数理解析研究所講究録 1295 巻 2002 年 35-41

極限コンパスの実装

宮本 健司
法政大学工学部*

1 はじめに

本稿ではコンピュータ作図における無限回の工程およびその極限を指定する方法とその実装について述べる.

コンピュータ作図はグラフィックによるユーザインタフェース (GUI) により定規やコンパス等に相当する限定されたコマンドを使って描画を行なうものである. 紙に描く作図との大きな違いは作図工程の起点となる点の配置をあとから変更できることである. たとえば, 角の二等分角を作図する作業が終ったあとで所与の角を変更したとき, ユーザは同じ作業を繰り返すことなく新しい二等分角を得ることができる. このような機能を実現するには, 作図コマンドの発行によって生成される幾何学的制約をコンピュータが記録しておき, 変更の要求に応じて制約の依存関係を手繰り寄せ影響を受ける部分を抽出して再実行すればよい.

作図におけるこのような再利用のしくみをさらに積極的に利用しようとする立場では起点の (可変の) 初期配置を入力, 作図で最終的に得られる図形を出力とするような関数を考え, 作図をその関数のプログラムとみなす [1][2]. Jackiw らは物体 (図形) の影や遠近図法への応用を示した [1]. (光源や消失点の位置変更に応じて影や構図が変化する.) 著者らの PDraw [2] は再帰と条件分岐を含む図形書き換え規則にもとづいており幾何学的制約に加えて (辺の数などに符号化するかたちで) 算術的制約も利用できる.

作業工程がそのままプログラムとしてつかえる反面, 作図可能性の限界により応用範囲が限定されている. そこでコンピュータ作図の操作性と再利用の利点を生かしながらいかにして表現力を向上させるかが問題になる.

本研究はプログラムとみなした作図に (プログラミング言語的な要素である) 不動点計算をいわば逆輸入して作図の表現力の拡大を試みるものである. 具体的には, 作図工程が意味する関数達に作用する不動点作用素を導入し, それに図形的な表現を与える. 不動点作用素は操作的には無限の繰り返しを意味するが, 同時にこの繰り返しによって生成される点列の極限を戻り値として記号的に生成する. コンピュータ作図における実装では通常の交点の計算などと同じく, 不動点作用素の使用を記号的に記録しておいて, 繰り返しと極限の実際の計算は画面の広さや解像度に応じて必要な回数/精度だけ行なう. これにより実際の計算は有限時間で終り, 生成された極限は直ちに引き続く工程で利用できる. これに対して無限の計算をおこなう理想的な装置を**極限コンパス**と呼ぶことにしよう. 本研究は極限コンパスの, コンピュータ作図としての実装であるということができる.

コンピュータ作図システム LIMIT は極限コンパスを実装した実験用システムであり, 極限コンパスのための GUI も備えている. 角の 3 等分や自然対数の底 (Napier 数) の作図などを描画操作だけで行なうことができる.

もとより作図は幾何学的関係の証明であるからこれをプログラムとして利用することは自然である. これをさらに計算論的に拡大することにより計算に適した幾何学構築できる. 以下の節では通常のコンピュー

*miyaken@k.hoesi.ac.jp

$\longrightarrow Node$	$Node$ はユーザが置いた点（**自由点**）.
$(Circle1, Circle2) \longrightarrow Node$	$Node$ は $Circle1$ を左に見た時の下側の交点.
$(Circle, Line) \longrightarrow Node$	$Node$ は $Line$ の向きになぞって最初の交点.
$(Line, Circle) \longrightarrow Node$	$Node$ は 上でない方の交点.
$(Line1, Line2) \longrightarrow Node$	$Node$ は $Line1$,$Line2$ の交点.
$(Node1, Node2) \longrightarrow Line$	$Line$ $Node1$ から $Node2$ に向いた直線.
$(Node1, Node2) \longrightarrow Circle$	$Circle$' の中心 $Node1$, 半径は $Node1$,$Node2$ の距離.

表 1: 依存関係とその意味

タ作図システムの実装からはじめて不動点作用素の定義，不動点図形のデザインの順に極限コンパスの実装をみていくことにする.

2 依存木

依存木はコンピュータ作図の実装で工程の保持に用いられる典型的なデータ構造である（[1] など）. ここでは後のために実際に LIMIT で用いた依存木とその記法を導入しておく.

