

物語理解のメカニズム

田中卓史

(国立国語研究所)

1. まえがき

これまで自然言語処理に関する研究は文法情報に基づく構文解析の形で進められることが多かった。言語理解のモデルを考えるうえで単一の文から構文解析木を作りだすことも重要なのであるが、複数の文から構成される文章の理解に関してはどのような情報処理のモデルを考えたら良いのであろうか。

ここでは文章理解のメカニズムを明らかにするための試みとして、物語(童話)を例に取り、演繹システム

Duck を用いて物語理解の実験を行っている。物語(童話)はその中の世界で起こる事象が時間軸に沿って述べられており、一般の文章よりも比較的単純に構成されている。複数の文に含まれる情報の問題に研究の焦点を当てるために、入力文章として自然語文の代わりに、ほぼ等価な内容を表現している命題の列を考える。物語の世界に関する知識は前向き推論規則、及び後ろ向き推論規則の形で与えておく。一つの命題が加えられると前向き推論規則が働き、入力文から想起される事象に対応して、新たな命題が定理として導かれる。物語の理解状態を確かめるための質問は、その質問に相当する命題を後ろ向き推論規則により証明する過程となる。

2. 文章の命題表現

物語「犬の話」S1 - S10 を例に取る。

「犬の話」

- S1: ある日、犬は肉のかたまりを拾いました。
- S2: 犬は肉をくわえて家へ帰りました。
- S3: 途中、小川の丸木橋までやってきました。
- S4: 橋から小川を見ると、
- S5: 川には自分の姿が映っていました。
- S6: 犬はそれを肉をくわえた別の犬だと思いました。
- S7: 犬はそれも取ってやろうと思いました。
- S8: 犬は影に吠えました。
- S9: しかし、口を開けたとたん、肉は川に落ち、
- S10: 犬は肉をなくしてしまいました。

各々の文は内容的に見ると、行為を表す文(誰が何をした)と現象を表す文(何が起こった)のどちらかに分類することができる。いずれも特定の時刻に起こった事象を表している。「ソクラテスは人間である」とか、「総ての人間は誤り易い」のような文は物語の文としては考えないことにする。

事象概念を表すための基本的な述語として "EVENT" を導入する。EVENT は事象のトークンと事象のタイプを表す二つの引数をとる。文 S1 は「ある日」の部分を除き次の命題 P1 に変換しておく。

```
P1: (AND (EVENT %1 (PICKUP DOG1 MEAT1))
          (IS-A DOG1 DOG)
          (IS-A MEAT1 MEAT))
```

文中に現れる普通名詞「犬」及び「肉」は一般の犬や一般の肉の概念を表すのではなく、特定の犬であり、特定の肉を指示しているものとする。DOG1及び MEAT1は存在を仮定した特定の犬、及び特定の肉に与えたトークン(スコールム定数)である。関数 PICKUP は「DOG1が MEAT1 を拾う」という事象概念を表している。述語 "IS-A" は「もの」概念のトークンとタイプの関係を与えている。

文 S2 は「犬が家へ帰ってしまった」のではなく「帰ろうとした」ことを述べていると解釈して次の命題 P2 に変換する。

```
P2: (EVENT %2
      (GOAL DOG1
        (PTRANS MEAT1 (HOME-OF DOG1))))
```

関数 HOME-OFは DOG1 の家の存在を示すスコールム関数である。関数 PTRANS は MEAT1が (HOME-OF DOG1) に移動することを表している。関数 GOAL は DOG1 が (PTRANS MEAT1 (HOME-OF DOG1)) を目的としていることを表している。すなわち、命題 P2 は「犬1が肉1を犬1の家へ移動させることを目的にすることが起こった」ことを表している。なお、入力文を自動的に命題の形に変換する場合には、第一文の犬や肉が第二文の犬や肉と

同一であるかどうかを判定することが問題となる。ここでは判定されたものと仮定している。

以下、同様にして入力文 S3 - S10 は命題 P3 - P10 に変換する。

P3: (AND (EVENT %3 (PTRANS DOG1 BRIDGE1))
(IS-A BRIDGE1 BRIDGE))

