日本語省略文理解のための手法とその実現法

堂坂浩二，唐沢 博，小川 均，田村進一
（阪大・基礎工）

1．はじめに

自然言語（日本語）を用いて柔軟に表現されたテキストにおいては、様々な省略表現が見られる．自然言語（日本語）文では省略しても文脈その他の情報から復元可能な要素は省略されるのがむしろ普通で[1]，省略可能な要素が省略されないで残されている場合は強調の意味あいを持つ。したがって、自然言語（日本語）を用いて表現されたテキストを理解するためには様々な省略表現を扱うことが必要不可欠となる。またこういった省略現象は後で述べるように、構文論的に解決できるものから、深い文脈の知識によってのみ解決できるものまで種々様々なものがある．よって省略表現を積極的に扱う言語処理システムは、構文・意味・文脈の知識を総合的に用いたかなり柔軟なものでなくてはならない。

本稿では、まず日本語における省略表現について考察し、そのような省略表現を含んだ日本語文を理解するモデルを考案する．次に個々の省略表現を扱う手法について述べる。また現在進行中のインプリメント実験についても述べる。

2．日本語文における省略現象

本章では、日本語文において見られる省略現象について考察し、それらの省略現象を扱うために必要な知識について考察する。

(1) 助詞の省略

〔例〕 太郎はりんご好きだ。

日本語において、助詞は、動詞に対する名詞句の格素性を表現しているものである．助詞が省略された場合、名詞句の格素性を決めるには、名詞の意味素性と動詞の格に対する選択制限によらなければならない．すなわち、助詞という構文情報の省略を意味解析によって補わなければならないわけである．助詞の省略そのものは、頻繁に見られるものではないが、次に見るように名詞句が主題化されて助詞が省略された場合や、助詞そのものにあいまいさが残る場合[2]には、同じような状況が起こる。

〔例〕 太郎はりんごは好きだ。（主題化による格素性の欠落）
花子の好きな次郎．（助詞のあいまいさ）

このような現象を扱うためには、構文だけでなく，意味・文脈の知識を必要とするのは言うまでもない．

(2) 再帰代名詞

再帰代名詞も指示対象が省略されているという意味で、省略現象の一つと言える。

〔例〕 太郎は花子を自分の部屋に行かせた．
（太郎は花子を｛太郎，花子｝の部屋に行かせた．）

(3) 構文的コントロールで説明できる'穴'(Hole) [3], [4]

〔例〕太郎は花子にプレゼントをあげると言った。

(太郎は花子に〔太郎が花子にプレゼントをあげる〕と言った。)

この例では、動詞'あげる'の主語と目的語が省略され穴が形成されているわけだが、この現象は語用論によらずとも構文的コントロールで説明されることが知られている。すなわち、'あげる'の主語の穴が主文の主語によってコントロールされ、目的語の穴が主文の目的語によってコントロールされて、穴が埋められている。(2)の再帰代名詞もこのようなコントロール現象によって説明される。

(4) 主題化文における'穴'(Hole) [4]

主題化文においては、助詞による格情報が失われてしまうことはすでに述べたが、更にこれに加えて、主題化文においては、(2), (3)のような構文的コントロールを用いないと説明できない現象がある。

〔例〕良子は、太郎が花子が殺していると思っている。

(太郎が、〔花子が良子を殺している〕と思っている。) [4]

上の例では、'殺している'の目的語が主題化され穴となっている。すなわち、殺しているの目的語の穴が主題の'良子は'によってコントロールされて、穴が埋められている。このようにコントロールの現象は、文の境界を越えても適用される。

(2)~(4)の省略現象を扱うためには、穴とその構文的コントロールを言語理解モデルの中に導入する必要がある。

(5) 並列文におけるアナロジーによる省略

〔例〕太郎は花子を殺し、次郎も殺す。

(太郎は花子を殺し、太郎は次郎も殺す。)

日本語の並列文においては、同じことの繰り返しを避けるために省略が行われる。これらの穴は、先程述べたような構文的コントロール現象では説明できない。この現象は、並列関係にある文の間でアナロジーがとられて、欠けている格要素が埋められるといったような、むしろ語用論的に説明されるべきものである。また、上の例では、第2文が'次郎も花子を殺す'といったようには解釈されないということを説明できるような制限や規定を言語理解モデルは持たなければならない。[1], [8]

