表構造の一般化に基づくオントロジの獲得

田仲正弘[†] 石田亨[†]

表からの情報抽出に関する従来の研究は，表の認知モデルなど表構造の先見的知識や，対象ドメインの語彙の知識ベースを用いるものであった．しかし幅広く Web から集められた表を対象とする場合には，同じ表構造に対して表ごとに違う解釈をしたり，様々なドメインの表を処理したりする必要がある．本研究では，表構造が表すデータ間の関係の解釈で与え，形式化された表構造とその構造が表すデータの関係との対応を得ることで，表形式データからオントロジを構築する手法を提案する．人手により解釈を与えられた構造を，セルの隣接関係や繰返し構造に注目して自動的に一般化することにより，表中のデータ間の関係を得る．Web 上の価格表・タイムテーブル・統計データなどの表に対し提案手法を適用した結果，少ないコストで表中のデータの関係を記述する多くの RDF ステートメントが得られた．

Ontology Extraction Based on Generalization of Table Structure

Masahiro Tanaka[†] and Toru Ishida[†]

Previous works on information extraction from tables make use of lexical knowledge bases of tables or prior knowledge such as a cognition model of tables. However, we often need to interpret table structures in each table differently and to treat lexicons in various domains for processing a broad range of tables on the Web. The method proposed in this paper extracts an ontology from a table by using relations represented by structures. Once the interpretations of table structures are given by humans, the table structures are automatically generalized to extract relations from the whole table. We defined a formal representation of generalized table structure based on the adjacency of cells and iterative structures. Our experiments showed that the method extracted class-hierarchies, property-value pairs and other various relations from the tables containing price lists, timetables and statistics on the Web.

1. はじめに

Semantic Web の実現には大量のメタデータが必要になる．人手によるアノテーションには大きなコストがかかるため，既存のデータからメタデータを自動的に生成することが望ましい[1]．Web ページからのメタデータ生成については，タグの出現パターンを利用する手法，ラッパを利用する手法[2),3)] などが提案されている．また，ブートストラップ的手法による，非常に大量のデータに対するアノテーションも報告されている[4]．

本研究では，表形式のデータから，オントロジを自動的に獲得する手法を提案する．表はある程度決まった構造を持っており，その構造により表中のデータ間の関係が表される．表からの情報抽出に関する従来の研究には，表を解釈するために，与えられた表を典型的な表構造のいずれかに帰着するもの[5),6)] や，表の認知モデル[7)] に基づいて解析を行うもの[8)] がある．より詳細な関係を獲得するために，対象ドメインの知識ベースを用いて表中のデータの種類を判定するもの[9),10)] もある．

しかし幅広く Web から表を集められた表を対象とする場合には，以下の点が重要になる．

表に応じた構造の解釈 ある表構造がどのような関係を表現するのに用いられるかは，基本的に表によって異なる．そのため，それぞれの表に応じた構造の解釈を用いて，表中のデータ間の関係を獲得する必要がある．

様々なドメインへの適用 Web から広く表を収集すると，多様な内容の表が得られることが多く，ドメインに特化した知識ベースを利用する方法はコストが大きくなる．そのため，表の内容によらず，様々なドメインで利用しやすい手法が必要になる．

そこで本研究では，以下の 3 つのステップからなるアプローチをとる．

† 京都大学大学院情報学研究科社会情報学専攻
Department of Social Informatics, Kyoto University

(1) 表構造の解釈を与える
(2) 解釈を与えた表構造の一般化
(3) 新たな関係の獲得

表構造が表す関係を表ごとに人手で解釈して与えることで，表に応じた表構造の解釈ができる．また，解釈を与えた表構造と似た特徴を持ち，同じ関係を表現するのに用いられている表構造も利用するため，セルの隣接関係や繰返し構造に注目して表構造の一般化を行う．一般化された表構造を用いて，表全体から表中のデータ間の関係を得る．この手法は与えた解釈に基づいて表からデータ間の関係を得るため，ドメインに特化した知識ベースを必要とせず，様々なドメインの表に容易に適用可能である．

以降では，まず 2 章で表構造の観察について述べる．次に 3 章で，表構造の形式化について述べる．4 章では，表からのデータ間の関係の獲得の処理について説明する．5 章で，提案手法の適用の結果得られたオントロジについて評価・考察し，6 章で結論を述べる．