P4: (AND (EVENT %4 (ATTEND DOG1 BROOK1))
(IS-A BROOK1 BROOK))

P5: (EVENT %5 (REFLECT BROOK1 DOG1))

P6: (AND (EVENT %6
(MBUILD DOG1 (POSESS DOG2 MEAT2)))
(IS-A DOG2 DOG)
(IS-A MEAT2 MEAT))

P7: (EVENT %7 (GOAL DOG1 (POSESS DOG1 MEAT2)))

P8: (EVENT %8 (BARK DOG1 DOG2))

P9: (EVENT %9 (PTRANS MEAT1 BROOK1))

P10: (EVENT %10 (LOSE DOG1 MEAT1))

このような命題のセットは「もの」タイプの個体が D
OG1, MEAT1, ... などで、「こと」タイプの個体が %1,
%2, ... で表される世界(物語の中の世界)において個
体相互の関係を与えている。すなわち、次々に入力され
る命題は物語の中に登場する犬や肉などの個体相互の間
に意味的なネットワークを張ってゆく。

入力文章を P1 - P10 のような命題のセットで表した
場合に、元の文章が次々に述べられたことに含まれてい
た事象間の順序関係の情報が失われている。そこで、述
語 "PRECEDE" を用いて、事象間の順序の関係も命題とし
て陽に与えることにする。

(PRECEDE %1 %2)

(PRECEDE %2 %3)

(PRECEDE %3 %4)

...

3. 知識の表現

3. 1 前向き推論規則

文章理解のモデルとして、世界に関する知識を事実と
推論規則の形で持つ公理系(演繹システム)とそれを外
部から制御するメタシステムを考えよう。入力文は新た
に付け加えられた公理である。入力文から想起される事

象は、世界の知識を利用して加えられた公理から前向き
推論規則を利用して導かれる定理であると考ええる。前向
き推論規則は

(-> 前提 結論)

の形で表される。前向き規則は前提が与えられた時点で
適用され、導かれた結論を新たにデータベースに加えて
ゆく。推論規則の形の結論も許される。

3. 1. 1 スクリプト

次の前向き推論規則は事象 "PICKUP" をより詳細な事
象の列(スクリプト)に展開する。

(-> (EVENT ?E (PICKUP ?A ?O))

(-> (S-EXPAND ?E)

(AND (EVENT (E1-OF ?E) (ATTEND ?A ?O))

(EVENT (E2-OF ?E) (GRASP ?A ?O))

(STATE (S-OF ?E) (POSESS ?A ?O))

(PRECEDE (E1-OF ?E) (E2-OF ?E))

(PRECEDE (E2-OF ?E) (S-OF ?E))))

"?"で始まる記号は全称記号の付加された変数を表して
いる。第一命題 P1 が加えられると、この前向き推論規
則が適用される。述語 "S-EXPAND" は前向き推論の深さ
を外部から陽に制御するために加えた制御用の述語であ
る。メタシステムが命題 "(S-EXPAND %1)" を加えると、
マクロな事象 %1 はスクリプトへと展開され、事象 (E1-
OF %1) と (E2-OF %1)、状態 (S-OF %1) とそれらの順
序関係が導かれる。(E1-OF %1) は犬1が肉1を見付け
るという事象が起こったことを表している。(E2-OF %1)
は犬1が肉1をくわえるという事象が起こったことを
示している。(S-OF %1) は犬1が肉1を所有するという
状態になったことを示している。

述語 STATE は EVENT と同じようにトークンとタイプの
二つの項をとるが、命題の列に時間的な解釈を加える場
合に取り扱いが異なってくる。EVENT は時点の記述とし
て取り扱い、データベースに加えられた時点で既に過去
を記述したもものとして扱う。ステートは時間区間を記述
したもものとして扱う。述語 PRECEDE で現れるステートの
トークンは状態の始まるの時点を表すものとする。