(6) 「ダ」パターン

〔例〕太郎は花子を殺し、次郎は良子だ。

並列文においては、動詞までもが省略されることがある。このときに用いられるのが「ダ」パターンである。(5)においては、並列文の第2文目、目的語を欠く文としてそのままで解析できるが、上の例の第2文はそのままでは動詞までが省略されているためにどんな解釈もできない。よって、「ダ」パターンを扱うためには、構文・意味解析中にアナロジーをとっている文から動詞を類推してやる必要がある。

(7) 場面の知識を利用した省略

〔例〕 太郎は昼休みに食堂へ行った。
彼はそれほど空腹でなかったので、トーストにした。

上の例において、第2文の「ニスル」パターンは「太郎はトーストを食べることにした」というような意味である。この「ニスル」パターンは、次のように「ダ」パターンが埋め込まれた形と考えられる。〔5〕

彼は 彼を 〔彼はトーストを食べる〕した。 → (「ダ」パターン化)
彼は 彼を 〔彼はトーストだ〕した。 → (同一名詞削除)
彼は 〔トーストだ〕した。 →
彼はトーストにした。

(6)の「ダ」パターンにおいては、省略されている動詞はテキスト上に存在したものだが、この例においては文脈から予測される場面の知識を用いて、「食べる」という動詞が復元されている。すなわち、「太郎はトーストだ」という文と予測される文脈から「食べる」という動詞が連想される。ここで「トースト」という料理の意味素性を持つ名詞句がテキスト上になければ、「食べる」という動詞は連想されない。すなわち、第1文において太郎が何か料理を食べるという場面が予想されるが、その予測場面が連想されて「ダ」パターンに適用されるには、目的語となりうる名詞句の存在が不可欠である(連想の前提)。もちろん、このような省略表現を扱うためには、深い文脈の知識が必要とされることは言うまでもない。

3. 日本語文理解モデル

本章では、本システムで用いている日本語文理解のモデルについて述べる。言語理解のモデルにおいては、構文・意味・文脈の知識をどう表現するかということが中心課題であるが、本システムでは基本的に1つ1つの知識をオブジェクトと見たてて、それぞれのオブジェクトにメッセージを送ることによって文の理解を進めていくという立場をとっている。

3.1 意味表現

本システムでは、文が解析された結果変換される意味表現(深層構造)は、基本的にフレームを用いている。1つ1つの意味表現はイベントと呼ばれる。例えば、「与える」という動詞の意味表現は、Prologの項表現を用いて左のように表される。Tagは、このイベントの識別子である。Agent, Object, R_to のそれぞれの変数には行為者格、対象格、受益者格に対応する名詞のイベントの識別子が代入される。このようなイベントを管理するのがイベント管理オブジェクトと呼ばれるオブジェクトである。イベントの生成、格スロットへの値の代入などは、このオブジェクトによって行なわれる。

```
event(Tag,[concept(give),
           agent(Agent),
           object(Object),
           r_to(R_to)]
).
```

3.2 意味オブジェクト

今、「太郎が花子を愛する」という文が与えられたとする。本システムでは、「太郎が」、「花子を」、「愛する」といった句に1つのオブジェクトを対応さ

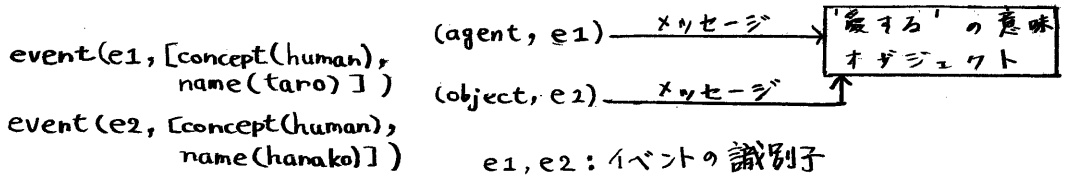

図1　意味オブジェクトへのメッセージの送信

せている。すなわち、'愛する' というオブジェクトに、'太郎'、'花子' に対応するイベントがそれぞれ行為者格、対象格となることをメッセージとして送る。'愛する' のオブジェクトはそれぞれのイベントの conceptスロットの値などから、そのイベントがその格として適切かどうか判別する。この動作を図1に示す。このオブジェクトを意味オブジェクトと呼ぶ。意味オブジェクトは意味解析を担当するオブジェクトである。オブジェクトは、次の形式で表現される。

object_define(オブジェクト名, メッセージパターン,
　　状態リスト,
　　本体　　).

