コンピュータと教育 4－10
（１９８９．２．２５）

幾何学図形の入力とその内部表現変換のインタフェイスに関する研究

松田　昇*　　°永島　聡**　　岡本　敏雄**
*金沢工業大学　　**東京学芸大学　　**東京学芸大学

幾何学図形の作図・修正が可能な図形入力インタフェイスにより作図された幾何学図形の構造を，意味ネットワーク表現に変換するシステムを構築した．ユーザは，マウス等を利用して幾何学図形を作図する．その際，線分，多角形，垂線等の基本幾何図形の作図支援機能を利用することができる．次にシステムは，作図結果を幾何学図形を表現するアトム知識に基づく意味ネットワーク表現に変換する．

筆者らは本システムを，既に開発した幾何論証知的ＣＡＩ，ＧＥＯＭＥＸⅡの教材作成に応用した．これにより，入力された課題図から，ＧＥＯＭＥＸの教材構造知識ベースをシステムに生成させることが可能となった．

"The study on the interface for drawing the geometry figures and the transformation to inner representation."

by Noboru MATSUDA (The center for CAI, Kanazawa Institute of Technology, 7-1, Ohgigaoka, Nonoichi-machi, Ishikawa-gun, Ishikawa-ken, 921, Japan), Satoshi NAGASHIMA and Toshio OKAMOTO (Tokyo Gakugei Univ.)

We have developed the interface which facilitates to draw the geometric figures. The user can draw the geometric figures using the mouse, through the flexible function of the system to draw figures (i.e. triangles, quadrangles, bidectors, etc). The system transforms thoes figures into a semantic network which represents the structure of the figures. The atomic knowledge is utilized to buid the structure of the geometric figures that becoms the knowledge base for GEOMEX-Ⅱ.

1．はじめに

コンピュータ技術の進歩にともない，コンピュータを利用する際の使いやすさ，即ちインタフェイスの機能が重要視されるようになってきた．例えば，現在急激に普及しつつあるワークステーションでは，マウスとビットマップディスプレイによるマルチウィンドウ・システムが利用され，注目を集めている．一般に，インタフェイスの役割は，コンピュータの内部処理のプロセスをユーザに理解しやすい形で表示すること，及び操作の簡易化にあると言われている[1)]．

筆者らは，幾何論証の学習世界における知的ＣＡＩとして，ＧＥＯＭＥＸⅡ[2),3)]を開発した．そこでは，学習者の証明計画を認識するために，教材構造知識ベースを利用している．しかし，個々の課題に対し，教材構造知識ベースを作成することが困難であり，ＧＥＯＭＥＸ利用上の問題点であった．そこで，前述した動向を踏まえ，教材作成者（教師）の課題入力の支援を目的として，幾何学図形入力インタフェイスを作成し，さらに作図された幾何図形を，幾何学図形を表現するアトム知識を用いた意味ネットワーク表現に変換するシステムを構築した．

2．システム構成

本システムは，幾何学図形入力モジュールと内部表現変換モジュールから構成される．図１にシステム構成を示す．

ＧＥＯＭＥＸにおける教材構造知識ベースを作成する過程は，以下の通りである．先ず，幾何学図形入力モジュールを利用し，課題となる幾何図形を作図する．内部表現変換モジュールは，作図された図形をアトム知識を用いた意味ネットワーク[2)]に変換する．その後，ＧＥＯＭＥＸの解法エキスパートは，解法知識を適用して教材構造知識ベースを生成する．

本稿では以下，３章で幾何学図形入力モジュール，４章で内部表現変換モジュールについて各々詳述する．

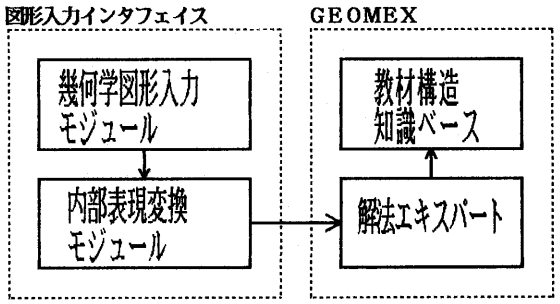


図１　システム構成

3．幾何学図形入力モジュール

幾何学図形入力モジュールは，マウス等のポインティングデバイスを用いることにより，幾何学図形の作図を可能にする．また課題文，課題の仮定・結論を入力することができる．