「鏡面反射」に関する知識として次の規則を準備して
おく。動作主 ?A が物体 ?O1 に注目し、動作主 ?A が対
象 ?O2 とある状態 (?FUN ?A ?O2) にあるとき、しかもそ

の物体 ?01が動作主を反映しておれば、動作主の状態のイメージを別の状態として導くことができる。

```
(-> (EVENT ?E1 (ATTEND ?A ?01))
      (-> (STATE ?S (?FUN ?A ?02))
            (-> (EVENT ?E2 (REFLECT ?01 ?A))
                  (AND
                    (STATE
                     (S-OF ?E1 ?S ?E2)
                     (?FUN (IMAGE-OF ?A)
                           (IMAGE-OF ?02))))
                  (RESULT ?E1 ?E2))))))
```

3. 1. 2 プラン, ゴール

次の前向き推論規則は物 ?0 を場所 ?T まで移動させること "(PTRANS ?0 ?T)" を動作主 ?A が目的とした場合に、中間地点 "(HALFWAY-OF ?E)" まで移動させることがサブゴールになること、及び物を移動させる行為そのもの "(PTRANS ?0 ?T)" がプランになることを示している。この推論も制御用の命題 "(G-EXPAND ?E)", 及び "(P-EXPAND ?E)" を加えることにより初めて導かれる。

```
(-> (EVENT ?E (GOAL ?A (PTRANS ?0 ?T)))
      (AND (-> (G-EXPAND ?E)
                (EVENT
                 (G1-OF ?E)
                 (GOAL
                  ?A
                  (PTRANS ?0(HALFWAY-OF ?E))))))
      (-> (P-EXPAND ?E)
            (EVENT (P1-OF ?E)
                    (PTRANS ?0 ?T))))))
```

動作主 ?A1がある物 ?0 を所有しており、別の動作主 ?A2 がその物 ?0 を所有したいと考え、しかも別の動作主の方は犬である場合、「吠える」というプランを使うという知識を規則にする。

```
(-> (STATE ?S (POSSESS ?A1 ?0))
      (-> (EVENT ?E (GOAL ?A2 (POSSESS ?A ?0)))
            (-> (IS-A ?A2 DOG)
                  (-> (P-EXPAND ?E)
                        (EVENT (E-OF ?E)
                                (BARK ?A ?02))))))
```

3. 1. 3 必然的な関係

物の移動を表す事象 PTRANS が起こると、その物が帰着点に存在するという状態が生ずる。これを前向き規則にする。

```
(-> (EVENT ?E (PTRANS ?0 ?T))
      (STATE (S-OF ?E) (LOCATE ?0 ?T)))
動作主 ?A が吠えるとその動作主の口は開いた状態になるということを規則にする。
(-> (EVENT ?E (BARK ?A ?0)))
      (STATE (S-OF ?E) (OPEN (MOUTH-OF ?A))))
```

3. 2 後ろ向き推論規則

3. 2. 1 時間的な順序関係

後ろ向き推論規則は Prolog の節と同じように、証明すべきゴールが与えられた時点で適用される。これは質問が与えられてから考えることに相当する。後ろ向き推論規則は次の形でかかれる。

(← 結論 前提)

後ろ向き推論規則により定義される述語 PREC は時間的に隔たった事象や状態間の開始時点の前後関係を言及することができる。すなわち、述語 PRECEDEにより表される時点の順序関係に実質的な推移律を定義している。

```
(← (PREC ?A ?B)
      (OR (PRECEDE ?A ?B)
           (AND (PRECEDE ?A ?X)
                 (PREC ?X ?B))))
```

前向き推論規則を用いると PRECEDEの推移律を直接的に書くことができる。

```
(-> (PRECEDE ?A ?B)
      (-> (PRECEDE ?B ?C)
            (PRECEDE ?A ?C)))
```

後ろ向き規則が、必要としている事象間の前後関係のみ調べるのにたいして、前向き規則の方は興味の対象がないものまで含めて総ての前後関係を導くことになる。

時間的に順序付けられた事象や状態の端点を表す述語として LAST を定義する。FAST も同様に定義できる。

```
(← (LAST ?E)
      (AND (PRECEDE ?X ?E)
```

