解　説

● 自然言語理解のための知識表現と推論—推論†

西　田　豊　明††

1. はじめに

自然言語で書かれた文章には陽に述べられていないことがたくさん含まれており，解釈の仕方も唯一に限定できないことが多い．また，自然言語テキストからの情報の取り出し方も自明ではない．

自然言語理解システムにおいてこれらの問題を解決するためにはさまざまなタイプの知識と推論が必要である．自然言語を理解するためには，言語構造・機能に関する知識や推論だけでは十分でなく，自然言語によって記述されている対象世界に関する知識・推論が必要になってくる．知識の問題については，本特集で浮田氏が解説することになっているので，本解説では推論の側面について取りあげる．

自然言語理解システム全体からみると，自然言語はわれわれ人間の思考全般にわたっているから自然言語理解システムはあらゆるタイプの推論を行うことができなければならない，ということになるであろう．しかし，それでは有意義な結論は得られそうもないので，本解説では視点を変えて，言語解析を行うコンポーネントと問題を解決するコンポーネントが与えられたとき，そのインタフェースを実現するためにどのような推論が必要となるかという問題を取りあげ，一般的な視点から議論したい．

2. 自然言語理解における推論

これまで構築が試みられてきた自然言語理解システムは次の三つのクラスに大別できる．

（1） 対話システム

比較的短い文あるいは文の断片が解析の対象となる．

（1a） データベースへの自然言語アクセス

（1b） タスク指向の対話の理解

（1c） エキスパートシステムのユーザインタフェース

などがある．

（2） 常識的文章の理解システム

日常的な話題に関する複数の文から構成される文章を解析する．

（2a） 物語の理解

（2b） 新聞記事の理解

（2c） 情景描写文章の理解

などがある．

（3） 専門文書の理解

専門分野固有の表現を多く含んだ文章を解析する．

（3a） マニュアルの理解

（3b） 特許請求文の理解

（3c） 法令文の理解

などがある．

一般に，自然言語理解システムの構成は図-1 のように特徴づけられることが多い．

すなわち，言語解析コンポーネントでは入力テキストの形態素・統語・意味・（言語内）談話解析を行い，言語の表層的な構造に依存しない概念的な表現形式に変換し，問題解決コンポーネントに送る．言語解析コンポーネントに置かれる知識は言語構造に関する知識（いわゆる言語内知識）が中心になる．問題解決コンポーネントでは，言語表現に独立したレベルでの問題解決を行う☆．対象世界に関する知識は問題解決コン

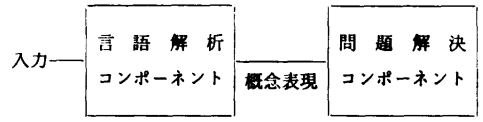


図-1　自然言語理解システムの構成
（2コンポーネントモデル）

† Knowledge Representation and Reasoning for Natural Language Understanding Reasoning by Toyoaki NISHIDA (Department of Information Science, Kyoto University).

†† 京都大学工学部情報工学教室

☆ これは必ずしも制御を言語解析→問題解決という順序で逐次的に行うことを意味していない．むしろ，これは情報のレベルとしてのみの分類であり，制御は言語解析コンポーネントと問題解決コンポーネントの間を必要に応じて行き来できるようにした方がよい．

ポーネントに置かれる.

解析する対象が曖昧性のない人工言語の場合はこのような設計が分かりやすいが, 自然言語の場合は問題がある. つまり, よく知られているように言語解析で生じる問題（たとえば, 語義や統語構造の曖昧性の問題）を解決するためには言語内知識だけでは十分ではなく, 対象世界に関する知識が必要である. したがって, 言語解析コンポーネントの問題解決能力の不完全性をなんらかの形で補償する必要がある. これにはいくつかの方式が考えられてきた.

（a） 可能な解析を全て出力する.
（b） もっとも確からしい解析だけを出力する.
（c） 可能な解析結果を一つずつ出力する.