2. 表構造の観察

本章では，表構造を一般化し，形式的表現を与えるために，表構造の特徴とその構造により表される関係についての観察を述べる．

2.1 表構造のセマンティクス

表では，その構造によって表中のデータ間の関係（クラス–インスタンス関係・クラスの階層関係・プロパティ–プロパティ値の組など）が表される．表の観察の結果，ある表構造が表中のデータ間の特定の関係を表すとき，その表構造が表す関係は同じ表の中では多くの場合一定であることが分かった．そこで，表構造が表すデータ間の関係を，その表構造のセマンティクスと呼び，この表構造のセマンティクスに基づいて表中のデータ間の関係を獲得する．

表構造のセマンティクスを説明する例として，**表 1** を取り上げる．表 1 は価格表であり，記載されている製品が PC 部品であることが 1 行目の “PC Component” という記述により示されている．また製品の種別による分類（“Memory”，“Processor”）が記述され，各製品について製品コード，名前，価格が列を分けて記述されている．

表 1 における表構造のセマンティクスは，次のようなものになる．1 行目・3 行目・6 行目にある幅の広いセルには，製品の属するクラスが記述される．幅の広いセルに囲まれた 2 行目の 3 つのセルには，製品のプロパティが記述される．プロパティの下側に位置する，3 つのセルに分かれた行には，各列のプロパティに対応するプロパティ値が記述される．

表 1 PC 部品の価格表
Table 1 A price list of PC components.

PC Component		
ProductID	ProductName	Price
Memory		
M27_512	PC2700 512 MB	$70
M27_256	PC2700 256 MB	$40
Processor		
P4_340	Pentium 4 3.40 E GHz	$260
P4_280	Pentium 4 2.80 A GHz	$140
A64_320	Athlon 64 3200+	$160

表の一部のデータの関係が既知であるときに，その関係を表す構造を一般化することによりこのような表構造のセマンティクスを得ることができれば，この表に記述されている多くのデータの関係が獲得できる．

2.2 表構造の仮定

ある表構造のセマンティクスが明らかならば，表中でその表構造が現れる箇所からは，その箇所に含まれるデータについての関係が得られる．ただし表中のより多くの箇所からセルのデータ間の関係を得るには，表構造をその特徴に基づいて一般化する必要がある．

一般化された表構造の形式的表現を定義するため，表構造の観察の結果から，以下の 3 つの点に関する仮定を置く．

セルの隣接 表 1 のように，表には複数の行や列にまたがるセルが含まれることがある．隣接する 2 つのセルで幅が異なる場合，それらのセルにはふつう異なる種類のデータが記述される．そのため，行や列の構造の特徴は，その行や列に含まれるセルとその周囲のセルとの幅の大小関係に注目して表すことができると仮定する．

同じ行や列内のセル プロパティとプロパティ値のように互いに関連を持つデータが記述されたセルは，同じ行や列に配置されることでその関連が表現されると仮定する．たとえば表 1 では，2 行目のプロパティとその下部に現れるプロパティ値は同じ列に配置されることで対応付けられている．

繰り返し構造 同じ行や列内で同じ特徴を持つセル（もしくは複数のセルから構成されるブロック）が連続して出現する場合には，それらのセル（ブロック）には出現回数によらず似た種類のデータが記述されていると仮定する．表 1 では，4–5 行目，7–9 行目はいずれも 3 つのセルで構成されており，1 つの製品のプロパティ値が記述されている．

以上の仮定に基づいて，表中のデータ間の関係を表

す一般化された表構造の形式的表現を定義する．

3. 表構造の形式化

本章では，2.2 節で述べた表構造に関する仮定に基づき，セルの隣接関係に基づく表構造のモデルを与え，さらに与えたモデルに基づく特定のセルの配列を表す文法を定義する．

3.1 セルの隣接

2.2 節で述べたセルの隣接に関する仮定に基づき，以下に述べるように表構造のモデルを定義する．

まず 1 つのセルに相当するボックスと呼ぶ要素を定義する．さらにボックスに対応するセルの周辺のセルとの辺の重なり方によって，セルの隣接関係を**図 1** のように 1 方向または双方向のボックス間の接続として表す．

図 1 の上部は元の表の隣接する 2 つのセルを示している．図 1 の下部はそれらのセルに対応するボックスとそれらの接続を表している．四角はボックスを表し，上下に並んだ 2 つのボックス間の接続は，対応するセルの隣接関係を表す．