状態リストはこのオブジェクトの内部状態を表わすものである。本体はこのオブジェクトにメッセージが送られたとき実行される部分で、いくつかのプリミティブが用意されている。例えば、'愛する' の意味オブジェクトは次のように表されている。これは '愛する' のオブジェクトが (agent, A) というメッセージを受けとった時の動作を記述したものである。(他のメッセージの場合の記述は別に定義している。)

```
object_define(love,(agent,A),
                   [Agent,Object],
                   (concept(A,Agent),
                    feature(Agent,animate),
                    new_state([Agent,Object])
).
```

本体部に見られるいくつかのプリミティブのうち、concept(Tag, C) は、識別子Tagをもつイベントの conceptスロットの値をCに代入するもので、feature(F1, F2)は、意味素性F1が意味素性F2として推論するものである。さらに newstate(状態リスト) は、このオブジェクトの内部状態を変更するオブジェクトである。各オブジェクトのインスタンスは、(オブジェクト名, インスタンス名, 状態リスト) の3組として管理されている。あるインスタンスにメッセージが送られると、そのオブジェクト名とメッセージパターンから object_define で定義されているオブジェクトを検索して、そのインスタンスの状態リストとオブジェクト定義の状態リストを単一化した後、本体が実行される。本体実行中に newstate というプリミティブが実行されると、そのインスタンスの内部状態は更新される。本体の実行が終るとこのインスタンスは次のメッセージを待つことになる。このようにプリミティブ newstate によって自己更新を繰り返しながら、オブジェクトのインスタンスはシステム上に維持される。

3.3　構文オブジェクト

3.2で述べた '太郎が花子を愛する' という文を理解する時、'愛する' の意味オブジェクトに、'太郎' を表すイベントを行為者格として送れるためには、'太郎が' の 'が' という助詞から この句が主語として、'愛する' に係り、また、'愛する' という動詞は主語として係る名詞句を行為者格としてその意味オ

ブジェクトに送るといった構文上の知識が必要となる。この構文知識を表現するのが構文オブジェクトである。このオブジェクトは、3.2で述べたようなオブジェクトとは趣きを異にするもので、内部状態を持たず自己更新もしない。むしろメッセージとそのメッセージに対する行動の組として表される。構文オブジェクトは、次のような5組のリストである。

[メッセージタイプ，ホールタイプ，
メッセージパターン，本体，コントロールリスト]

この5組をメッセージーアクション対と呼ぶ。これに対して送られるメッセージは次の形式による。

[メッセージタイプ，メッセージ内容]

メッセージタイプは、このメッセージが主語として送れたものであるとか，目的語として送られたものであるといったことを表すもので、構文オブジェクトは、送られてきたメッセージのメッセージタイプを持つメッセージーアクション対を見つけ、メッセージ内容とメッセージパターンを単一化した後、本体を実行する。ホールタイプ，コントロールリストについては後述する。例えば'殴する'の構文オブジェクトは次のように表現される。

```
[[sbj,hole(sbj),[Event1],
      (Sem_Obj <== (agent,Event1),event_manager <== (agent,Event1)),
      [hole(sbj),hole(ref)]],
 [obj,hole(obj),[Event2],
      (Sem_Obj <== (object,Event2),event_manager <== (object,Event2)),
      [hole(obj)]
]
```

この構文オブジェクトは、2つのメッセージーアクション対から成っていて、それぞれsbj，obj というメッセージパターンを持っている。（sbj は主語，obj は目的語を示すタイプ名）またメッセージパターン中のEvent1(Event2)は、この動詞に係る名詞のイベント名である。Sem_Obj，event_manager はそれぞれ意味オブジェクト，イベント管理オブジェクトのインスタンス名であり、オブジェクトにメッセージを送るのに　インスタンス名 <== メッセージ　という記法を用いている。