本モジュールを構成する主なサブモジュールは，①線分データ作成部，②頂点名入力部，③課題文入力部，④ファイル出力部である．図２に本モジュールの構成を示す．

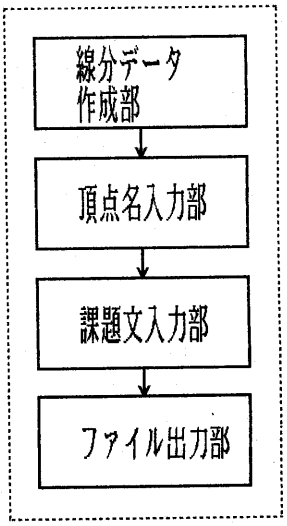


図２　幾何学図形入力部の構成

3.1　線分データ作成部

本モジュールは，線分データ入力部，線分データ処理部からなる．現時点では，ＧＥＯＭＥＸの対象学習世界に依存して，線分を利用した図形のみが作図可能である．本モジュールは，以下の機能を有する．

①基本幾何学図形の作図機能

線分，三角形，四角形，平行線，垂線等の幾何図形をメニュー方式で選択し，作図することができる．図３に現在利用可能なメニューを示す．

②ポインティング支援機能

線分の端点，中点といった，幾何学図形を作図する

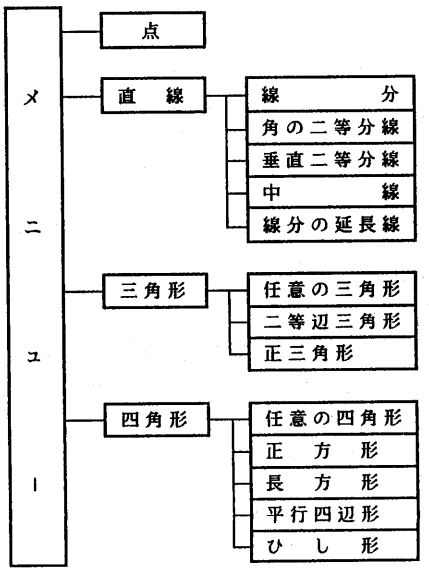


図３　作図可能な基本幾何図形

際に重要となるポイントを指定することができる．この機能により，例えば2つの線分の中点を通る直線を作図することが可能となる．

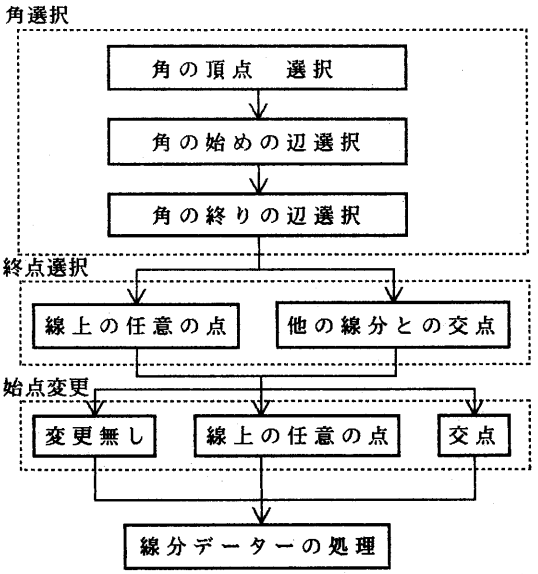


図4　角の二等分線の作図

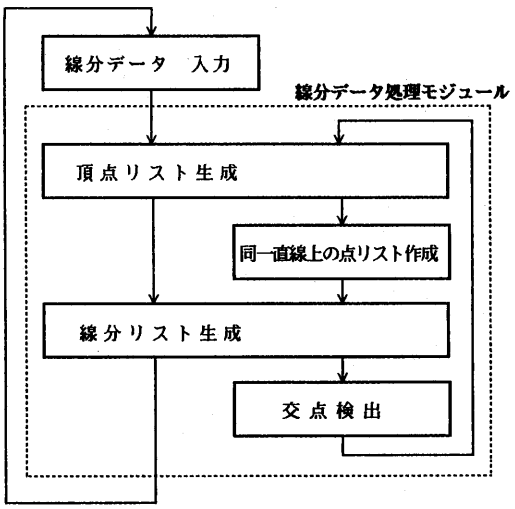


図5　線分データ処理部の構成

③線分の削除・変更機能

作図された図形の一部修正及び変更が可能である．

3.1.1　線分データの入力

マウス等を利用して図形を作図する形式で，図形を構成する各線分のデータを入力する．その際システムは，作図された線分に仮の頂点名を付け，各線分を区別する．　幾何学図形の作図は，頂点または線分を指定することで行う．例えば，線分では2つの頂点を指定し，角の二等分線では，1つの頂点と2つの線分を指定する．その際，頂点は以下の方法で選択することが可能である．

①任意の点

②線分の端点または2つの線分の交点

③線分のn等分点（中点，3等分点など）

④線分上の任意の点

上記の機能を組み合わせることにより，様々な幾何学図形を作図することができる．

図4に角の二等分線を作図する方法を示す．ユーザは，まず目的である角の頂点を指定し，次に角の2辺（線分）を指定する．システムは，指定された角の二等分線を画面に表示する．ユーザは必要であれば，その長さを変えることができる．そのとき，上述した頂点の選択機能を利用し，例えば特定の線分との交点までといった指定が可能である．

またシステムは，二等辺三角形，長方形といった特定の三角形，四角形において，指定された頂点に基づき他の頂点を相対的に決定する機能を有する．