図 1 (a) のように，隣接する 2 つのセルで重なっている辺の長さが同じ場合には，2 つのボックスを双方向に接続して表す．図 1 (b) のように，一方の辺がもう一方の辺に含まれる場合には，短い辺を持つセルに相当するボックスから長い辺を持つセルに，1 方向に接続して表す．図 1 (c) のように，隣接するセルの重なっている辺の一方が，もう一方に含まれない場合には，セルに相当するボックスは接続しない．また，表において隣接する 2 つのセルのどちらが上（左）でどちらが下（右）かということは重要であるため，接続の上下左右の向きも区別する．

ここで表 1 に現れるいくつかの語の関係が，**図 2** の RDF グラフによって与えられていたとする．このとき，表 1 の構造をボックスとその接続により表現し，どのような種類の語が記述されているかというセマンティクスが明らかなボックスにはラベルをつけると，**図 3** のようになる．

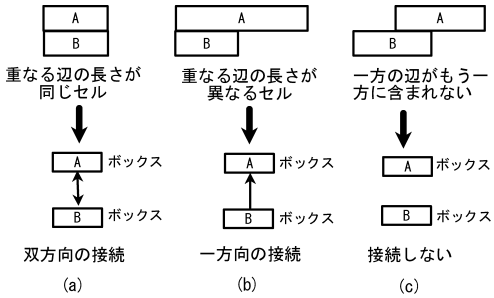

図 1 セルの隣接関係と対応するボックスの接続
Fig. 1 Correspondences between adjacency of cells and a link between boxes.

3.2 記号の定義

前節で与えたセルの隣接関係に基づく表のモデルに基づき，特定のセルの配列を表す文法を定義する．ここで，ある行や列を構成するボックスおよびそれらをつなぐ接続を 1 つの記号と見なす．さらに，連続する行（列）をつなぐ接続も 1 つの記号と見なす．図 3 に現れるボックスとその接続を垂直方向に見た場合，**図 4** に示す記号が含まれている．1 行に相当するボックスとそれらの間の接続を表す b_1，b_2 および各行の間の接続を表す e_1，e_2，e_3 がある．

これらの記号は**図 5** に示す手続き GenerateSymbol により求める．split(r, $table$) は $r+h$ 行（列）目と $r+h+1$ 行（列）目にまたがるセルのないような最小の $h(\geq 0)$ を返す手続きである．また box-symbol(r,

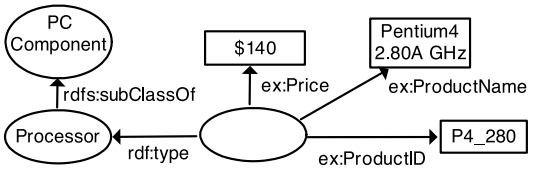

図 2 表 1 中の語の関係を表す RDF グラフ
Fig. 2 An RDF graph describing the relations between data in Table 1.

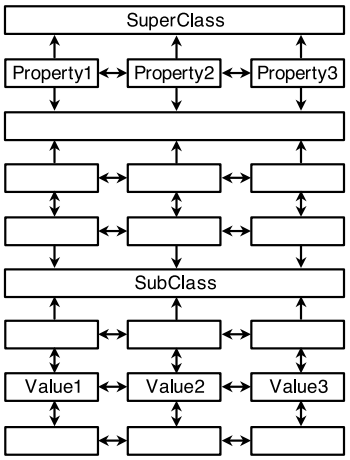

図 3 図 2 の関係に対応する構造
Fig. 3 The structure corresponding to relations shown in Fig. 2.

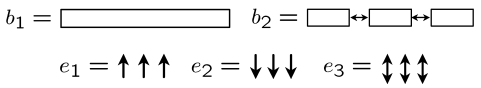

図 4 記号例
Fig. 4 Examples of symbols.

```
GenerateSymbol(table)
  r ← 1, i ← 1, j ← 1
  repeat
    h ← split(r, table)
    if box-symbol(r, h, table) が {b_m} に含まれない
      b_i ← box-symbol(r, h, table), i ← i + 1
    if r + h 行 (列) が最終行 (列)
      return {b_m}, {e_n}
    if edge-symbol(r, table) が {e_n} に含まれない
      e_j ← edge-symbol(r, table), j ← j + 1
    r ← r + h + 1
```