(a)はいわゆる semantics-on-top-of-syntax というアプローチであるが, 少し複雑な文になると中間結果の数が非常に多くなり（数十から数千）, 問題を引き起こす. (b)はこれと対照的なアプローチである. 統語的な構造の「良さ」の基準を用いて優先度を与える試みはあるが, 言語内知識だけで確からしい解析を決定するのは本来無理である. 一般にはこの問題を避けるために, 対象世界の知識を制約や優先という形式で持ち込むことになるが, そのような知識は多くの場合アドホックになってしまう. (c)のようなアプローチでは(b)のような問題は生じないが, (a)と比較すると, 単に空間と時間のトレードオフを行っただけである. 対象世界の情報を予測などの言語解析コンポーネントにおける問題解決に積極的に利用できるようにするためには問題解決コンポーネントにはかなり柔軟な制御構造が要求されるが, データベース管理システムのような既存の応用システムは必ずしも自然言語理解システムを意識したインタフェース機能を有してはいない.

要するに, 言語解析-問題解決という2コンポーネントアーキテクチャではそれぞれのコンポーネントの相互依存性が高く, システム全体に高度の問題解決機能をもたせようとすると両コンポーネントの中に他のコンポーネントの情報をもち込まざるを得なくなるが, これは必ずしも好ましいことではない.

本解説では, 図-2 のように言語解析コンポーネントと問題解決コンポーネントにはさまれた第3のコンポーネント（言語-対象世界インタフェース：LDI (Language-Domain Interface) と略記する）をもつ3コンポーネントアーキテクチャを考える. ここで, 言語解析コンポーネントも問題解決コンポーネントも独自の問題解決機構をもち, 互いに他のコンポーネントのアーキテクチャを意識しないものとする. 実際, 多くの応用システムで用いられる問題解決機構は自然言語とのインタフェースを意識しているわけではない. また, 言語解析コンポーネントを設計する場合, 特定の応用を意識せずに標準的な言語解析機能を提供する方が楽である. (もちろん, 特定の応用のために拡張できるようにしておく必要があるのは言うまでもない.)

LDI が取り扱うべき問題は, 取り扱う言語表現の現象が問題解決器の能力を越えているか否かで二つのクラスに分けられる.

（1） 言語現象が問題解決器の能力を越えている場合

言語表現の方が問題解決器で用いられる問題表現よりも表現力が高く, 多くの情報を含んでいる場合であり, 対話理解システムや専門文書理解システムの一部で生じる.

もっとも典型的な場合はデータベースへの自然言語アクセス[5),15)] である. データベース管理システム (DBMS) は限られた情報への効率的ではあるが限定された範囲内のアクセスを提供するに過ぎないが, 自然言語インタフェースをとおして DBMS をみたとき, ユーザは提供された機能以上のことを期待しがちである.

データベースへの自然言語アクセスで問題になるのはデータベースに陽に与えられていない情報へのアクセスである. 求める情報がデータベース中に暗黙的に存在している場合は, LDI においてそれをつなぎ合せて, 求める情報が得られるようにする必要がある. そうでない場合, つまり求めている情報を取り出すことが容易でなかったり, 本質的にデータベース中に存在しない場合は, ユーザにそのことを知らせたり, 求められている情報に近い情報を探したりしなければならない.

データベースの意味論も問題を引き起こす. つまり, データ

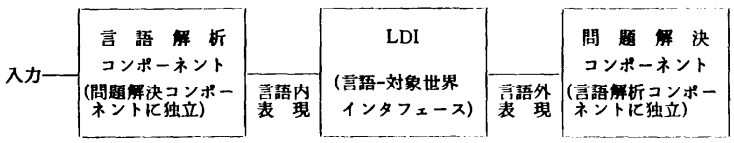


図-2 自然言語理解システムの構成（3コンポーネントモデル）

ベースシステムでは閉世界仮説が採用されていることが多いが，その場合，否定情報は陽に与えられないので，否定と無知，すなわち¬Pを知っていることとPの真偽性を知らないこと，を区別できない．もちろん，自然言語では否定と無知とを区別するから，このギャップは自然言語インタフェースにおける誤解などの問題を引き起こす．

自然言語における非手続き的表現と問題解決器の実行時の意味論との対応づけが問題になることもある．すなわち，問題解決器に与える指令は手続き的でなければならないことが多いが，自然言語表現の中には非手続き的なものも多く含まれている．

また，実行によって問題解決器がエラーを起こさないことを保証する必要もある．たとえば，「100！ はいくらか」という質問に対して，100！ をそのまま実行すると多倍長計算ができないシステムではオーバフローになることがある．このような事態を防ぐにはLDI は問題解決器の能力に関する知識と推論が必要になる．