(THNOT (PRECEDE ?E ?X))))

THNOT (Micro PLANNER) は引数となるゴールの証明が失敗したときに成功するオペレータで Prolog の not に相当する。

3. 2. 2 ゴールに関する後ろ向き規則

次の後ろ向き規則は動作主 ?A が対象物 ?O を所有したくなる事象が起こるのは動作主が一般的にその対象物を好む時であることを規則にしている。

(<- (POSSIBLE-EVENT

(PE1 ?A ?O)

(GOAL ?A (POSESS ?A ?O)))

(AND (IS-A ?A ?AA)

(IS-A ?O ?OO)

(LIKE ?AA ?OO)))

(LIKE DOG MEAT)

3. 2. 3 必然的な関係

次の後ろ向き規則はある犬 ?A がある物 ?O をはなすという事象が起こるのはその犬が初めにその物を所有しており、後に口を開けたことが証明できればよいことを規則にしている。

(<- (EVENT (E-OF ?S1 ?S2) (UNGRASP ?A ?O))

(AND (STATE ?S1 (POSESS ?A ?O))

(IS-A ?A DOG)

(STATE ?S2 (OPEN (MOUTH-OF ?A))))

(PRECEDE ?S1 ?S2)))

3. 2. 4 スクリプトに関する後ろ向き規則

動作主 ?A が対象物 ?O を失うという事象が発生していることを証明するには、初めに動作主 ?A が対象物 ?O を所有しており、後にそれを放すという事象が起こっておればよいことを規則にする。

(<- (EVENT (E-OF ?E ?S) (LOSE ?A ?O))

(AND (STATE ?S (POSESS ?A ?O))

(STATE ?E (UNGRASP ?A ?O))

(PRECEDE ?S ?E)))

3. 2. 5 非単調推論規則

次々に入力文(公理)が付け加えられると、一般的に導かれる定理も増加するが、それまで導かれていた定理を取り消す必要も生じてくる(非単調論理⁽³⁾)。前向き推論規則を用いて、データベース中に陽な形で定理を導

いておくと、取り消しが生じた場合には定理がどの公理や規則あるいは他のどの定理から導かれたかを記録する Data-dependency メカニズムが必要になる。後ろ向き推論規則で導かれる定理はその定理を証明しようとする時点で推論規則が適用されるので、後ろ向き推論規則は取り消しの可能性のある定理を導くのに適している。

次の述語 PRESENT-STATE は THNOT を利用して非単調な推論を行うもので過去に動作主 ?A が対象物 ?O を所有しており、現在までにその対象物を失うという事象が起こっていないければ現在も所有状態にあることを定義している。

(<- (PRESENT-STATE (POSESS ?A ?O))

(AND (STATE ?S (POSESS ?A ?O))

(THNOT (AND (EVENT ?E (LOSE ?A ?O))

(PREC ?S ?E))))

この述語を利用して、現在、犬が肉を所有しているかどうかを決定することができる。この命題は P1 が加えられたあとは真となり、P10 が加えられたあとは偽となる。事象は時点記述として扱い、状態は時間記述として扱ってきた。状態の開始時点に関しては、

述語 PRECEDE で他との関係を与えてきたが、終了時点は明確でなかった。ここで述べた非単調な推論規則は状態の終了が特定の事象の発生するまでであることをのべている。これは時相論理における UNTIL オペレータに似ている。

4. 物語理解のモデル

4. 1 演繹システムだけのモデル

物語理解のメカニズムとしてどのようなモデルを考えればよいのであろうか。初めに、最も単純なモデルとして演繹システムだけからなるモデルを考える。入力文章から前向き推論で導かれた定理の集合を文章の理解状態に対応させ、質問文から考え始めて答えを導く過程を後ろ向き推論に対応させる。このモデルは先行する文と現在の入力文との間に成り立つ意味的關係を求めようとするメカニズムを特に含んでいない。文と文との関係としては予め与えられた事象間の順序関係だけである。従って、文と文との間に何の脈絡もない文章でもそれなりの