図 5 記号を生成する手続き
Fig. 5 Procedure generating symbols.

h, $table$) は，r〜$r+h$ 行（列）を表すボックスとその接続を返し，edge-symbol(r, $table$) は，r 行（列）目と $r+1$ 行（列）目の間の接続を返す．$\{b_m\}$，$\{e_n\}$ はそれぞれ，すでに得られているボックスと接続を表す記号の集合である．行と列のどちらに基づいて記号を生成するかは，従来研究[5] で提案された手法により，表が縦向きか横向きかを判定することにより決定する．

表 1 の構造を図 4 に示した記号を用いて表現すると，以下のような記号列で表現できる．

$b_1e_1b_2e_2b_1e_1b_2e_3b_2e_3b_2e_2b_1e_1b_2e_3b_2e_3b_2$

3.3 文法の定義

2.2 節で述べたように，同じ行や列内で同じ特徴を持つセルが連続して出現する場合には，それらのセルには出現回数によらず似た種類のデータが記述されていると考える．

この考えに基づき，同じ接続で同じ特徴の行や列が繰り返される場合に，それらのボックスをまとめて表す．そのため，ボックスを用いた表構造の表現に繰返し構造を表す+記号を導入し，同じ接続を持つボックスが連続して出現する構造を表現する．

図 3 の構造を＋記号を用いて表したものが **図 6** となる．＋記号を付した括弧の中の部分が，繰り返し出現する構造を表す．＋記号の横の接続は，括弧の中の構造が繰り返し現れる際のそれらの間の接続を表す．異なるラベルのついたボックスは，異なる種類のデータが記述されていることが分かっている．そこで同じラベルを持つボックスどうしか，または片方がラベルのないボックスの場合にだけ，＋記号を用いてボックスをまとめるものとする．このように，ボックスとその接続，＋記号を用いた表構造の表現を，一般化表構造と呼ぶものとする．

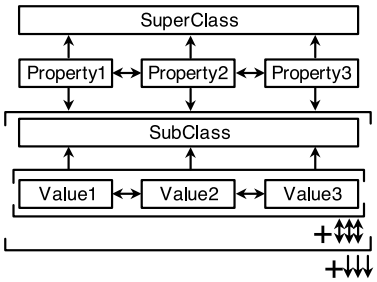

図 6 表 1 の一般化表構造
Fig. 6 Generalized structure of Table 1.

図 6 と同等の構造を，図 4 に示した記号を用いた文法で表すと，以下のような生成規則 P と開始記号 S を持つ文法 $G =< P, S >$ で表現できる．ただし E_0，E_1 は非終端記号である．

$$P = \{S \rightarrow b_1e_1b_2e_2E_0,$$
$$E_0 \rightarrow E_0e_2b_1e_1E_1|b_1e_1E_1,$$
$$E_1 \rightarrow E_1e_3b_2|b_2\}$$

本研究では，ある表に関して，図 2 のような表中に記述されたデータの関係を表す RDF ステートメントを人手で与えることで，このような生成規則を持つ文法を自動的に獲得し，その文法に基づいて用いて表全体を解析する手法を提案する．

4. 表構造の抽出の処理

1 章で述べたように，本研究では表中のデータの関係を獲得するために，以下の 3 つのステップからなるアプローチをとる．以降でそれぞれのステップについて説明する．

(1) 表構造の解釈を与える
(2) 解釈を与えた表構造の一般化
(3) 新たな関係の獲得

構造の解釈を与える 人手により表中のデータ間の関係を RDF ステートメントで記述する．ここで与える RDF ステートメントの集合は，表の構造が表すデータ間の関係と対応するものである．本研究ではそのような RDF ステートメントの集合をエピソードと呼ぶ．図 2 は表 1 の構造に対応するエピソードである．

解釈を与えた構造の一般化 ラベルを与えたボックスによる構造の表現から繰返し構造を発見し，繰返しに対応する生成規則を得る．

表の構造を表す記号列 s_{box} と既知の RDF ステートメントの集合 R から，文法 $G =< P, S >$ を得る手続き GenerateGrammar を **図 7** に示す．GenerateGrammar では，まず記号列へのラベル

```
GenerateRule(s_box, P)
  Seq ← s_box の部分配列の集合（短い順に整列）
  for each α in Seq
    if s_box 中に α(eα)+ が出現
      P' ← P ∪ {E_i → E_i eα, E_i → α}
      s'_box ← s_box 中で α(eα)+ に一致する最長
               の列を E_i に置き換え
      return GenerateRule(s'_box, P')
  return {S → s_box} ∪ P

GenerateGrammar(s_box, R)
  for each r in R
    s_box 中のボックスに r に基づきラベル付け
  P ← GenerateRule(s_box, ∅)
  return < P, S >

ExtractRelation(table, R)
  s_box ← table の構造を表す記号列
  G ← GenerateGrammar(s_box, R)
  R' ← R ∪ Parse(s_box, G)
  return R'
```

図 7 表から関係を得る手続き
Fig. 7 Procedure extracting relations from a table.

