. □

3.2 階層タスクグラフ（HTG）

学振 MIRAI コンパイラでは，その中間表現に階層タスクグラフ（HTG: Hierarchical Task Graph）[1] を拡張したものを使用している．階層タスクグラフは図 1 のようにループ単位で階層化された構造となっている．ループを表すループ節点の内部には，そのループ本体の文を表す節点と START 節点，STOP 節点が存在する．START 節点は入枝を持たないループの入口を表す節点であり，START 節点から他の節点へは常にパスが存在する．STOP 節点は出枝を持たないループの出口を節点であり，他の節点から STOP 節点へは常にパスが存在する．

単調データフローシステムは，階層構造を持たない CFG に対して定義されているものであるが，本稿ではこれを HTG に適用できるようにするべく，HTG の任意の節点 v について定義される $\mathrm{pred}(v)$ を再定義する．すなわち，v が START 節点の場合は v を直接含むループ節点の直前に位置する節点ならびに対応する STOP 節点からなる集合を，v がループ節点の場合は v が直接含む STOP 節点のみからなる集合を，v がその他の節点の場合は CFG の場合と同様に v の直前に位置する節点の集合を $\mathrm{pred}(v)$ とする．

3.3 剰余区間伝播

スカラ変数に関する値域情報を剰余区間で表現し，単調データフローシステムによってスカラ変数の値域を求める解析技術を剰余区間伝播と呼ぶことにする．剰余区間伝播では，HTG に伝播させるデータフロー情報，すなわち各スカラ変数の値域情報を，スカラ変数とその値域を含む剰余区間からなる組の集合として定義する．HTG の各節点は，自身の保有する値域情報をその節点に対応する演算に応じて加工したものを，自身に続く節点に伝播させなければならない．各節点での値域情報の加工の方法，すなわちアルゴリズム 1 における f_v の定義は，下記のようにその節点の種類に応じて異なる．

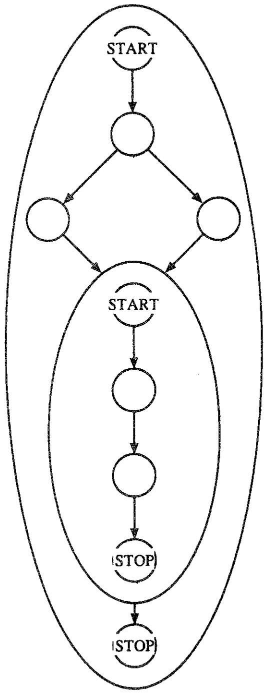

図 1: 階層タスクグラフ

代入文節点の場合：v が代入文 $y = g(x_1, x_2, \ldots)$ を表す節点の場合，f_v は次のように定義される．

$$f_v(X) = X \cup \{(y, g(X_1, X_2, \ldots))\}$$

但し，X_i はスカラ変数 x_i の値域を含む剰余区間であり，(x_i, X_i) は $\mathrm{INF}(v)$ に含まれているものである．スカラ変数 y の値域情報 $g(X_1, X_2, \ldots)$ は，性質 1–3 に従って算出する．いずれの性質も実際より大きな値域を与え依存解析の精度を下げ得るが，値

域を大きく見積もる方が安全に依存解析が行われることに注意されたい．なお剰余区間演算に伴う誤差の削減については文献 [7] を参照されたい．

START 節点の場合： v が START 節点の場合，f_v は次のように定義される．

$$f_v(X) = X \cup \{(i, I)\}$$

但し，i は START 節点を直接含むループ節点に対応するループの指標となるスカラ変数，I はその値域を含む剰余区間である．

STOP 節点の場合： v が STOP 節点の場合，f_v は次のように定義される．

$$f_v(X) = X$$

ループ節点の場合： v がループ節点の場合，f_v は次のように定義される．

$$f_v(X) = X$$

条件分岐節点の場合： v が条件式 $p(x_1, x_2, ..., x_n)$ の条件分岐節点の場合，自身の保有する値域情報を加工し，p の値が真または偽の時に次に実行する文の節点に伝播させる際の，値域情報の加工を表す関数，それぞれ f_v^T，f_v^F は次のように定義される．

$$\begin{aligned} f_v^T(X) &= X \cap X_p \\ f_v^F(X) &= X \cap X_{\neg p} \end{aligned}$$

但し，X_p，$X_{\neg p}$ はそれぞれ，条件式 p の値を真または偽とし得る，変数 x_i の値を全て含む剰余区間 X_i からなる値域情報 (x_i, X_i) の集合である．

このような値域情報は次のように求めることができる．例えば，$p(x,y)$ が $x + y \leq 100$，$X = \{(x, [1, 99]_{2(1)}), (y, [1, 99]_{2(1)})\}$ の場合，p の x に $[1, 99]_{2(1)}$ を代入し，

$$\begin{aligned} y &\leq 100 - [1, 99]_{2(1)} \\ y &\leq [1, 99]_{2(1)} \\ y &\in [-\infty, 99]_{1(0)}. \end{aligned}$$

