帰納論理システムを用いたフロアプランの自動生成*

2J－3

溝口文雄†　清水健一　大和田勇人
東京理科大学　理工学部†

1　はじめに

フロアプランニング問題は，与えられた幾つかの制約条件を満足するように部屋の大きさや配置を決定する問題である．与えられる条件としては，部屋は敷地内に必ずあるとか部屋同士は重ならないなどという必然制約や，台所は広くするなどというユーザー制約がある．しかし，間取りの可能性は無数に存在するため，条件を満たす間取りを効率良く求める手法が必要とされる．

そこでこの問題に対し，事例から知識を獲得する一手法である帰納学習を適用する．帰納学習は背景知識を有効利用でき表現力があるという長所を持つ．学習により実際に設計された図面において成り立つ一般的関係が獲得できれば，学習された知識をプランニングを行なう際の前提条件として活用でき，また，制約条件，初期条件と学習された規則から，ユーザー制約が反映される以前の基本的なフロアプランを自動生成できると考えられる．

本研究では，学習システムとしてC4.5[2]，Progol[1]を用い，フロアプランのデータからの学習と，それを用いたフロアプランの自動生成の結果について報告する．

2　データ

本研究では，実際に建築された100個の図面データを対象とし，そのうち1階の図面(総部屋数1061)だけから学習を行なった．

フロアプランのデータは以下のようになっている．

```
図面，部屋，クラス　　，座標値
1    ,1    ,living_dining,[12,18,12,42,48,42,48,18]
1    ,2    ,kitchen      ,[0,18,0,42,12,42,12,18]
     :     :     :              :
```

この元のデータから，部屋の面積や東西南北の隣接関係等の情報を獲得する．

```
room(1,1,living_dining).space(1,1,864).
east(1,1,kitchen). west(1,1,wall).
        :
```

3　学習システム

フロアプランのデータは，隣接関係のような記号データと面積のような数値データからなるので，記号データと数値データを同時に扱える機構が必要となる．このような学習システムにC4.5とPrgolがある．

C4.5は数値データを扱うのに特に適しており，また，多重なクラスの分類が得意である．ただし，属性-値の形式で記述できないものは学習できない，対象間の関係を学習できない，などといった学習の限界がある．

Progolは組み込み述語や演算子などの利用により確定節や数値データも扱えるシステムである．論理プログラムの特徴から，複雑な関係の記述・学習が可能である．

4　学習に基づくプランニング

図面における部屋の配置や隣接関係に関して必ず満たされる規則があれば，プランニングを行なう際の重要な手がかりとなる．そのような規則を用いて，以下の手順にてプランニングを行なうことができると考えられる．

1. 初期条件(敷地面積，玄関位置)が与えられる．
2. 初期条件から決定される部屋の配置を行なう．
3. 部屋同士の関係から残りの配置を行なう．
4. 大まかな配置が決定する．

そこで，初期条件から決定される基本的な部屋の配置に関する規則や，隣接関係などの部屋間の関係に関する規則の学習を行なう．

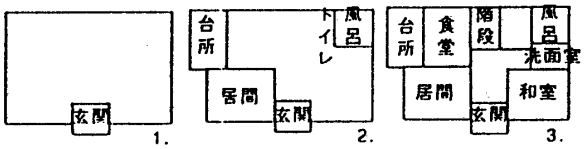

図 1: プランニング過程の概略図

4.1　初期条件から決定される基本的な関係

背景知識として，玄関の位置，敷地の面積を与え，四隅に配置される部屋，壁に接する部屋，玄関に接する部屋などをそれぞれターゲット述語として，Progolで学習を行なう．その結果，以下のような規則が学習される．

```
adj_wall(A,bath,north) :- doorposition(A,east). [56]
adj_wall(A,jproom,south) :- doorposition(A,north),
  fspace(A,S), S<60, S<70. [34]
adj_wall(A,living,south) :-
  doorposition(A,north). [16]
adj_wall(A,kitchen,north) :- doorposition(A,east),
  doorposition(A,south). [11]
```

*Floor-Planning using Inductive Logig System
†Fumio Mizoguchi, Kenichi Shimizu, Hayato Ohwada
‡Faculty of Sci. and Tech., Science Univ. of Tokyo

```
    :
corner(A,kitchen,west_north) :-
  doorposition(A,east), doorposition(A,south),
  fspace(A,S), S<50, S<60. [6]
    :
```