理解状態を得ることができる。最も、前向き規則を用いて先行する文により後続の文に適用できる推論規則を導くことや、後ろ向き規則で複数の文を条件として定理を導くことができるので、脈絡のある文章の方がより多くの定理を導くことになる。これまで示してきた推論規則から制御用に加えた述語の部分を除く除外すると、このモデルで直ちに物語理解の実験を行うことができる。

4. 2 制御系を含むモデルによる物語理解のシナリオ

物語理解のモデルとして演繹システムとそれを外部から制御するメタのシステムを考えると、より柔軟なモデルを構成することができる。制御系を定めるために物語理解のシナリオを示す。

第一命題 P1 が加えられると、制御系は PICKUP がマクロな事象であり、外にアクティブなスクリプトがないことから、制御用の命題 (S-EXPAND %1) を加えて入力命題をスクリプトへ展開する。第二命題が入力されると制御系はデータベースに加える前に、その命題で表される事象がアクティブなスクリプトから予測されるかどうかをテストする。これは P2 のトークンの%記号を?記号に書き換え証明を試みる。この場合 %2 に対応する事象はスクリプト内に存在しないので証明は失敗する。次に P2 が前文 P1 と話の内容が繋がったものとして扱えるかどうかをテストするために、P2 に含まれる項がこれまでの命題中に含まれていたかどうか調べる。これは DOG1, MEAT1 がすでに存在しているので成功する。そこで、P2 をデータベースに加え、次に制御系は事象のタイプが先行する命題とどのような関係にあるかを調べる。P2 が GOAL というメンタルな事象であり、先行する事象が行為に関する事象であることから、先行する事象 P1 が P2 を引き起こす原因になったと解釈して新たな命題

(INITIATE %1 %2)

をデータベースに加える。第三命題 P3 が入力されると P3 がプリミティブ・アクション PTRANS を表しており先行する命題 P2 がメンタルな事象 GOAL であることから、現在の事象は先行する命題のプランになっているのではないかと予測し、制御用の命題 (P-EXPAND %2) を加

え P2 をプランへと展開する。しかし、展開された命題は入力命題 P3 と似てはいるが帰着点が BRIDGE1 ではなくて (HOME-OF DOG1) となっている。ここで「犬の家が橋(の下)である」という仮定も成り立つが、さらに制御用の命題 (G-EXPAND %2) を加え第二命題 P2 をサブゴールへと展開し、さらに命題 (P-EXPAND (G1-OF %2)) を加えると、再び P3 と類似の命題が導かれる。この場合 P TRANS の帰着点が BRIDGE1 ではなくて (HALFWAY-OF %2) となっている。ここで BRIDGE1 と (HALFWAY-OF %2) が同一の個体を指示するものと仮定すると話のつじつまが合って文章のつながりがよくなる。そこでこれらが同一の個体を指示することを表す命題

(ALIAS BRIDGE1 (HALFWAY-OF %2))

を加える。また第三命題 P3 で表される事象は第二命題 P2 のプランになっていたことが導かれたとして命題

(GOAL-PLAN %2 %3)

を加える。第三命題 P3 が加えられた時点で前向き規則が働き次の命題

(STATE (S-OF %3) (LOCATE DOG1 BRIDGE1))

が定理として導かれる。

第四命題 P4 も他と同様に先行する命題から予測されるかどうかのテストを受ける。P3 に含まれる BRIDGE1 をキーワードとして「橋と小川」の場面のスクリプトが導かれ、たまたまスクリプト内に第四命題と同じタイプの事象が含まれておれば、P4 は予測された命題として直接証明されることになる。「橋と小川」のスクリプトを持たない場合においても橋と小川に関連することを表す知識

(ACROSS-OVER BRIDGE BROOK)

