3C−8

# 組合わせ幾何を用いた有理数プレスブルガー文真偽判定アルゴリズムにおける投影操作の高速化

柴田 直樹　岡野 浩三　東野 輝夫　谷口 健一
大阪大学 大学院基礎工学研究科 情報数理系専攻

## 1 まえがき

加算を持つ有理数の理論 (有理数変数, 有理数定数,+, −,=,<,∧,∨,∀,∃ からなる理論) の冠頭形の閉論理式 (以降 PRP 文と呼ぶ) の真偽判定ルーチンはプロトコルのテスト, ハードウェアのタイミング検証などに利用される [1]. 筆者らは以前時間計算量が $r\alpha^{\beta d}n^{\gamma d(b+1)^a}$($n$ は入力の式に含まれる不等式の個数, $d$ は変数の個数, $a$ は限定子交替数, $b$ は同じ限定子が続く最大の個数, $r$ は入力の式の各係数と定数の分母, 分子のビット数,$\alpha,\beta,\gamma$ は定数) の計算幾何学の手法を利用した PRP 文真偽判定アルゴリズムを提案した [3]. 本稿では, 筆者らが提案したアルゴリズムの実装上の高速化手法について述べる.

## 2 以前に考案したアルゴリズム

まず, 3 次元の場合について必要な概念を定義する. 平面で空間を平面の一方, もう一方, そして平面に含まれる空間の 3 つに分割することを考える. 平面の集合 $H$により空間は, 点, 両端を含まない線分, 外周を含まない多角形, 外面を含まない多面体に分割される. こうして分割されたそれぞれの部分をフェイスと呼ぶ. こうして空間が分割されてできたフェイス全ての集合を $H$ のアレンジメントと呼ぶ.

入力の PRP 文を $F = \forall x\exists y\exists z\{x \geq 0 \wedge y \geq 0 \wedge [(z - x \geq 0 \wedge y + z \leq 3/4) \vee (x \leq 1/4 \wedge z \leq 1/4)]\}$ とする. 最初に PRP 文 $F$の母式 $E$より得られたアレンジメント $A$ の各フェイス $f$に, $f$に含まれる任意の点の座標を母式に代入して得られる真偽値を割り当てる (図 1(a)). 次に変数 $z$を消去する操作を行う. これは $E$を真にする $z$が存在する $x,y$の値の領域を $xy$平面上に図示する操作である. これによって得られた図が図 1(b) である. 投影を作る操作は図 1(a) 中の直線の $xy$平面への投影 (直線) 全ての集合から 2 次元のアレンジメントを作り, 図 1(a) の真になる部分の影に含まれるフェイスにのみ真を割り当てることで行う.

入力の論理式の限定子の並びは $\forall x\exists y\exists z$なので, 同様に $\exists y$を消去して, 1 次元の真偽を割り当てたアレンジメント (図 1(c)) を得る. 実際のアルゴリズムでは連続する同じ限定子で束縛された変数は一度に消去する.

∃ を消去する操作が, 真になる領域に座標軸に並行な光を当ててできる影を真とする領域であるのに対し, ∀ を消去する操作は, 偽になる領域に座標軸に並行な光を当ててできる影を偽とする領域である. 残った $\forall x$ を消去し,0 次元の真偽が割り当てられたアレンジメントを得る. これは偽が割り当てられた 1 つの頂点からなる. したがって, 式全体は偽と判定される.

## 3 高速化のための工夫

以下では上記のアルゴリズムに対する実装上の高速化法について述べる. 高速化の工夫は次の 3 点からなる. 1. フェイスの投影から直接アレンジメントを作ることで真偽判定に必要ないフェイスを処理する手間を減らす. 2. 投影のアレンジメントを作る際のフェイスの交差判定の回数を減らす. 3. アレンジメントから真偽判定に必要ないフェイスをなくす. これらの高速化によって, アルゴリズムの最悪時間計算量は改善しないが, 平均時間計算量は大幅に改善すると考えられる [3].