付けを行う．その後 GenerateRule に記号列と空集合を与えて呼び出し，生成規則を求める．GenerateRule では，まず s_{box} の部分配列 α の集合 Seq を作る．このとき得られた部分配列 α は長さの短いものから順に整列する．次に s_{box} 中で正規表現 $\alpha(e\alpha)+$ に一致する記号列を探す．もし見つかった場合には，$E_i \to E_i e\alpha$，$E_i \to \alpha$ の 2 つの生成規則を P に追加し，s_{box} 中で $\alpha(e\alpha)+$ に一致する最長の記号列を E_i に置き換えたものを s'_{box} とする（E_i はこれまでに追加された生成規則に現れていない新たな非終端記号とする）．複数の α についてこの置き換えが可能な場合，s_{box} の後方での置き換えを優先して行う．置き換えが行われた場合には，s'_{box} と生成規則を追加した P を与えて GenerateRule を呼び出す．ただし連続する 2 つの記号が同じでも，ボックスに異なるラベルがつけられている場合には，それらは異なる構造を表すとして，ルールの追加と記号の置き換えは行わない．

新たな関係の獲得 前のステップで求めた文法を用いて表を解析し，ボックスのラベルとセルに記述されたデータを対応付ける．このため生成規則の追加時に，終端記号のボックスとラベルとの対応を記録しておく．新たな関係を獲得するには，記号列として表した表を得られた文法に従って解析する．得られた文法で与えた表が受理された場合には，各生成規則でのボックスとラベルの対応に従って新たな RDF ステートメントが得られる．

任意の表 *table* と RDF ステートメントの集合 R から新たな RDF ステートメントを得る手続き ExtractRelation を図 7 に示す．ExtractRelation では，まず与えられた表 *table* を記号列に変換する．GenerateGrammar により文法が得られた後，文法 G に基づき記号列 s_{box} を解析して RDF ステートメントを得る手続き Parse(s_{box}, G) を実行し，新たに得られた RDF ステートメントの集合を与えられた RDF ステートメントの集合に加えて返す．

5. 評　　価

本章では，4 章で述べた処理を実際に Web 上に存在する表に対して適用した結果について述べる．

本研究の手法では，ある 1 つの表を解析するために人が RDF ステートメントを記述して与える必要がある．表と RDF ステートメントが与えられると，文法を得て新たな関係を獲得する処理は自動的に行われる．表を完全に解析する文法を得るには，繰返し構造になっていない部分と，繰返し構造を構成する要素のうちいずれか 1 つの部分に記述されたデータについて，RDF ステートメントを記述する必要がある．本研究は少ないコストで表から多くのメタデータを得ることを目的としており，本研究の有効性を知るには，RDF ステートメントを与えるためのコストと，処理の結果得られる RDF ステートメントの量の関係が重要となる．そこで，様々な表に対して本研究の手法を適用し，得られた RDF ステートメントの数と与えた RDF ステートメントの数を調べる．本研究の手法では，表中のデータについて記述を多く与えるほど，正しく表を解析できる．仮に表中のあらゆるデータについて RDF ステートメントを記述すれば，必ず正しい結果が得られる．今回の評価においては，その表から誤りなくデータを獲得するために必要となる最小の RDF ステートメントの集合を与える．

表の構造によって手法の適用の結果は異なると考えられる．従来研究[6),8)] では，レイアウトの点から以下の表のクラスが提案されており，この分類に従って処理の対象とする表を選択する．

1-dimensinal table 表の左端もしくは上端に属

Flight	Day	Departs	Arrives
SM202	Sun	11:10	13:20
LM208	Mon	20:20	21:45
LM208	Thu	20:45	22:10
LM208	Tue	20:00	22:10

(a)