を最小限、持っているとする。制御系は P4 が直接証明されなくても、P3 から DOG1 が BRIDGE1 におり、BRIDGE1 が BROOK1 と ACROSS-OVER の関係にあることから、先行する命題と話がつながっていると解釈し、P4 が物理的事象であることから、P3 が P4 の事象を可能にしたと判断して、命題

(ENABLE %3 %4)

をデータベースに加える。

P5 に対して「鏡面反射」の場面を記述したスクリプトを用意しておく。P5 は先行する命題から予測できな

くても、命題 P4, P5 及び P1 のスクリプト内命題 (S-OF %1) から導かれるスクリプトの方に ATTEND が含まれており、スクリプトから与えられる関係

(RESULT %4 %5)

をデータベースに加える。

同じ「鏡面反射」の知識から動作主の状態のイメージも導かれる。ここで

(ALIAS DOG2 (IMAGE-OF DOG1))

(ALIAS MEAT2 (IMAGE-OF MEAT1))

を仮定すると P6 命題 MBUILD の第二項と一致する。そこで制御系は P5 の事象が原因となり P6 のメンタルな事象を引き起こしたと判断して命題

(INITIATE %5 %6)

を加える。

P7 が先行する命題と関連することを証明するために、動作主が対象物を所有したくなるのは動作主が対象物を好むこと、及び犬は一般的に肉を好むことを規則にしておけば P7 の事象は起こりうる事象として導かれることになる。P7 のタイプがメンタルな事象 GOAL であることから、制御系は P6 と P7 の関係を次のように与える。

(INITIATE %6 %7)

P8 は P7 を実現するためのプランとして導かれる。犬は何かを欲しいときに「吠える」プランを使う。また P8 から「吠える」と口が開いた状態になることが導かれる。また犬が物を手放すということは始めに犬が物を所有しており、後に口をあげたことが証明できればよいことを規則にしておく。

P9 は犬が物を手放すとき物の移動 PTRANS が起こることから導かれる。

P10 は「物を失う」という事象は過去に物を所有しており、後にそれを手放すという事象が起こっておればよいという規則から導かれる。

5. システムの動作と問題点

制御系を含まない 4. 1 のモデルで行った実験を示そう。P1 - P10 をシステムに公理として入力した後にシステムに発した質問とシステムからの解答を示す。それぞ

れ自然語文に翻訳した会話と実際の会話を示す。

Q1: 誰が何を拾ったか?

A1: 犬が肉を拾った。

Q2: 肉はどこへ移動したか?

A2: 肉は川へ移動した。

Q3: 犬は何をしたかったか?

A3A: 肉を家へ運びたかった。

A3B: 肉を得たかった。

Q4: 犬は吠える前に何を見たか?

A4: 犬は川を見た。

Q5: 犬は川を見たら何が起こったか?

A5: 犬が川に映った。

(DUCK)

Q1: G?> (EVENT ?E (PICKUP ?A ?O))

A1: (EVENT %1 (PICKUP DOG1 MEAT1))

O=MEAT1

A=DOG1

E=%1

?>A

NO MORE

Q2: G?> (EVENT ?E (PTRANS MEAT1 ?T))

A2: (EVENT %9 (PTRANS MEAT1 BROOK1))

T=BROOK1

E=%9

?>A

NO MORE

Q3: G?> (EVENT ?E (GOAL DOG1 ?X))

A3A: (EVENT %2 (GOAL DOG1

(PTRANS MEAT1 (HOME-OF DOG1))))

X=(PTRANS MEAT1 (HOME-OF DOG1))

E=%2

?>A

A3B: (EVENT %7 (GOAL DOG1 (POSSESS DOG1 MEAT2)))

X=(POSSESS DOG1 MEAT2)

E=%7

?>A

NO MORE

Q4: G?> (AND (EVENT ?E (BARK DOG1 ?O))

```

(PREC ?X ?E)
(EVENT ?X (ATTEND DOG1 ?A)))
A4: (AND (EVENT %8 (BARK DOG1 DOG2))
(PREC %4 %8)
(EVENT %4 (ATTEND DOG1 BROOK1)))
A=BROOK1
X=%4
O=DOG2
E=%8
?>A
NO MORE