3.4　文法

文法は、DCG[7]の記法を用いて記述する。それぞれの文法カテゴリーは、その文法カテゴリーのあらわしている句に対応するイベントの識別子，意味オブジェクトのインスタンス名，構文オブジェクトを、それぞれ第1，第2，第3引数に持っている。図2に単文を理解する簡単な文法を示す。なお、図2の文法において、send_message(メッセージ，構文オブジェクト)は、構文オブジェクトにメッセージを送るプリミティブで，create_object(インタンス名，オブジェクト名)は、オブジェクト名で示されるオブジェクトのインスタンスを生成してインスタンス名を返すプリミティブである。

3.5　構文的コントロールの実現

2章で述べた(2)～(4)の省略表現を扱うためには、次の表現とその構文的コントロールの機能が必要となる。ある動詞の構文オブジェクトに含まれるメッセージーアクション対のうち、その動詞に直接係る名詞句によってはメッセージを送

```
vp(V_Event,V_Sem_Obj,Syn_Obj,_) -->
        np(N_Event,N_Sem_Obj,_,Mes_type),
        vp(V_Event,V_Sem_Obj,Syn_Obj,_),
        {send_message([Mes_type,[N_Event]],Syn_Obj)}.
vp(Event,Sem_Obj,Syn_Obj,_) -->
        verb(Event,Sem_Obj,Syn_Obj).
np(Event,Sem_Obj,_,sbj) -->
        noun(Event,Sem_Obj,_),cm(ga).
np(Event,Sem_Obj,_,obj) -->
        noun(Event,Sem_Obj,_),cm(wo).
verb(Event,Sem_Obj,
     [[sbj,hole(sbj),[Event1],
           (Sem_Obj <== (agent,Event1),event_manager <== (agent,Event1)),
           [hole(sbj),hole(ref)]],
      [obj,hole(obj),[Event2],
           (Sem_Obj <== (object,Event2),event_manager <== (object,Event2)),
           [hole(obj)]]
     ]) -->
         [aisuru],
         {event_manager <== create(Event,[concept(love),agent(A),object(0)]),
          create_object(Sem_Obj,love)}.
noun(Event,Sem_Obj,_) -->
         [taro],
         {event_manager <==
            create(Event,[concept(human),name(taro),sex(male)]),
          create_object(Sem_Obj,human)}.
noun(Event,Sem_Obj,_) -->
        [hanako],
        {event_manager <==
           create(Event,[concept(human),name(hanako),sex(female)]),
         create_object(Sem_Obj,human)}.
cm(ga) --> [ga].
cm(wo) --> [wo].
```

図2. 単文を解析するための文法

られなかったものは、その動詞にとって穴となる。メッセージーアクション対が穴となったときそれが主語の穴であるとか目的語の穴であるといったことを示しているのがホールタイプである。穴となったメッセージーアクション対はホールリストとしてひとまとめにされ、さらに高次の動詞へと渡されていく。2章の(3)の例で示した'太郎は花子にプレゼントをあげると言った'において'あげる'の主語と目的語に対応するメッセージーアクション対は、穴としてさらに高次の文の動詞である'言った'のホールリストに持ち上げられていく。こういった穴の伝播はFoo七素性の原則[3]として知られている。更に構文的コントロールは次のようにして実現できる。すなわちあるメッセージーアクション対にメッセージが送られた時、そのメッセージーアクション対のコントロールリストに含まれているホールタイプを持つメッセージーアクション対がホールリスト中にあるなら、そのメッセージーアクション対にも同じメッセージを送ってやる。本システムでは、主語の穴をhole(sbj)，目的語の穴をhole(obj)，再帰代名詞の穴をhole(ref)で表している。普通，主語の穴はより高次の文の主語によって，目的語の穴はより高次の文の目的語によって、再帰代名詞による穴は主語によってコントロールされることが知られている。[3]，[4]

4. 並列表現における省略表現のアナロジーによる復元

2章の(5)~(6)で示したような省略表現の処理について述べる。2章の(5)のように動詞が省略されていない時は、並列文の解析が終った後、まだ穴となって

れるメッセージーアクション対があるなら、並列文の間でアナロジーをとって、穴を埋めてやる。(6)のような場合は動詞を解析中に復元する必要がある。このため並列文を解析する時には、その並列文を管理するためのオブジェクトを生成して、省略されるかもしれない動詞をそのオブジェクトに記憶させ、その並列文を解析している途中で「ダ」パターンが現れたら、そのオブジェクトにメッセージを送り動詞を復元する。本システムでは、並列文における「ダ」パターンは、主文(一番上のレベルの並列文)のみにこの方法を適用する。