3.1.2　線分データの処理

本モジュールは，前述した方法で入力された線分データを基に，①重複チェック，②頂点リスト作成，③線分リスト作成，④交点検出，⑤同一直線上の頂点リスト作成を行う．図5に線分データ処理部の構成を示す．

①重複チェック

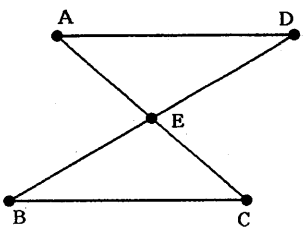
後述する線分リストを基に，重複した線分の入力を禁止する．重複チェックは，以下に述べる各処理において適宜行われる．

②頂点リスト作成

新たに入力された頂点を頂点リストに追加する．各頂点は，次の形式で表現される．

（頂点名　X座標　Y座標

頂点名X座標　頂点名Y座標）




```
'((B C) (B D) (D A) (A C) (A E) (C E) (B E) (D E))
'((A E C) (D E B))
'((B  29 173  24 175) (C 164 173 159 175) (D 166  36 159  19)
  (A  34  36  24  19) (E  98 104 109  96))

(put '*problem* 1 '(¦  ２つの線分ＡＣとＢＤ¦
                   ¦が、各々の中点Ｅで交わ¦
                   ¦っているとき、線分ＡＤと¦
                   ¦ＢＣは平行であることを¦
                   ¦証明しなさい．¦))

(PUT '*PROBLEM* '仮定 '((AE = CE) (BE = DE)))
(PUT '*PROBLEM* '結論 '(AD !! BC))
```

図6　作図例

頂点名のX，Y座標は，画面に頂点名を書き出す位置を表す．

③線分リスト作成

新たに入力された線分を，線分リストに追加する．各線分は，両端点のリストとして，次の形式で表現される．

（頂点名　頂点名）

線分リストは，次の形式で表現される．

（（頂点名　頂点名）（頂点名　頂点名）…）

④交点検出

新たに入力された線分データとの交わりを，線分リストの先頭から順に調べる．入力された線分と交わる線分が見いだされた場合には，交点を頂点リストに追加し，同一直線上の頂点リストを更新する．

⑤同一直線上の頂点リスト作成

同一直線上にある頂点のリストを作成する．このリストは，線分の端点が他の線分上にある場合，及び線分が他の線分と交わる場合に更新される．その際，線分リストも新たに追加される．例えば，既存の線分に交わる線分を描いた場合，同一直線上の頂点リストの更新とともに，新たに４つの線分がリストに追加される．