（2） 言語の表現力が問題解決器の能力より小さい場合

対象世界の対象や事象を記述するための適切な表現形式あるいは定型的な表現 (criche) が知られていない場合や，言語表現でなされる区別が問題解決器でなされる区別より詳細度が低い場合などに生じる．

応用別にみると，この問題は特定の応用を限定していない常識的文章の理解で頻繁に生じ，専門文書の理解でも一部生じる．これはやや逆説的であるが，われわれの常識は非常に奥が深く，それを記述している言語表現に比べて複雑であるためである．

視点を変えると，この問題は与えられたテキストが求めたい詳細度の情報を含んでいないことに起因する．このような問題は一般に漠然性 (vagueness) の問題と呼ばれる．可能な解釈が離散的になった場合は，曖昧性 (ambiguity) の問題と呼ばれる．

このような問題が生じる典型的なケースは，多次元連続世界，すなわち，信号，イメージ，図形などに関する記述の取り扱いにおいてである．自然言語は非常に冗長性の高い符号系であるから，このような対象を自然言語によって詳細に記述しようとすると膨大な記述量になり，現実的ではない．実際，われわれが自然言語によって上のような対象を逐一詳細に記述することはまれである．

われわれが多次元連続世界の事態を記述するのに自然言語を用いるのは，対象の定性的な側面に言及したいときである．対象がイメージであっても，的を得た表現は画像よりも的確かつ鮮明に情報を伝達する働きがある．

(a) 広場と家は近い．
(b) 家は広場の領域外にある．
(c) 窓は家の一部である．
(d) 窓は南を向いている．
(e) 窓から見える範囲内に丘と飛行機がある．
(f) 飛行機は丘の上にある．

図-3 「広場の近くの家の南の窓からは丘の上に飛行機がながめられた」に含まれる空間に関する制約

たとえば，「広場の近くの家の南の窓からは丘の上に飛行機がながめられた」という文の解釈について考えてみよう．この文に直接現れる対象は，広場，家，窓，丘，飛行機であるが，それらの空間的な配置について厳密に述べられているわけではなく，図-3 のような制約が与えられているに過ぎない．それにもかかわらず，この文を読むとある程度情景を思い浮かべることができる．しかし，自然言語のこのような側面はまだ十分には解明されていない．

漠然性の問題は求める情報の詳細度に相対的な問題であるから，自然言語理解では常に発生し得る問題である☆．

与えられた問題の詳細度が低い場合，LDI は与えられた文章に不足している情報がなんであるか認識し，必要ならば不足した部分を推定する必要が生じる．

3. 言語現象が問題解決器の能力を越えている場合のインタフェース

3.1 インタフェースでの処理

簡単な場合からはじめて考えられるケースについて述べる．問題解決器として関係型のデータベース管理システムを想定する．

（1） データベースに暗黙的に含まれている情報を利用する

データベースに陽には含まれていないが，以下のようにデータベース中の関係をつなげば得られる場合が多い．

例1 「国立大学の先生の（標準的な）年休の日数はどれだけか」

データベース中に「国立大学の先生一年休」という関係は含まれていないが，「国立大学の先生は国家公

☆ これに対して，人工言語の場合は，表現によって行われる区別についても一定の規約があると考えられるので，漠然性や曖昧性の問題は生じない．

務員の一種（教育職）である」という情報と，「国家公務員（教育職）の（標準的な）年休の日数」に関する情報を含んでいる関係が存在すれば，その二つの関係表の join を取ることによって，求める情報を得ることができる．これは上位下位関係の典型的な利用例である．

例2 「与えられた人の勤務地を知りたい」

データベース中に，氏名-勤務地という関係はないが，氏名-勤務先，勤務先-住所という二つの関係が含まれている場合は，上と同様に二つの関係を join によってつなげばよい．

例1，例2は，データベースに潜在的に含まれている情報を最大限に利用して，データベースに仮想的に定義されている関係を言語表現と対応づけたものであると考えることができる．このような操作はふつうはデータベースのユーザ自身が行っていることであるが，自然言語インタフェースではそれを取り込む必要がある．このような視点からの研究は演繹的データベース (deductive data base) の研究分野で行われてきた．