作図作業の各ステップは図形や点の間の依存関係をつくりだす. たとえば，円 C_1,C_2 からそれらの交点 N をつくった場合これらの間の依存関係を

$$(C1, C2) \longrightarrow N$$

のように表す. 正確には，この表式は C_1 を左に C_2 を右に見た時二つの交点のうち下側のものが N であることを意味するものとする. 同様に N が上側であるときには

$$(C2, C1) \longrightarrow N$$

と書く. われわれは直線と円しか考えないので通常の作図で生成される依存関係 DEP はつぎのようなものに限る.

$$\begin{aligned} Fig &::= Line|Circle \\ DEP &::= \longrightarrow Node \\ &\quad | \quad (Fig1, Fig2) \longrightarrow Node \\ &\quad | \quad (Node1, Node2) \longrightarrow Fig, \end{aligned}$$

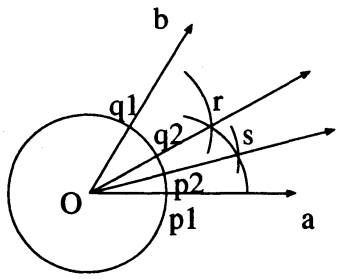
ここで $Node$, $Line$, $Circle$ はそれぞれ交点,（有向）直線，円である. すべての DEP の意味は表 2 に示す.

作図作業の**依存木**はその作業の間に生成された依存関係すべての集合で定義される.

図1（左）は角の4等分の作図を示したものである．作図工程の依存木は図式的に図1（右）に示した．図1（右）で楕円1とそれに入ってくる2本の矢印の始点の楕円2，楕円3は依存関係

$$(\text{楕円 } 2, \text{楕円 } 3) \longrightarrow \text{楕円 } 1$$

を表す．楕円2, 楕円3の順は楕円1に入ってくる矢印のやじりの順（左から右）．入ってくる矢印のない楕円は自由点を表している．



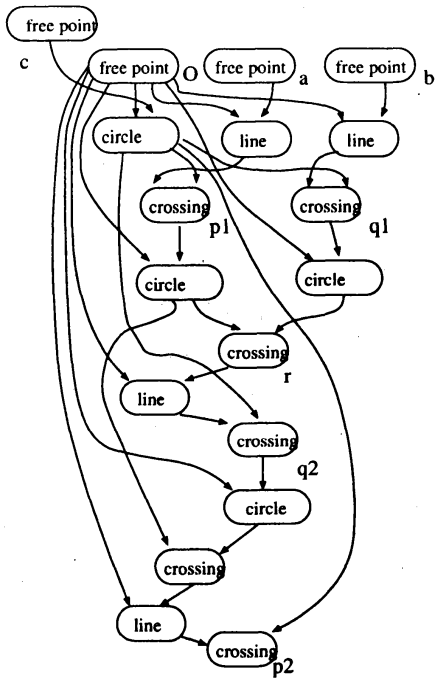


図 1: 角の4等分の作図（左）と依存木（右）．

3 不動点作用素

依存木に作用する不動点作用素を考えれば無限木を表現することができる．抽象的には**不動点作用素**Fixは次の等式で特徴づけられる．

$$(\forall f) Fix(f) = f(Fix(f))$$

依存木の場合の対応する等式を図式的に描いたものが図2である．図2でfやgは依存木の部分，細い矢印は依存関係の矢印の束である．Fixでは繰り返すべき工程fの開始点達と終点達を指定する．工程fの終点達は次に繰り返されるfの開始点達でなければならないので開始点の数は終点の数と一致しこの数は不動点作用素のarityと呼ばれる．不動点作用素はarity kを明示してFix/kと書かれることもある．gは工程内で生成される図形ないし点（以下オブジェクト）ではなく工程で参照される（すなわち依存関係の左辺にのみ現れる）だけのオブジェクト達を意味する．fとgはfの開始点達と終点達を指定すれば一意に定まることに注意せよ．このことにより不動点作用素の図形化が可能になる．図形化に先だって，不動点作用素による繰り返し指示を一つの作業として依存木に加えられるようにしておこう．不動点作用素がつくる依存関係を

$$(s_1, t_1, \ldots, s_k, t_k) \longrightarrow Fix/k.$$

のように書くことにする．ここで$\{s_i\}, \{t_i\}$は開始点の集合と終点の集合で同じ添字は対応を示すものとする．

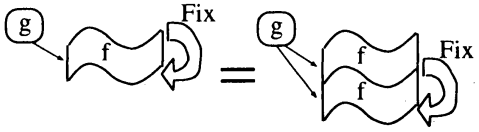


図 2: 依存木上の不動点作用素 Fix を特徴づける等式.