同様に p の y に $[1, 99]_{2(1)}$ を代入し，

$$x \in [-\infty, 99]_{1(0)}.$$

$X_p = \{(x, [-\infty, 99]_{1(0)}), (y, [-\infty, 99]_{1(0)})\}$ を得る．最終的に $f_v^T(X) = \{(x, [1, 99]_{2(1)}), (y, [1, 99]_{2(1)})\}$ が得られる．$f_v^T(X)$ の x ならびに y の値域には，$x + y \leq 100$ とし得る x と y の値を全て含んでいるが，$x + y \leq 100$ としない x と y の組も含み，実際より大きく見積もられていることに注意されたい．

4 誘導変数の値域解析

ループ中の誘導変数の値域解析は多大な計算時間を消費し得る．例えば，次の簡単なプログラムでさえ，アルゴリズム 1 をそのまま適用すると，変数 SUM の値域情報が得られるまでに 100 回アルゴリズム 1 のステップ 2–4 のループを回り，結局プログラムを実行して SUM の値域を調べるのと計算量が変わらない．さらに，SUM の実際の値域は $[0, 5050]_{1(0)}$ であるのに，$[0, 10000]_{1(0)}$ と過大に値域を見積もる．

```
SUM = 0
DO I=1,100
    SUM=SUM+I
ENDDO
```

このような誘導変数の値域を求める一般的なアルゴリズムを開発することは，漸化式を一般式に変換するアルゴリズムを開発することに等しく容易ではない．そこで誘導変数の値域解析に関しては，総和計算，最小／最大値計算（いわゆる reduction 計算）等の代表的な代入文について個別に対処することにする．学振 MIRAI コンパイラの中間表現は，代入文に対応する HTG 節点の下にその代入文を表す抽象構文木（AST: Abstract Syntax Tree）がぶら下がっている構造になっており，この AST にパターンマッチングを適用することにより，誘導変数の値域を個別に解析する．対処できないパターンに関しては，安全な依存解析のため，関係する誘導変数の値域を最大に見積もる．

5 まとめ

本稿では，自動並列化コンパイラにおける依存解析のために，単調データフローシステムを用いてスカラ変数の値域を剰余区間の形で求める，剰余区間伝播のアルゴリズムを示した．

本研究に近い位置にあるものとして，一般の区間を用いて値域解析を行う小川らの研究 [3] が挙げられる．本研究は，i) 値域表現に剰余区間を用いている点，ii)HTG に対応している点，iii) より複雑な条件分岐に対応している点（小川らの研究で対応している条件分岐は $x \odot y$ のタイプの変数同士の比較演算である）で異なる．また，剰余区間演算に近い研究として，Paek らの LMAD (Linear Memory Access Descriptor) [4] があるが，LMAD は部分配列表現法であり，本稿で議論している値域解析に応用されるものではない．

剰余区間伝播は，学振 MIRAI コンパイラに現在実装中であり，その効果の検証は今後の課題である．また，困難な誘導変数の値域解析はパターンマッチングで対処するが，実アプリケーションにおける誘導変数の出現パターンの洗い出しも重要な今後の課題である．

謝辞

本研究は日本学術振興会未来開拓学術研究推進事業の助成を受けています．

参考文献

[1] M. Girkar and C. D. Polychronopoulos, "The Hierarchical Task Graph as a Universal Intermediate Representation," *Int. J. of Parallel Programming*, Vol.22, No.5, pp.519–551, Oct. 1994.

[2] W. H. Harrison, "Compiler Analysis of the Value Ranges for Variables," *IEEE Trans. on Software Engineering*, Vol.SE-3, No.3, pp.243–250, May 1977.

[3] O. Ogawa, K. Takagi, Y. Itoh, S. Kimura, and K. Watanabe, "Hardware Synthesis from C Programs with Estimation of Bit Length of Variables," *IEICE Trans. on Fundamentals*, Vol.E82-A, No.11, pp.2338–2346, Nov. 1999.

[4] Y. Paek, J. Hoeflinger, and D. Padua, "Simplification of Array Access Patterns for Compiler Optimization," *Proc. of the 1998 ACM SIGPLAN Conf. on Programming Language Design and Implementation*, pp.60–71, May 1998.

[5] H. Yamashita, H. Tomiyama, A. Inoue, E. F. Nurprasetyo, T. Okuma, and H. Yasuura, "Variable Size Analysis for Datapath Width Optimization," *Proc. of Asia Pacific Conf. on Hardware Description Languages (APCHDL '98)*, pp.69–74, July 1998.

[6] H. Zima, *Supercompilers for Parallel and Vector Computers,* ACM Press, 1990.

[7] 曽山 典子, 中西 恒夫, 加古 富志雄, 福田 晃,「多項式における剰余区間演算誤差削減のための演算規則」, 情処学会数理モデル化と問題解決シンポジウム論文集, pp.223-230, 2000 年 3 月.

[8] 中西 恒夫, 福田 晃, 「剰余区間演算の定義とプログラム解析への応用」, JSPP2000 論文集, pp.293–300, 2000 年 5 月.