3.2　頂点名入力部

システムは最初，入力された線分データに仮の頂点名を付ける．ユーザはここで，各頂点に名前を付けることが可能である．また頂点名の表示位置も指定することができる．その際システムは，頂点名の重複チェックを行う．

3.3　課題文入力部

システム内の簡易ワープロを利用して，課題文及び課題の仮定・結論等を入力することが可能である．入力された課題文等は，テキストとして保存される．

3.4　ファイル出力部

入力された頂点・線分データ，および課題文をＬＩＳＰのリスト形式で，テキストファイルに出力する．その際，頂点リスト，線分リスト，同一直線上の頂点リスト，課題文，仮定，結論の順に出力する．その際，同一直線上の頂点リストは，各頂点の座標によってソートする．図６に作図例及びそのファイル出力結果を示す．

4．内部表現変換モジュール

内部表現変換モジュールは，図形入力モジュールにより作図された課題図を，幾何学図形を表現するアトム知識を用いた意味ネットワーク表現に変換する．アトム知識とは，幾何学図形を表現する最小限度のノードである．現時点では頂点，線分，半直線，直線，角を表現するアトム知識がある．特定の幾何学図形は，アトム知識をノードとし，各アトム知識の関係をリンクとする意味ネットワークで表現される．

本章では，頂点・線分データを意味ネットワーク表現に変換する方法について述べる．

システムは，前述した頂点リストから頂点ノード（アトム知識）を生成し，線分リストから線分ノードを生成する．また同一直線上の頂点リストから半直線ノード及び直線ノードを生成する．最後に頂点リスト及び生成されたノードから角ノードを生成する．

4.1　頂点ノードの生成

頂点を表現するアトム知識は，前述した頂点リストから生成される．図７に頂点を表現するアトム知識及びリンクの意味を示す．図８に頂点を表現するアトム知識を生成するプログラムを示す．図の中で，(gen-pnt c)は，ノード番号ｃの頂点ノードを生成する．

4.2　線分ノードの生成

線分を表現するアトム知識は，前述した線分リストから生成される．図９に線分を表現するアトム知識及びリンクの意味を示す．図１０に線分を表現するアトム知識を生成するプログラムを示す．図の中で，(get-pnt P) は，前述した方法で生成された頂点ノードの中から，頂点Ｐに該当するものを取り出す関数である．

4.3　半直線ノードの生成

半直線を表現するアトム知識は，既に生成された頂点ノード及び線分ノードから生成される．図１１に半直線を表現するアトム知識及びリンクの意味を示す．

半直線ノード生成のアルゴリズムは，次の通りである．

```
repeat
  次の頂点ノードPを読む
  Pを通る線分ノードを Segs に代入する
  repeat
    次の線分ノードSegを読む
    if PはSegの端点ではない do
        if Pを端点とし，Segを含む半直線Rayが
           存在する
           do  Segのonthe属性にRayを代入
      else do  新たな半直線ノードを生成する
  until  Segsが空でない
until  頂点ノードが空でない
```

図１２に半直線を表現するアトム知識を生成するプログラムを示す．図の中で，(is-endp-of-seg P)は，線分ノードの中から，endp属性に頂点Ｐをもつものを取り出す関数である．(mid-pnt-p P Seg) は，頂点Ｐが線分Seg 上の点か否かを評価する述語である．

また(owned-p Seg P) は，頂点Ｐを端点とし，線分Segを含む半直線を取り出す関数である．

4.4　直線ノードの生成

直線を表現するアトム知識は，同一直線上の頂点リストから生成される．図１３に直線を表現するアトム知識及びリンクの意味を示す．図１４に直線を表現するアトム知識を生成するプログラムを示す．図の中で，(search-part Pntl Pnts) は，Pntlを中心とする逆方向に伸びる２つの半直線を取り出す関数である．