例3 「昨日入院したのは誰ですか」

この文には様相性（時制）が含まれているが，データベースは様相性を陽に取り扱わないものとしよう．そのような場合でも，データベース中に暗黙的に含まれている様相情報を利用して上のような質問に答えることができる．たとえば，データベース中に患者-入院日という関係が含まれていれば，意味論的な変換をLDI で行うことによって，上の質問に対する答を生成できる．

（2） 部分的な回答を生成する

データベースに含まれている情報だけからでは回答できない場合でも，部分的な回答なら生成できる場合がある．

例4 「来院してすぐ入院したのは誰ですか」

この例では，副詞「すぐ」の解釈が問題になる．データベース内での時間の区別が粗すぎるとき，たとえば，日単位でしか時間情報が管理されていないとき，適切な回答を返すことはできない．しかし，条件を緩めて，「すぐ」を「データベース内で区別できる最小の単位以内で」と解釈して部分的な回答を生成することはできる．

（3） 内包的な質問に外延的情報から推定する

例5 「この会社の定年は何才ですか」

この質問に答えるためには定年の規定が必要である．しかし，規定に関する情報が得られない場合でも，データベースに登録されている外延的な情報から推定することは可能である．つまり今の場合，従業員の年齢の最大値を計算しそれが 59 だったとすると，「規定に関する情報は得られませんが，従業員データベース内のデータから判断すると，定年は 60 才だと考えられます．」というような回答を生成することは可能である．

（4） 前提についての推論

ふつうデータベースへのアクセスコマンドでは情報の新旧性は陽には区別されないが，自然言語では区別している．Kaplan の作成した COOP という自然言語インタフェース[7]は，与えられた質問の前提部（古い情報へのアクセス）と帰結部（求める情報の規定）を区別することによって，検索結果が空集合になった場合について，それがたまたま空集合になったのか，それともユーザの誤解によって生じたのかを推定する．

例6 「1986年の人工知能論で不可になった学生は誰ですか」

この質問では「1986年に人工知能論が開講され，履修したが不可になった学生が存在すること」が前提であり，求める情報は「その学生の氏名」である．COOP では質問を評価したとき空集合になった場合は，それが質問の前提を評価した結果生じた空集合によるものであるかどうかを調べる．もしそうであれば話者の前提に誤りがあると推定し，警告のメッセージを出力する．

LDI の独自の問題解決能力を強めていくともっといろいろなことができるようになる．たとえば，「もし，従業員の給与を 10 パーセント上げたらどうなるか」というような仮定法の質問や，「これに似たものはありますか」というような類推的表現などの取り扱いも可能になる．このためには，非単調推論[8]，デフォールト推論，類推などの推論機能が必要である．しかし，こうなると，ここで設定した LDI の範囲を越えることになるのでこれらに関しては本稿ではこれ以上言及しないことにする．

3.2 内包と外延の区別

いくつかの応用では，内包と外延の区別をできることが必要になる．たとえば，「あすの会議」の内包は，「この表現が置かれた文脈の次の日を開催日とする会議」というように，文脈から会議の集合への写像であると考えられる．一方，この表現の外延は与えられた具体的な文脈に対して内包が与える写像を適用してできた

像，つまり具体的な会議の集合（たとえば，{meeting 1, meeting 2}）である．入力のモード（疑問文か平叙文か）などにより内包，外延のどちらが参照されているのか区別する必要がある．

3.3 インタフェースの方式

ここでは問題解決コンポーネントの制約を受けにくい制御構造のインタフェースについて考えてみよう．問題解決コンポーネントの言語としては Prolog を用いることにする．例題として，次のような質問の取り扱いについて考えてみよう．

「明日の会議は何時からですか？」

（1） 与えられた質問を一括して変換する

もっとも単純な取り扱いは，上の質問から次のような Prolog の節を生成し，問題解決コンポーネントで実行することである．

today (T), next (T, D), meeting (M), date (M, D), start-time (M, Time), show (Time).