規則の右側に示した数字は，包含する正事例の数である．学習は負事例を満たさないように行なわれたので，正事例の数が多いほど重要な規則であると考えられる．なお，玄関の位置と，敷地面積に関する事例の数の分布は表1,2のようになっている．

ある部屋が壁に接するという規則よりも家の四隅に配置されるという規則の方が，間取りを決定する上でより重要であると考えられるが，この結果から四隅の配置の規則を直接学習するのは困難であるといえる．

表 1: 玄関の配置と事例数

east	west	south	north
57	30	26	70

表 2: 敷地面積と事例数

~50(m^2)	50~60	60~70	70~80	90~
7	27	52	12	2

4.2 部屋間の関係

部屋間に成り立つ一般的な関係を獲得するため，部屋の種類を識別する規則を学習する．

背景知識として，部屋の面積，部屋の長辺と短辺の長さ，隣接関係(含む否定情報)，玄関からの距離などを与える．このような入力の元，C4.5 と Progol により学習した結果，以下に示すような規則が学習された．

```
room(A,bath) :- adjacent(A,washroom),
    adjacent(A,wall), roomspace(A,S),
    S>100, S<200.
```

両システムによって得られた規則の正確さを比較すると，C4.5 の規則の方が正解率が高かった (表 3参照)．これはより厳密に数値データの分類を行なえたためである．よって実際にプランニングを行なう際には，C4.5 による学習規則を用いる．

表 3: 110 個のテストデータの予測結果

	Correct	Wrong	Empty	Accuracy
Progol	62	20	28	56 %
C4.5	79	31	0	72 %

5 プランニングの実行例

学習された規則を用いて，ユーザー制約を考慮しない場合ののプランニングを行なう．ここでは例として，敷地面積が $58m^2$，玄関の位置が南西である場合を考える．

まず 4.1節で学習された規則のうち，入力条件を満たす規則を抜き出す．

```
adj_wall(A,bath,north) :- doorposition(A,east). [56]
adj_wall(A,living,south) :-
  doorposition(A,east). [15]
adj_wall(A,kitchen,north) :- doorposition(A,east),
  doorposition(A,south). [11]
    :
corner(A,kitchen,west_north) :-
  doorposition(A,east), doorposition(A,south),
  fspace(A,S), S>50, S<60. [6]
corner(A,bath,east_north) :-
  doorposition(A,east), doorposition(A,south),
  fspace(A,S), S>50, S<60. [5]
```

これらの規則を用いて，第一回目の配置を行なう．矛盾が起こる場合は，優先度の低い規則の結果を無視する．そうして得られた結果が以下のものである．

```
corner(A,kitchen,west_north).
corner(A,bath,east_north).
adj_wall(A,living,south).
adj_wall(A,washroom,north).
```

次に，4.2節で学習された以下のような規則を用いて，

```
room(A,dining) :- adjacent(A,kitchen),
  roomspace(A,S), B<576.
room(A,toilet) :- adjacent(A,toilet),
  not adjacent(A,jproom),
  roomspace(A,B) :- B<72, B>36.
```

すでに配置が決定されている部屋の隣の部屋の配置を決定する (図 2参照)．ただし，これらの規則だけからでは，ある部屋はある部屋の隣にあるなどという曖昧な決定しかできず，さらに厳密な配置を行なうためには，未決定のスペース等の考慮が必要である．また，複数の規則によって生じる矛盾する箇所の整合も重要な課題である．

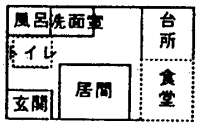

図 2: レイアウト例 (点線部は未確定)

6 おわりに

本研究では，実際の図面から学習された知識を用いて，フロアプランを自動生成することを試みた．学習された規則の中には，常識的な知識と照らし合わせても当然であるものもあり，このような知識は背景知識として利用できる．また，学習された規則に基づき基本的なレイアウトを求めることができた．しかし，レイアウトを行なう上で矛盾する箇所が現れることが多くあり，これを解消することが課題である．また，レイアウトされた図面を表示するインターフェースの設計も急務である．

参考文献

[1] S.Muggleton, Inverse Entailment and Progol, *New Generaton Computing*,Vol.13,1995.

[2] J.Quinlan, C4.5:Programms for Machine Learning, *Morgan Kaufmann Publishers*, 1993.