４．５　角ノードの生成

角を表現するアトム知識は，頂点ノード，線分ノード，半直線ノードから生成される．図１５に角を表現するアトム知識及びリンクの意味を示す．

角ノードを生成するアルゴリズムは，次の通りである．

```
repeat
  次の頂点ノードPを読む
  Pを端点とする線分ノードをSegsに代入する
  repeat
    次の線分ノードSeg2を読む
    頂点がP，辺がSeg2であるような新しい角ノード
    を生成する
  until Segsが空
until  頂点ノードが空
repeat
  次の角ノードを読む
  ２辺の座標を基に角の方向を決定する
  角のコード（名前）をつける
until 角ノードが空
```

図１６に角を表現するアトム知識を生成するプログラムを示す．図の中で，(is-apex-of-ang P)は，頂点Ｐを端点とする線分を２つ組にする．(sort-by-slope Segpar Apex) は頂点Apexを中心として，角の２つの辺Segparの相対的な座標を基に，角の方向を決定する関数である．(put-side Apex Segs)は，角Apexの２つの辺Segsをfside属性，eside属性に代入する関数である．また(give-angcode Apex)は，角Apexのコードを label属性に代入する関数である．

図１７に図６の図形を意味ネットワーク表現に変換した結果の一部を示す．意味ネットワークは，ＬＩＳＰの連想リストとして表現されている．

５．おわりに

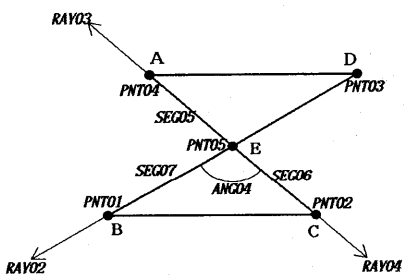
幾何学図形の作図が可能なインタフェイスにより入力された幾何図形を意味ネットワーク表現に変換するシステムについて述べた．

幾何学図形作図システムは，メニュー方式により基本幾何図形を選択・描画する．その際，様々な作図支援機能をもたせることにより，効率よく図形を描くことが可能となった．現時点では，線分で構成された図形しか扱っていないが，今後より一般的な図形の作図を可能にする必要があろう．

システムは，作図データを幾何学図形を表現するアトム知識を用いた意味ネットワーク表現に変換する．前述した一般的な図形を処理するためには，アトム知識を拡張し，より一般的な図形を表現するプリミティブを用意する必要があろう．

６．参考文献

1) 溝口，インタフェイスの科学，淵監修，インタフェイスの科学，pp.1-22，共立出版社 (1987)．
2) 岡本ほか，幾何論証の学習世界における知的ＣＡＩの構成について，情報処理学会論文誌，Vol.29，No.3，pp.311-324 (1988)．
3) 岡本ほか，幾何論証の学習世界における知的ＣＡＩの構成について（６），信学技報，ET88-1，PP.57-64 (1988)．




```
*ANG04* ((LABEL <BEC) (ESIDE . *RAY04*) (FSIDE . *RAY02*)
         (SEGPAR *SEG06* *SEG07*) (APEX . *PNT05*) (ISA . ANG))

*RAY04* ((HAS *SEG06*) (ENDP . *PNT05*) (ISA . RAY))
*RAY03* ((HAS *SEG05*) (ENDP . *PNT05*) (ISA . RAY))
*RAY02* ((HAS *SEG07*) (ENDP . *PNT05*) (ISA . RAY))

*SEG07* ((CONTHE *RAY12* *RAY08* *RAY02*) (ENDP *PNT01* *PNT05*) (ISA . SEG))
*SEG06* ((CONTHE *RAY10* *RAY06* *RAY04*) (ENDP *PNT02* *PNT05*) (ISA . SEG))
*SEG05* ((CONTHE *RAY10* *RAY06* *RAY03*) (ENDP *PNT04* *PNT05*) (ISA . SEG))

*LIN01* ((PART *RAY03* *RAY04*) (JCT . *PNT05*) (ISA . LIN))

*PNT05* ((POS 109 96) (XY 98 104) (LABEL . E) (ISA . PNT))
*PNT04* ((POS 24 19) (XY 34 36) (LABEL . A) (ISA . PNT))
*PNT03* ((POS 159 19) (XY 166 36) (LABEL . D) (ISA . PNT))
*PNT02* ((POS 159 175) (XY 164 173) (LABEL . C) (ISA . PNT))
*PNT01* ((POS 24 175) (XY 29 173) (LABEL . B) (ISA . PNT))
```