例

角の 4 等分の作図（図 1（左））で $\angle p_1Oq_1$ からその 4 分の 1 の角度 $\angle p_2Oq_2$ を作図したのと同じ手順でその $\angle p_2Oq_2$ からさらにその 4 分の 1 を作り $\cdots$ という風に繰り返す場合を考えよう．このばあい，開始点は p_1, q_1 であり終点は p_2, q_2 であるので

$$(p1, p2, q1, q2) \longrightarrow Fix/2$$

を付け加えればよい．この依存関係およびこれにより繰り返される部分を図 3 に示す．

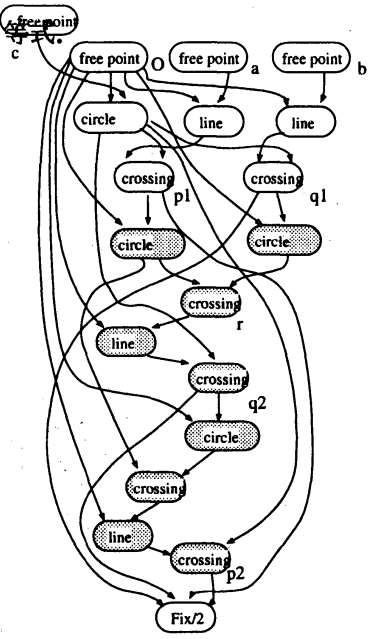


図 3: 角の 4 等分の作図に $Fix/2$ を加えた依存木．繰り返される工程部分を影つきで示す．

4 不動点図形

不動点作用を用いるにあたっては開始点のリスト（順不同）と対応する順の終点のリストを指定すれば十分であることがわかったのでこれらの点を頂点とし頂点間の対応関係がわかる図形をデザインすれば不動点作用素の図形化は完了する．これに極限を指し示す頂点を追加したものを**不動点図形**と呼ぶ．LIMIT では工程の終点達のうち第一のものが繰り返しのたびに生成されてできる点列にたいしてのみ極限をもしあれば生成する．このため第一のものであることが図形中で表される．

LIMIT で用いた不動点図形 $Fix/3$ を図 4 に示す．図 4 で波線でつながった点は開始点達，それらを始点する矢印は対応する終点をやじりに持つ．これらの矢印のうち丸がついたものが第一の開始点／終点である．丸印つき矢印のやじりから更にのびた二重やじりの矢印はその極限を指し示す．

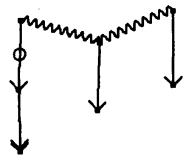


図 4: 不動点図形 $Fix/3$.

LIMIT では極限はまた通常の点と同様にべつの工程の起点になる．このばあい依存関係 DEP としては Fix/k が $Node$ として取り扱われ，左辺に現れることになる．

図 3 の例でさらに原点 O と極限を結ぶ直線 l を引いた場合，$Fix/3$ 図形を F として

$$(O, F) \longrightarrow l$$

が依存木に追加される．

これで繰り返しとその極限のモデル化と図形化が完了した．つぎにこれらが実際のエディタ LIMIT でどのように使用されるかを見てみよう．

5 使用例

角の 3 等分

LIMIT において，図 1 の工程が終了した状況で前節図 3 に表されるような繰り返しを指示してみよう．ユーザは Shift キーを押しながら，p_1 から p_2 にドラッグし引き続き Shift キーを押したままで q_1 から q_2 にドラッグすることで根元が波線でつながった 2 本の矢印を描画する．描画完了と同時に指示された繰り返し工程が起動する．このばあいは収束して極限が生成され，2 重矢印が現れる．実行後の画面を図 5 に示す．極限だけが欲しい場合は当該不動点図形を選択しそれに由来する繰り返し工程を隠すこともできる．(図 5 左.)

図 5: 角の 3 等分．繰り返し部分非表示（左）と表示（右）．

自然対数の底（Napier 数）

図 6 に Napier 数 e の作図の概略を依存木で示す．LIMIT における実際の作図を図 7 に示す．この図の場合，$Fix/4$ を含めて 51 ステップの工程のうち繰り返し部分は 29．繰り返し 9 回で $1.7182821 \approx e - 1$

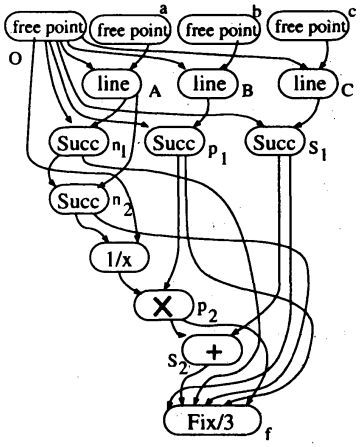


図 6: Napier 数 e の依存木の概略．$Succ$, $+$, $\times$,and $1/x$ はそれぞれ後継（$+1$），和, 積, 逆数である．詳細略.