この方法の長所は LDI の制御構造が簡単なことであるが，一方，いろいろな言語現象に対応できないという問題がある．このような一括処理をすると，前提に誤りのある質問に対して柔軟な対応がしにくい．

（2） データ指向プログラミングを利用する（フレーム型）

Bobrow らが作成した GUS[1] をはじめとして多くの自然言語インタフェースで採用されている方式である．上での例では，図-4 のようなフレームをいったん作成してからフレームインタプリタにかける．

この方式の長所は柔軟な制御構造が実現できることであるが，一方，短所として制御が入り組んで設計が困難になる傾向が強い点があげられる．

（3） ルール指向プログラミングを利用する

第三の方式はプロダクションシステム型の知識表現を用いた方式である．作業記憶に与えられた入力の特性に応じて起動されるプロダクション規則によって，入力の解釈や必要な行動の決定を行う．実行は，入力文に含まれている情報の新旧性を反映した順序で入力の各部に対応する Prolog アトムを逐次評価していく．制御構造としては agenda 機構が適切であろう．

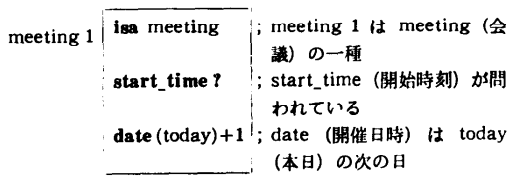


図-4 「明日の会議は何時からですか？」のフレーム表現

上の例については実行は次の順序で行う．

today(T)
next (T, D)
meeting (M), date (M, D)
start_time (M, Time)
show (Time)

その途中で解が空になると，それが文の前提にあたっていれば，警告のメッセージを出力する．たとえば，上で date (M, D) まできたところで，fail が生じると，「発話者はあす開催される会議があると考えているが，そのような会議はない」と推論し，たとえば「あす会議はありませんが」という発話を生成することができる．

同様に「明日」に会議が二つ以上ある場合については，値が唯一に定まるべき変数を指定し，その条件をチェックすることによって発話が曖昧であることを発話者に知らせることができる．

このような方式の長所は知識の追加，例外処理が容易なことである．一方，短所はルールの管理が困難になることである．

4. 言語の表現力が問題解決器の能力より小さい場合のインタフェース

このような場合言語処理コンポーネントは，不足した情報を推定する補完能力が必要である．また，新しい情報が与えられたとき，以前設定していた仮説を修正する機構をもつ必要がある．

4.1 言語理解を仮説を用いた説明過程と捉える立場

基本的な推論機構として，仮説推論，デフォールト推論をはじめとする非単調推論，不確実な知識や情報に関する推論が必要である．また全体的な推論機構に関しては統合的なパーシング[12]という視点が有効であろう．

（1） マーカ伝達

Charniak と Hirst のモデル[2),3),6)]は，Fahlman のマーカ伝達という概念[4)] に基づいて，与えられた発話を説明するもっとも確からしい説明を生成する方法を与えている．この方法は，Waltz と Pollack のモデル[16)]のように非記号的な処理に全面的に依存するのではなく，経路証明と呼ばれる論理的な操作も併用したハイブリッドな構成を採用している．

マーカ伝達のモデルでは，入力文に含まれる語の異

なる語義に対してマーカが生成される．生成されたマーカは，知識として与えられた概念のネットワーク上を伝播する．各マーカには活性度を示す値（zorch）が含まれる．zorch の値は出発点から遠ざかるにつれて小さくなっていき，設定された閾値より小さくなるとそれを含んだマーカとともに消える．このような伝播の結果，マーカが入力文中の別の語の語義に到達すると，そのときのマーカの伝達経路にあるノードとリンクが活性化される．活性化された経路はその両端のノードに概念的な関係があることを示すものであり，経路証明と呼ばれる．経路証明は論理的な証明ではなく，いくつかの仮定を含んだ一種の説明である．経路証明の確からしさはそこに含まれている仮定の数の少ないほど高いと考えられる．

（2） 劉-西田-堂下の統合パーサ

劉-西田-堂下のパーサ[11]では，矛盾の生じない範囲内でもっとも確からしい仮説をたてて，与えられた発話の各部の脈絡の復元を試みる．このパーサは二つのコンポーネントからなる．

（a） CME (Consistency Maintenance Engine) は，論理的整合性を維持する働きをする．観測された事実にあわなかったり自己矛盾に陥る仮定の組合せが見つかると，それを記録し，以後の解析過程で同じ仮定の組合せに基づいた解析が行われないようにする．また，部分結果を保存し，その後の解析に再利用する．

（b） PME (Plausibility Management Engine) は，どのような仮定をたてどの方向に解析を進めたらよいかを判断するために，信念 (belief) の確からしさに関する情報を維持する．