### 3.1 フェイスの投影から直接アレンジメントを作る

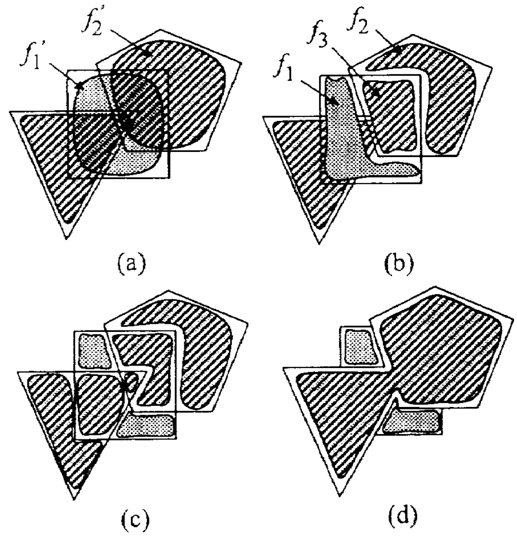


図 2 フェイス同士の重なりがなくなるようにする過程
**Fig. 2 Redivision of crossing faces.**

高速化を行う前のアルゴリズムでは, 投影を取る前のアレンジメントを構成する平面の投影全ての集合を求め, それらから投影のアレンジメントを作る. アレンジメントに含まれるフェイスのうち真偽判定に必要なのは, 真と偽の領域の境界に位置するフェイスのみである. 高速化を行ったあとのアルゴリズムでは, 真と偽の境界に位置しないフェイスをはじめからなるべく作らないようにするため, 次のように投影のアレンジメントを作る. まず投影を取る前のアレンジメントの各フェイスの投影の集合を求める (図 2(a))(図 2において, 斜線のフェイスが真が割り当てられたフェイスである). この状態ではフェイス同士が重なっているので, 重なり

A Technique for Reducing Computation Time of Projection in a Decision Procedure for Rational Presburger Sentences.
Naoki Shibata, Kozo Okano, Teruo Higashino and Kenichi Taniguchi
Division of Informatics and Mathematical Science, Osaka University
Toyonaka-shi, Osaka 560-8531, Japan

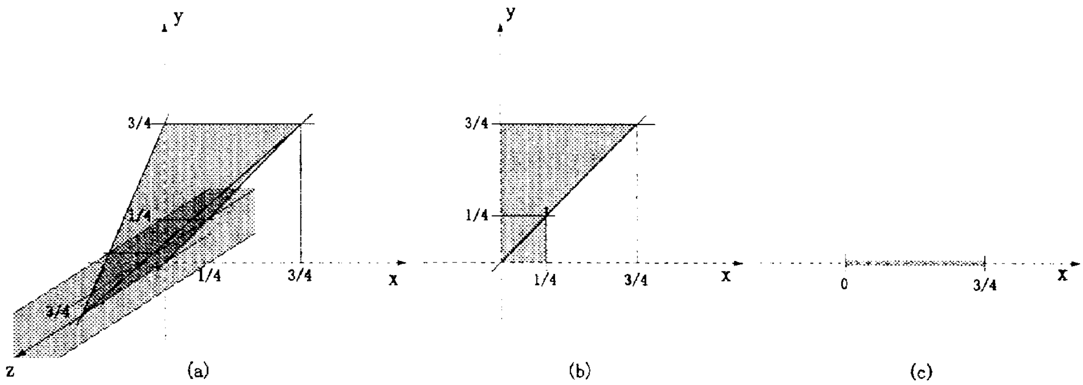


**図 1** アルゴリズムの適用によって変化していくアレンジメント

**Fig. 1** Transformation of the arrangement in execution of the algorithm.

あったフェイスを分割して重ならないようにする (図 2 (b)). 各フェイスに割り当てる真偽値は, 次のようになる : $f_1, f_2$にはそれぞれ $f'_1$と $f'_2$に割り当てられている真偽値を割り当て, $f_3$には消去する変数が $\exists, \forall$ で束縛されている場合に $f_1, f_2$割り当てられている値のそれぞれ論理和, 論理積を割り当てる. 全ての重なりあったフェイスに対し同様の処理を行うと図 2 (c)(図は消去する変数が $\exists$ で束縛されている場合) のようになる.