	Monday	Tuesday	Wednesday
Morning	Clemens	Aaron	Celina
Afternoon	Aaron	Celina	Clemens

(b)

Items & Period		Regular	Float
Fixed Deposit	3 Months	4.4	4.4
	6 Months	4.95	4.95
	9 Months	5.05	5.05
	1 Year	5.15	5.15
	2 Years	5.25	5.25
	3 Years	5.25	5.25
Regular Fixed Deposit	1 Year	5.25	5.25
	2 Years	5.35	5.35
	3 Years	5.35	5.35

(c)

Tour Code			DP9LAX01AB	
Valid			01.05.-.30.09.04	
Class/Extension			Economic	Extended
Adult	PRICE	Single Room	35,450	2,510
		Double Room	32,500	1,430
		Extra Bed	30,500	720
Child		Occupation	25,800	1,430
		No Occupation	23,850	720
		Extra Bed	22,990	360

(d)

図 8 表の例
Fig. 8 Examples of tables.

性が記述される．属性は複数の行や列にまたがって階層的に表現されることがある．**図 8** (a) はこのクラスに属する表の例である．

2-dimensinal table 表の上端と左端に属性が記述される．1-dimensional table 同様，属性は複数の行や列にまたがって階層的に表現されることがある．図 8 (b) はこのクラスに属する表の例である．

Complex table このクラスの表は，様々な特徴を持つ．以下に特徴に基づく分類をいくつか示す．

Partition label 表をいくつかに分割するラベルが含まれる．分割された各部分は共通の属性を持つ．表 1 はこのクラスに属する．

Over-expanded label 複数の行や列にまたがるセルを含む．複数の行や列にまたがるセルは，他のより幅の小さいセルと隣接して属性の階層関係を表す場合や，連続するセルが同じデータを持っていることを表す場合がある．図 8 (c) にこのクラスに属する表の例を示す．

Combination 属性と属性値の対応などを持たない，相互に独立した複数の表によって構成される．図 8 (d) にこのクラスに属する表の例を示す．図 8 (d) では，1–2 行目と 3–9 行目が互いに独立した表となっている．

またドメインによっても現れる表の構造やデータの特徴が異なる．そこで，価格表・タイムテーブル・統計データの 3 つのドメインで，各クラスに属する表を 5 つずつ集め，計 75 の表を対象として処理を行った．表の収集のため，各ドメインについて，Google で “price list”，“timetable”，“statistics” というキーワードを与えて返された HTML 文書から，実際にそのドメインに属し，各クラスに属する表を含むものを上位から選択し，該当する Table タグを抽出した．価格表は上に述べたいずれの構造の表も比較的よく現れ，特に単純な構造の表において多数のデータが含まれることが多い．タイムテーブルでは，隣接するセルが同じ値を持つ場合などにそれらのセルが結合され，複雑な構造を持つことが多い．統計データでは単純な構造に数値が多数記述された表が多いが，他のドメインに比べ属性が記述された表の上部で複雑な階層構造が現れるものがある．

表 2 得られた RDF ステートメントの数
Table 2 The number of extracted RDF statements.

	price list	timetable	statistics
1-dimensional	93.6 (3.6)	160.8 (4.4)	163.2 (7.2)
2-dimensional	36.0 (3.2)	84.0 (3.2)	169.0 (3.0)
Partition Label	215.8 (4.4)	136.6 (5.4)	193.8 (20.0)
Over-expanded Label	104.8 (4.8)	106.2 (8.0)	184.4 (6.0)
Combination	51.0 (6.8)	154.8 (5.6)	269.8 (4.6)

以上の 1-dimensional table，2-dimensional table，Complex table の Partition label，Over-expanded label，Combination という 5 つの分類と，3 つの異なるドメインごとに得られた RDF ステートメントの数の平均を**表 2** に示す．括弧の中の数値は与えた RDF ステートメントの数の平均である．ここではプロパティの値としては，1 つのセル中のデータをとるものとした．

得られる RDF ステートメントの数は，表の大きさによって大きく異なる．同じ構造が多数繰り返される構造を持つ表では，多くの RDF ステートメントが得られることになる．表 2 に示した結果では，数値のみが記述されたセルが数多く含まれることが多い統計データの表において，比較的多くの RDF ステートメントが得られている．人手で記述して与えた RDF ステートメントの数は，多くの場合数個程度であった．多くの RDF ステートメントを与える必要があったのは，主に以下のような場合である．まず，表が数多くの属性を含む場合には，1-dimensional table のような単純な構造でも，多くの RDF ステートメントを与える必要がある．また，隣接するセルを結合して，それらのセルに同じ語が記述されることを表している表では，単純な繰返し構造が現れず，やはり解析に多く