```

```

Q5: (AND (EVENT %4 (ATTEND DOG1 BROOK1))
(PRECEDE %4 %5)
(EVENT %5 (REFLECT BROOK1 DOG1)))
XX= (REFLECT BROOK1 DOG1)))
X=%5
E=%4
?>A
NO MORE

```

質問 Q1 -Q5 に対するシステムからの解答 A1 -A5 は主として演繹システムが持つ関係データベースに対する問い合わせ能力と事実や状態間の推論を利用している。

スクリプトを用いて入力された命題を展開すると「犬は肉を見たか」とか、「犬は肉をくわえたか」などの入力文章の中に述べられていない言外の情報に答えることができる。

さらに制御系を含むモデルでは、スクリプトやプランを利用して事象間に成り立つ原因・結果、理由などの意味的關係を積極的に与えるので、事象や状態間の単なる順序関係だけでなく「なぜ犬は橋へ行ったか」、「なぜ犬は吠えたか」、「なぜ犬は肉を失ったか」などの質問に答えることができるようになる。

スクリプトやプランを用いる上で仮定を行うメカニズムが必要になる。例えば P2 の「犬が肉を家へ持って帰ろうと思った」というゴールに対して、サブゴールとプランを用いると、「犬が家への中間地点へ移動する」という命題が導かれる。ここで「中間地点」と「橋」とを同一視するという仮定を置くと、P3 はプランの実行を述べたものとして、P2 との意味的なつながりがよくな

る。P3 の BRIDGE1 を (HALFWAY-OF %2) で置き換えてやれば P2 をサブゴールとプランに展開したものから証明することができる。そこで、どのような仮定を置くかある命題が証明できるようになるかという「仮定」を求める操作が必要になる。これは文章理解のメカニズムに「確かな推論」だけでなく「仮定を置くかつじつまが合う」といった「確からしい推論」を行うメカニズムが必要になることを意味している。これは確かな推論を行う演繹システムの内部で実現することは難しく、制御系を含む全体のシステムで実現することになる。

6. むすび

単一の文の理解と異なり、文章理解を考える上では文法的・言語学的知識よりもむしろ、文章の記述^{している}世界に関する知識の方が重要な役割を果たす。演繹システムとそれを制御するメタシステムは事実

と推論規則の形で与えられた知識を利用して、文章に述べられていない言外の情報を埋め合わせ、文の記述する事象の間に意味的な関係を与えることができる。文章理解のシステムはその文章の記述する世界の知識を持っているかどうかということが決定的に重要である。

文章理解のモデルをより良いものとするために、より多くの知識を形式化してみる必要がある。また物語以外の一般の文章に対しても、どのような制御系を考えれば良いか検討する必要がある。

謝辞

演繹システム Duck の使用を許可して頂いた Yale 大学の Drew McDermott 教授に感謝する。

参考文献

- (1) Charniak, E., Riesbeck, C., McDermott, D. "Artificial Intelligence Programming", Lawrence Erlbaum Associates, New Jersey, (1980)
- (2) Manna, Z., Pnueli, A. "Verification of Concurrent Programs, part1: The Temporal Framework" Computer Science Dept. Stanford Univ., (1981)
- (3) McDermott, D., Doyle, J. "Non-monotonic Logic I" Artificial Intelligence, Vol.13, (1980).

- (4) McDermott, D., "Duck: A Lisp-based Deductive System", Computer Sci. Dept. Yale Univ. (1982)
- (5) Schank, R., Riesbeck, C. "Inside Computer Understanding", Lawrence Erlbaum Assoc., (1981)
- (6) Nilsson, N.J., "Principle of Artificial Intelligence", Tioga Pub., (1980).
(白井, 辻井, 佐藤訳「人工知能の原理」, 日本コンピュータ協会)
- (7) 田中卓史, 「概念情報処理」, 国語研究所報告83 pp105-161, 秀英出版, (1985).