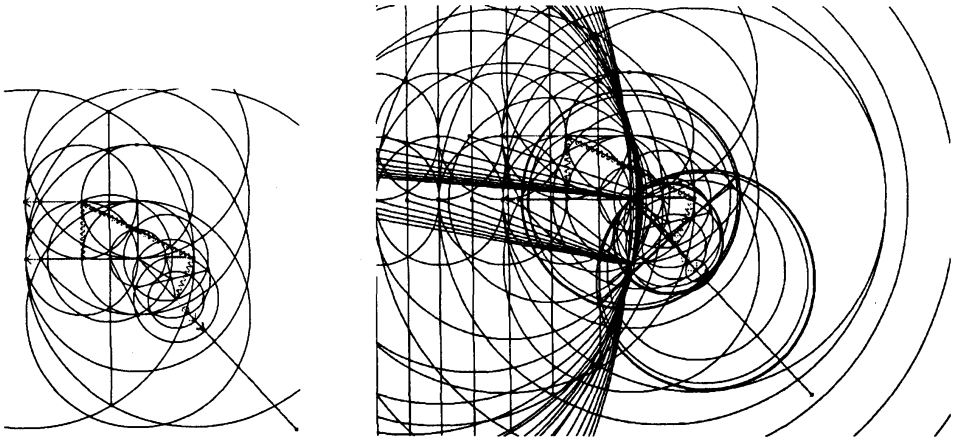


図 7: Napier 数．繰り返し部分非表示（左）と表示（右）．左で原点 O と二重矢印の先との距離が $e-1$.

6 実装

開始点のリスト $S=[s_1,\cdots,s_k]$ と終点のリスト $T=[t_1,\cdots,t_k]$ から繰り返し部分を集める手順を述べておく．依存木を図 1 のような図で表した時,（依存木の）節点の間の順序について，二つの節点 A,B が矢印で $A\longrightarrow B$ のように結ばれているとき，または A から B 矢印をたぐっていけるときに $A>B$ と定義する．S,T に関する仮定として $s_i>t_i(i=1,\cdots,k)$ とする．$s_i\leq t_j$ なる $i,j(i\neq j)$ があってもよい．節点全体の集合を X, 節点の集合 A にたいして次の記号を用いる．

$$Down(A)=\{x\in X|\exists(a\in A)a>x\}\backslash A$$

$$Up(A)=\{x\in X|\exists(a\in A)a<x\}\backslash A$$

いま，仮定を満たす S,T に対して，

$$Mid(S,T)=Down(S)\cap Up(T)$$

と定義する．一回の繰り返しでコピーすべき節点の集合 Gen は,

$$Gen=(Mid(S,T)\cup T)\backslash S$$

で表される．Gen のすべての要素についてその直接の親（入ってくる矢印で直接つながった相手）が S の要素であるかコピー済みの節点であるか，外部参照の節点であるかのいずれかの場合にあてはまるものからコピーしていけばよい．

7 議論

本稿では作図における工程の無限回の繰り返しを不動点作用素でモデル化し，それを図形化して単純な描画操作で指定できる GUI を設計した．

今回は簡単のため無限回の繰り返しのみで実装を行なったが，停止判定のモデル化（図形化）を追加すれば，有限回の繰り返し指示も可能であると予想される．

今回は論理との対応関係は綿密には行なわなかった．通常の型つきプログラミング言語（たとえば ML など）では不動点作用素の導入は型の体系に矛盾を生じることが良く知られている．作図プログラミングの場合，論理に相当するものは幾何学であるので，このことは極限が存在しない（幾何学意味を持たない）ことを意味する．帰納的定義を導入した作図の（幾何学的）意味論の詳細は別の機会に譲る．

計算論的な体系で幾何学を記述することは，検証や合成など計算機による機械的な加工技術を利用できるようにするのが狙いであった．今後描画エディタやグラフィクスへの応用を検討してゆく予定である．

参 考 文 献

[1] Jackiw, R.N. and Finzer, W.F.: The Geometer's SketchPad: Programming by Geometry, it Watch What I Do: Programming by Demonstration (Cypher, A., ed.), MIT Press, 1993, 292–307.

[2] 宮本健司，原田康徳，尾内理紀夫: 拡張できるGUIアプリケーションにおける大規模な作業の予測, コンピュータソフトウエア, **15**(5), 1998, 49–64.