の RDF ステートメントが必要になる．さらに表 1 の 3 列にまたがるセルが，データを記述したセル 1 つと空白のセル 2 つを用いて表現される場合，表中に挿入された分類を表す行とその他の行で構造的に差がないため，記述されたデータの種類の違いを繰返し構造で表現できず，人手での記述のコストが大きくなる．

6. 複数の表への適用

本研究の手法では，1 つ 1 つの表に人手で RDF ステートメントを与える必要があり，大量の表を処理できない．そこで，ある表から得られた RDF ステートメントを，別の表を解釈するために用いることを試みた．このための手続き ExtractFromMultipleTables を**図 9** に示す．入力として既知の RDF ステートメントの集合 R と，処理対象である表の集合 T を与える．各表について文法の獲得と新たな RDF ステートメントの獲得を，既知の RDF ステートメントの集合に変化がなくなるまで繰り返す．

ただし，エピソードとして与える RDF ステートメントが十分ではない場合には GenerateGrammar の呼び出しにより誤った文法が獲得されることがある．誤った文法に基づいて RDF ステートメントの獲得を行うことを防ぐために，以下の条件を満たす文法が得られた場合のみ Parse の呼び出しを行った．

- プロパティとプロパティ値（または上位クラスと下位クラス・クラスとインスタンス）の関係にあるラベルのついたボックスは，同じ列（または行）に位置する．
- これらの関係にあるラベルのついたボックスが，繰返し構造により 1 対多の関係にある．

これらの条件は，表構造に現れる典型的な関係の特徴であり，得られた文法が正しいかどうかを判定するのに用いている．

処理の対象となる表を集めるため，“pentium”，“price list” という語を検索エンジンに PC 部品に関するキーワードを与えて，1,000 程度の Web ページを収集した．さらにこれらのページ中のネスト構造を持たない Table タグまたは Table タグのネスト構造の最も内側にある Table タグの内容を表と見なして抽出した．

初めに与えた RDF グラフを**図 10** に示す．今回は表中で与えた RDF ステートメントで記述された語を探す際には，大文字小文字の区別をしない，一部の記号を無視するなどの曖昧検索を行った．さらに，“CPU” の同義語として “Processor” を，“Pentium 4 3.20 EGHz” の同義語として，“Pentium 4 (3.2 GHz)”，“Intel Pentium 4 3.2 GHz” を与えた．

```
ExtractFromMultipleTables(R, T)
  while R が変化する
    for each table in T
      s_box ← table の構造を表す記号列
      G ← GenerateGrammar(s_box, R)
      R ← R ∪ Parse(s_box, G)
  return R
```

図 9 複数の表を処理する手続き
Fig. 9 Procedure extracting relations from multiple tables.

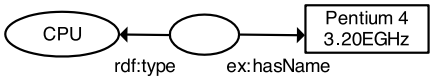

図 10 人手で与えた RDF ステートメント
Fig. 10 An initial set of RDF statements given by humans.

図 10 の RDF ステートメントを用いて図 9 の while の中の処理を一度実行すると，2 つの表から新たな文法と，合計 104 個の “CPU” クラスのインスタンスを記述する RDF ステートメントが得られた．これらの内容を確認した結果，実際に hasName プロパティにより CPU を表していると解釈できたものは 92 個であった．

2 度目の while の中の処理では，新たに 1 つの表で文法と 15 個の CPU クラスのインスタンスを記述する RDF ステートメントが得られ，いずれも hasName プロパティにより CPU を表していると解釈できるものであった．3 度目の while の中の処理では，新たな文法は得られず，図 9 の手続きは終了した．

最終的に得られたインスタンスの記述のうち，正しいものは 107 個であり，誤っているものは 12 個であった．誤っているものの内容を確認すると，“CPU” クラスのサブクラスとして解釈するべきものが hasName プロパティの値として得られていた．これは，複数列にまたがる幅の広いセルの代わりに空白セルを用いている表において，サブクラスとして解釈できる製品の分類を記述した行と，インスタンスのプロパティを記述した行を区別できず，1 つの繰返しとしてまとめてしまうことによる．