図１７　幾何学図形の意味ネットワーク表現

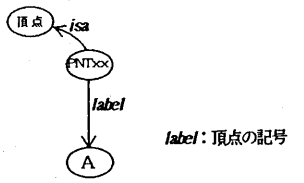


図７　頂点を表現するアトム知識

```
(defun defpoint (PntList          ; 点定義
                 c NewPnt)
  (setq c 0)
  (mapc '(lambda (Pnt)
    (setq NewPnt (gen-pnt (incq c)))
    (put  NewPnt 'label (car Pnt))
    (put  NewPnt 'xy  (butlast (cdr Pnt) 2))
    (put  NewPnt 'pos (nthcdr 2 (cdr Pnt))))
  PntList))
```

図８　頂点ノード生成プログラム

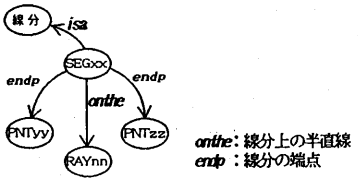


図９　線分を表現するアトム知識

```
(defun defsegment (SegList          ; 線分定義
                   c NewSeg)
  (setq c 0)
  (mapc '(lambda (Pnts)
    (setq NewSeg (gen-seg (incq c)))
    (put  NewSeg 'endp (mapcar 'get-pnt Pnts)))
  SegList))
```

図１０　線分ノード生成プログラム

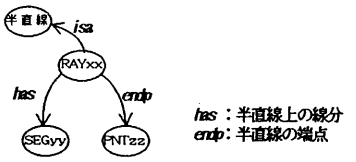


図１１　半直線を表現するアトム知識

```
(defun defray (Pnts                 ; 半直線定義
               c NewRay Segs)
  (setq c 0)
  (mapc '(lambda (Pnt)
    (setq Segs (is-endp-of-seg Pnt))
    (mapc '(lambda (Seg
                    Owner)
      ((mid-pnt-p Pnt Seg))
      ((setq Owner (owned-p Seg Pnt))
        (put Seg 'onthe  (cons Owner (onthe Seg))) )
      (setq NewRay (gen-ray (incq c)))
      (put  NewRay 'endp Pnt)
      (put  NewRay 'has   (list Seg))
      (put  Seg    'onthe (cons NewRay (onthe Seg))) )
    Segs) )
  Pnts))
```

図１２　半直線ノード生成プログラム

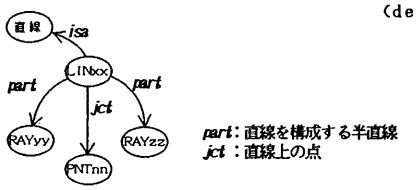


図１３　直線を表現するアトム知識

```
(defun defline (ColpList
                c)
  (setq c 0)
  (mapc '(lambda (Pnts)
    (mapc '(lambda (Pnt1
                    NewLine)
      (setq NewLine (gen-lin (incq c)))
      (put  NewLine 'jct  (code2con Pnt1))
      (put  NewLine 'part (search-part Pnt1 Pnts)) )
    (butlast (cdr Pnts)) ) )
  ColpList) )
```

図１４　直線ノード生成プログラム

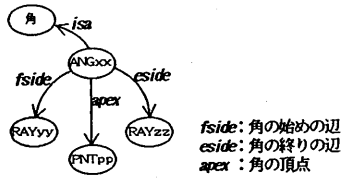


図１５　角を表現するアトム知識

```
(defun defangle (Pnts
                 c NewAang Segs)
  (setq c 0)
  (mapc '(lambda (Pnt)
    (setq Segs (is-apex-of-ang Pnt))
    (mapc '(lambda (Seg2)
      (setq NewAng (gen-ang (incq c)))
      (put  NewAng 'apex Pnt)
      (put  NewAng 'segpar Seg2) )
    Segs) )
  Pnts)
  (mapc '(lambda (Ang
                  seg2)
    (setq seg2 (sort-by-slope (segpar Ang) (apex Ang)))
    (put-side Ang seg2)
    (give-angcode Ang) )
  *Angs*) )
```

図１６　角ノード生成プログラム