（3） その他

Schank のスクリプト理論[14]も以上のような視点に基づくものである．またこれ以外に，確からしい仮説を生成するためには類推や事例ベースの推論によって類似の事例を検索し，利用することも有用であろう．

4.2 記号表現の背後の深層知識を利用する

対象世界が多次元連続の場合は可能性の数を数えあげることができないという問題がある．これに関してはファジィ推論[9]や分散表現[13]などの考え方を利用することが考えられる．

山田-西田-堂下はポテンシャルモデルという手法を提案している[17]．これは次のような仮説に基づくものである．

具体化仮説：われわれ人間は情報量の低い記述をそのまま放置せず，過去の経験などに基づいて積極的に仮説を立て，推論を行いやすいレベルまで詳細度を高めている．

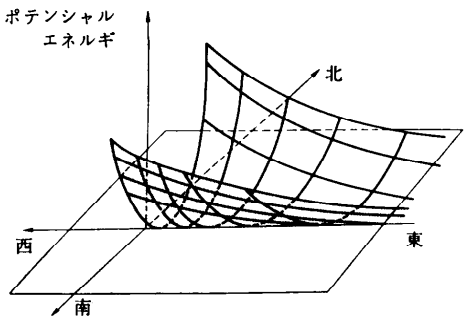


図-5 方向概念「～の東に」を表すポテンシャル関数

与えられた文章に十分な情報が含まれていないときは，われわれは仮説をたてるためにある種の知識源にアクセスしているはずである．そのような知識源の一つは，言語表現の背後にある「語感」とでも呼べるものであろう．山田-西田-堂下はこれをポテンシャル（エネルギ）関数を用いて記述する．この関数は与えられたパラメータ値の組の受け入れ難さを表示するものである．**図-5** に方向概念「～の東に」という概念に対応づけられたポテンシャルモデルを示す．

以上の考え方は情景描写文の解析に応用され，SPRINT というプログラムが実現された．解析ではさらに，言語表現に含まれる視線や視点の動きに関する断片的な情報や，情報の新旧性に関する情報の抽出も行われる．**2.** 図-3 で述べた問題については，SPRINT の言語解析部は**図-6** のように分析する．

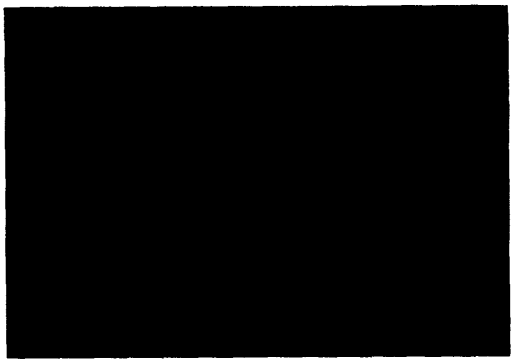
抽出された空間的関係からポテンシャル関数が生成される．ポテンシャル関数の最小化（実際は極小化しかできない）を行うと，**図-7** のような空間イメージ（内部的には3次元モデルになっている）が生成される．「上」の解釈に関して曖昧性が生じたので，ここでは二つのイメージが生成されている．

生成された空間イメージは，暫定的なものである．さらに情報が与えられると，それに対応するポテンシャルが重畳され，最小点の移動にともなって空間イメージも変化する．

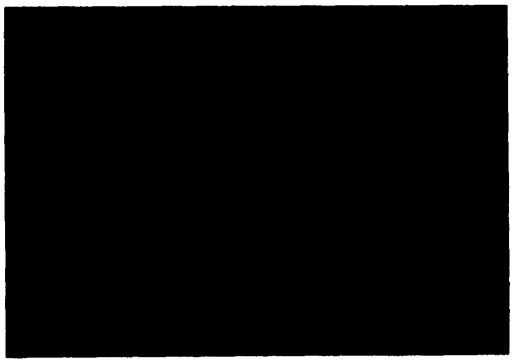
上に述べたアプローチを実現するためにはコネクショニストモデルの考え方[10]も有用であると考えられる．またそれ以外に，多次元連続世界を取り扱う方法として，ある観点からみたとき定性的に異なるものに分類する（分節化：articulation）という考え方も有用であろう．