図 9 の手続きでは，初めに与える RDF ステートメントを増やしたり，同義語として多くの語を設定したりすることにより，容易に得られる RDF ステートメントを増やすことができる．また初めに与える RDF ステートメントの内容によって，どのような関係を

表す RDF ステートメントを得るかもある程度コントロールできる．しかし，ループが多く実行された場合には，誤って得られた RDF ステートメントからより多くの誤った RDF ステートメントが得られることが多くなった．

7. おわりに

本研究では，広く用いられている表から，オントロジを獲得する手法を提案した．提案手法の特長は以下のとおりである．

人手による表構造の解釈の利用 表構造が表すデータ間の関係を表ごとに人手によって解釈して与えることで，表によって同じ構造が異なる関係を表す場合でも，表中のデータ間の関係が獲得できる．

様々なドメインへの対応 表構造が表す関係に基づいて，データ間の関係を獲得するため，対象ドメインの知識ベースを用意する必要がなく，容易に様々なドメインへ適用できる．

さらに，提案手法の有用性を確認し，問題点を明らかにするために，異なるドメインで異なる構造を持つ表を収集して提案手法を適用した．その結果，表構造の解釈として人手で RDF ステートメントの小さな集合を与えることで，与えた解釈に従って表中の数多くのデータについてデータ間の関係が獲得できた．今後の課題としては，6 章で示した自動的に大量の表を処理する手法の改善があげられる．

謝辞 本研究は，日本学術振興会科学研究費基盤研究（A）(15200012，2003–2005）の補助を受けた．

参考文献

1) Maedche, A.: *Ontology Learning for the Semantic Web*, Kluwer Academic Publishers (2002).
2) Ashish, N. and Knoblock, C.: Wrapper Generation for Semi-Structured Internet Source, *ACM SIGMOD Records*, Vol.26–4, pp.8–15 (1997).
3) Cohen, W. and Fan, W.: Learning Page-independent Heuristics for Extracting Data from Web Pages, *8th World Wide Web Conference*, pp.1641–1652 (1999).
4) Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J. and Zien, J.: Semtag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation, *12th World Wide Web Conference*, pp.178–186 (2003).
5) Chen, H., Tsai, S. and Tsai, J.: Mining Tables from Large Scale HTML Texts, *18th International Conference Computational Linguistics*, pp.166–172 (2000).
6) Wang, H., Wu, S., Wang, I., Sung, C., Hsu, W. and Shih, W.: Semantic search on Internet Tabular Information Extraction for Answering Queries, *9th International Conference on Information and Knowledge Management*, pp.243–249 (2000).
7) Hurst, M.: Layout and Language: Beyond Simple Text for Information Interaction - Modelling the Table, *2nd International Conference on Multimodal Interfaces*, pp.243–249 (1999).
8) Pivk, A., Cimiano, P. and Sure, Y.: From Tables to Frames, *3rd International Semantic Web Conference*, pp.166–181 (2004).
9) Embley, D., Tao, C. and Liddle, S.: Automatically Extracting Ontologically Specified Data from HTML Tables with Unknown Structure, *21st International Conferenceon Conceptual Modeling*, pp.322–337 (2002).
10) Tijerino, Y., Embley, D., Lonsdale, D. and Nagy, G.: Ontology Generation from Tables, *4th International Conference on Web Information Systems Engineering*, pp.242–252 (2003).

(平成 17 年 6 月 1 日受付)
(平成 18 年 2 月 1 日採録)

田仲 正弘（学生会員）
2004 年京都大学工学部情報学科卒業．2005 年同大学院社会情報学専攻修士課程修了．現在，同大学院社会情報学専攻博士課程に在学中．セマンティック Web 技術，マルチエージェントシステムに興味を持つ．

石田 亨（フェロー）
1976 年京都大学工学部情報工学科卒業．1978 年同大学院修士課程修了．同年日本電信電話公社電気通信研究所入所．ミュンヘン工科大学，パリ第六大学，メリーランド大学客員教授等，経験．工学博士．IEEE フェロー．情報処理学会フェロー．現在，京都大学大学院情報学研究科社会情報学専攻教授，上海交通大学客員教授．自律エージェントとマルチエージェントシステム，セマンティック Web 技術に取り組む．デジタルシティ，異文化コラボレーションプロジェクトを推進．