5. 予測文脈による省略の復元

2章の(7)で述べたような省略表現を扱うためには文脈の知識を表現する枠組が必要である。本システムでは文脈の知識はMOPs[6]を基本として表現している。文を解析して生成されるイベントをMOPs中の場面とマッチングをとることにより、場面が想起され、次に起りうるであろう場面が予測される。「ダ」パターンが現れた時、予測されている場面から動詞が予測できるならそれを用いて解析を進める。2章の(7)の例を理解するための場面知識を表現しているMOPsの記述を図3に示す。各場面は以下の構造を持つ。

scene(場面番号, lead_to(次に予測される場面の場面番号),
この場面を表現するイベント,
presupposition(前提リスト),
constraint(制約リスト)).

前提リストは、この場面を連想するのに必要不可欠なスロット名のリストである。図3の場面2では、前提リストに含まれるスロット名は、agentとobjectであるから、agentスロットとobjectスロットの値が決まればconceptスロットの値がなくても、すなわち'食べる'という動詞そのものが欠けていてもこの場面が連想されることを示している。制約リストは、この場面が想起された時に満たさなければならない条件を示している。

```
mops(go_to_restaurant,
  [scene(1,lead_to(2),
         [concept(go),agent(Agent1),d_to(D_to1)],
          presupposition([concept,agent,d_to]),
          constraint([concept(D_to,C),feature(C,restaurant)])
    ),
   scene(2,lead_to(nil),
         [concept(eat),agent(Agent2),object(Object2)],
          presupposition([agent,object]),
          constraint([concept(Object2,O),feature(O,dish),
                      eq(Agent1,Agent2)])
    )
  ]
).
```

図3 レストランでの場面を表現したMOPs

る。信頼度を計算する基準には、1)係り受け関係における統語的、及び意味的な結合度の評価，2)省略の順序に関する評価，3)視点情報による評価，4)文脈のつながり方の評価などを考えているが〔8〕，詳細は省略する。

7. インプリメントに関するコメント

現在、本報告で述べたシステムをBUP〔9〕の上にインプリメントを進めている。オブジェクトの実現は、3.2で述べた方法によっている。また、BUPは逐次的なパーザであるから、6章で述べたような信頼度によって並列的に解析を打ち切れるとは限らない。ある句の解析において、信頼度の低い解釈から先に生成された場合には、その解析を途中で打ち切ることはできない。その後で、信頼度のより高い解析結果が得られた時にその信頼の低い解析結果が捨てられる。

8. まとめ

本稿では、日本語におけるいくつかの省略表現を考察し、そういった省略表現を取り扱えるような日本語文理解モデルを考案し、言語理解に必要な知識をオブジェクトとして表現することを提案した。それらの知識は、Prologの項形式で実現されている。インプリメント実験は、まだ初歩的な段階であるが、オブジェクトとして知識を表現し、それにメッセージを送ることによって解析を進める方法は、関連ある知識がひとまとめに記述でき、柔軟な処理が可能という意味で有望な方法だと考えている。今後は、オブジェクトを記述するプリミティブを充実させ、インプリメント実験を進めて行くつもりである。

参考文献

〔1〕 久野暲，"談話の文法"，大修館書店，1978
〔2〕 柴谷方良，"日本語の分析"，大修館書店，1978
〔3〕 郡司隆男，"日本語におけるコントロール"，情報処理学会自然言語処理研究会資料35-3，1983
〔4〕 Gunji, T.，"Generalized phrase structure grammar and Japanese reflexivization"，Linguistic and Philosophy 6，1983
〔5〕 奥津敬一郎，"ボクハウナギダの文法"，くろしお出版，1983
〔6〕 Schank, R.C.，"Reminding and Memory Organization: An Introduction to MOPs"，Yale Univ. Dept. of Computer Science，1979
〔7〕 Pereira, F. and Warren, D.，"Definite Clause Grammar for Language Analysys - A survey of the Formalism and a Comparison with Augmented Transition Networks"，Artificial Intelligence，13，pp231-278，1980
〔8〕 堂坂，唐沢，小川，田村，"Prologによる省略された日本語の復元システム"，情報処理学会第29回全国大会講演論文集，2N-2，1984
〔9〕 松本，清野，田中，"BUPの高速化"，情報処理学会自然言語処理研究会資料集，39-7，1983