### 3.2 フェイスの交差判定回数の削減

上で述べた高速化を行う上で, フェイス同士が重なっているか調べる必要がある. ある 2 つのフェイスが重なっているかの判定は線形計画問題に帰着できる. したがって, フェイスの数の 2 乗回の線形計画問題を解く必要が生じる. この回数をなるべく減らすために次のような高速化を行う. 空間を 1 つの超平面の片側 (上側) ともう一方 (下側) の 2 つに分割し, それぞれのフェイスに対し「上側にのみ交わる[1]」,「下側にのみ交わる」,「上側と下側の両方に交わる」のいずれであるかあらかじめ調べておく. 上側にのみ交わるフェイスと下側にのみ交わるフェイスは交差しないので, これらのフェイス同士の交差判定は省略できる. 空間をさらに細かく分割することで, 最良の場合フェイス同士の交差判定の回数はフェイスの数 $n$ に対して $O(n \log n)$ 程度にまで少なくできる. また, 最悪の場合でもフェイス同士の交差判定の回数は変わらない.

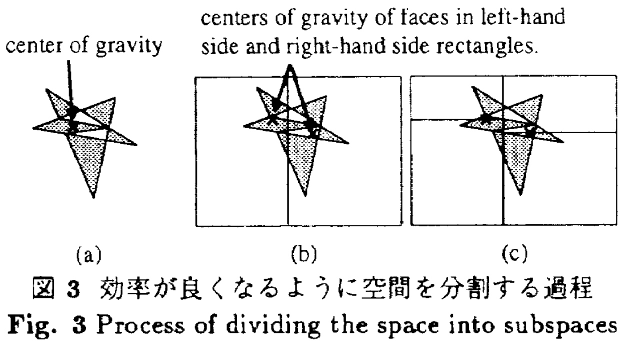


**図 3** 効率が良くなるように空間を分割する過程

**Fig. 3** Process of dividing the space into subspaces in order to improve efficiency.

上記の高速化法が有効に働くようにするためには次のような方法で空間を分割する. まず, アレンジメント中の各フェイスに均等な重さがあると仮定して, アレンジメントの重心を求める (図 3 (a)). 次に, 求めた重心を通るような超平面で空間を分割する (図 3 (b)). 分割によって得られた各部分空間に対しても, 同様に重心を通るような超平面で分割することを何度か繰り返す. この方法により, 時間計算量が最良, 最悪の場合, それぞれ $O(n \log n), O(n^2)$ 程度で空間を分割することができる[2].

### 3.3 フェイスの併合によるフェイス数の削減

3.1で述べた方法でフェイス同士の重なりがなくなるようにしたアレンジメントに対して, 真と偽の境界に位置しないフェイスをアレンジメントから取り除く処理を行う. アレンジメントの各フェイス $f$に対して, $f$と隣接する, $f$より次元の大きいフェイスに全て $f$と同じ真偽値が割り当てられていれば, $f$を取り除き, $f$と隣接する, より次元の大きいフェイスを一つに併合する. 併合してできる凹のフェイスも扱うことにする (図 2 (d)).

## 4 あとがき

本稿では筆者らが以前提案した PRP 文真偽判定アルゴリズムに対する実装上の高速化手法を提案した.

今後の課題として, 今回提案した手法を使った真偽判定ルーチンを製作を行うこと, 高速化の効果を計測することなどが挙げられる.

## 参考文献

[1] 東野輝夫, 北道淳司, 谷口健一 : “整数上の線形制約の処理と応用”, コンピュータソフトウェア, Vol.9, 6, 31-39 (1992).

[2] Hopcroft, J.E. and Ullmann, J.D.: “Introduction to Automata Theory, Languages and Computation,” Addison-Wesley,(1979).

[3] 柴田直樹, 岡野浩三, 東野輝夫, 谷口健一 : “有理数プレスブルガー文の真偽判定アルゴリズムの提案とその高速化手法”, 1998 年夏の LA シンポジウム予稿集, pp.4.1-4.6, 1998.

[4] Edelsbrunner, H.: “Algorithms in Combinatorial Geometry”, Springer-Verlag Berlin Heidelberg, (1987).

[1] 正確には上側の部分空間とのみ共有点を持つ.

[2] この時間計算量は分割後のフェイス同士の交差判定の回数にほぼ比例する.