「広場の近くの家」
「広場」: 2次元領域，地面の上（デフォールト）
「家」: 3次元対象，地面の上（デフォールト）
「広場」と「家」の位置関係：近い，領域外
「家の南の窓」
「窓」: 3次元対象
「家」と「窓」の関係：
家は壁（地面に垂直な2次元領域）をもつ
「窓」は壁に含まれる
「窓からはながめられた」
「視線」の生成：
起点＝窓，向き＝窓に垂直（＝南）
「丘の上にながめられた」
「丘」: 3次元物体，地面の上（デフォールト）
「視線」: 向き＝丘の上
（領域内か領域外か曖昧）
「飛行機がながめられた」
「飛行機」: 3次元対象
「視線」: 終点＝飛行機

図-6 「広場の近くの家の南の窓からは丘の上に飛行機がながめられた」に含まれる空間的な制約の記号レベルでの解釈



（a） 解釈1：「上」を「離れて上」（上空）と解釈した



（b） 解釈2：「上」を「接して上」と解釈した

図-7 入力文「広場の近くの家の南の窓からは丘の上に飛行機がながめられた」に対して再現された情景．「の上に」の曖昧性に対して2とおりの解釈が得られている．

5. ま と め

本解説では自然言語理解に関する推論の問題を取りあげた．問題独立の汎用の言語解析器と，とくに自然言語インタフェースとの接続を意識していない問題解決器との間のインタフェースの問題に焦点を絞り，取り扱う言語現象が問題解決器の能力より複雑な場合と，その逆の場合についてそれぞれ考察した．

参 考 文 献

1) Bobrow, D. G., Kaplan, R. M., Kay, M., Norman, D. A., Thompson, H. and Winograd, T.: GUS: A Frame-driven Dialog System, Artificial Intelligence, Vol. 8, pp. 155-173 (1977).
2) Charniak, E.: Passing Markers: A Theory of Contextual Influence in Language Comprehension, Cognitive Science 7, pp. 171-190 (1983).
3) Charniak, E.: A Neat Theory of Marker Passing, in Proc. AAAI-86, pp. 584-588 (1986).
4) Fahlman, S.: NETL: A System for Representing and Using Real-World Knowledge, The MIT Press (1979).
5) Hendrix, G., Sacerdoti, E., Sagalowicz, D. and Slocum, J.: Developing a Natural Language Interface to Complex Data, ACM Trans. on Database Systems, Vol. 3, No. 2, pp. 105-147 (1978).
6) Hirst, G.: Semantic Interpretation and the Resolution of Ambiguity, Cambridge University Press (1987).
7) Kaplan, S. J.: Cooperative Responses from a Portable Natural Language Database Query System, in: Brady, M. and Berwick, R. C. (eds.): Computational Models of Discourse, The MIT Press, pp. 209-266 (1983).
8) 松本，佐藤：非単調論理と常識推論，情報処理，Vol. 30, No. 6, pp. 674-683 (1989).
9) 水本雅晴：ファジイ理論とその応用，サイエンス社（1988）.
10) 西田：コネクショニストモデルによる自然言語処理—可能性を探る—，情報処理学会「自然言語処理技術」シンポジウム論文集，pp. 7-24 (1988).
11) Nishida, T., Liu, X., Doshita, S. and Yamada, A.: Maintaining Consistency and Plausibility in Integrated Natural Language Understanding, Proc. COLING-88, pp. 482-487 (1988).
12) 西田豊明：自然言語理解入門—ことばがわかるコンピュータをめざして—，新オーム社文庫，オーム社（1988）.
13) Rumelhart, D., McClelland, J. and the PDP Research Group: Parallel Distributed Proces-

sing, Vol. 1, 2, The MIT Press (1986).
14) Schank, R. C. and Abelson, R.: Scripts, Plans, Goals, and Understanding, Lawrence Erlbaum Associates (1977).
15) Waltz, D. L.: An English Language Question Answering System for Large Relational Database, C. ACM, Vol. 21, No. 7, pp. 526-529 (1978).
16) Waltz, D. L. and Pollack, J. B.: Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation, Cognitive Science 9, pp. 51-74 (1985).
17) 山田, 網谷, 星野, 西田, 堂下: 情景再構成としての文章理解, 人工知能学会研究会資料 SIG-KBS-8902-6(7) (1989).

(平成元年6月21日受付)