Vol. 30 No. 10　　情　報　処　理　　Oct. 1989

解　説

談話理解実験システム DUALS†

橋田浩一†† 杉村領一††† 田中裕一††

1. はじめに

DUALS (Discourse Understanding Aimed at Logic Systems) は ICOT で開発中の談話理解実験システムであり，第1版，第2版，第3版を経て，現在では第3.5版まで開発が進んでいる．以下では第3版[11]，特に談話理解特有の機能にもっとも深く関わっている問題解決モジュールについて述べる．

DUALS-III の構成を図-1 に示す．このシステムは，DUALS-I と DUALS-II の開発を通して蓄積された要素技術として得られる汎用の手続きモジュール群および言語データベースからなる，LTB (Language Tool Box) と呼ばれるシステムの上に構築されている．LTB は，主として形態論的制約と統語論的制約の処理を行うためのモジュール群とデータベースからなる．構文解析用モジュールは SAX と呼ばれ，Prolog のような論理型手続き言語に適したアルゴリズムを用いて高速の処理を実現している．形態素解析モジュール LAX は，このアルゴリズムを形態素解析用に特化したものである．生成モジュールは，省略や代名詞化，語順の決定，語集の選択などがすべて終わった後の文構造の表現を入力として，それに表層の形を割り当てる．言語データベースは辞書と文法からなり，これらの一部は上記の手続きモジュールにおいて用いられる．LTB は，談話処理に限らず他の自然言語処理にも広く用いることができる．

DUALS-III は，LTB の上に新たに問題解決モジュールと文生成プラニングモジュールを加えたものである．問題解決モジュールは，後述のように，ネットワークのような組合せ的な対象に課せられる制約を処理する，一般的な制約処理系であり，この単一の機構によって，照応の処理，問題解決のための推論，会話の制御などを統一的に行う．文生成プラニングモジュールは，問題解決によって得られた意味構造を言語表現に変換するために，省略や代名詞化，語順の決定，語集の選択などを行い，得られた構造を文生成モジュールに送る．

このように DUALS-III は，既存の手続きプログラミングのパラダイムがもたらした要素技術に問題解決などの新しい技術を接続しようという試みであったが，この試みを通じていくつかの問題点が明らかにされた．その反省に基づき，とりあえず DUALS-III の場合よりも限定された範囲の問題を明示的に手続き型プログラミングによって扱うシステムとして作成したのが DUALS の3.5版であるが，本格的な談話処理

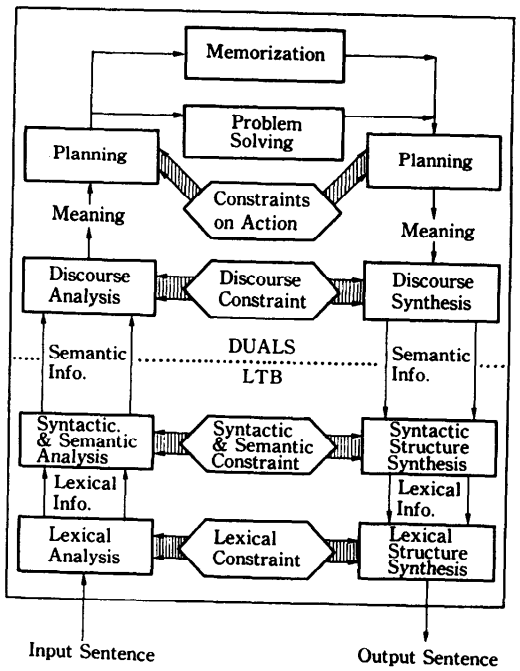

図-1　DUALS-III の構成

† DUALS—An Experimental System for Discourse Understanding by Kôiti HASIDA, Ryoichi SUGIMURA and Yuichi TANAKA (The 2nd Research Laboratory, Institute for New Generation Computer Technology).

†† (財)新世代コンピュータ技術開発機構
††† 現在，(株)松下電器情報通信関西研究所

システムを実現するには，DUALS-III の問題解決モジュールにおいて用いられた制約プログラミングのアプローチが必須と考えられる．

以下では，この問題解決モジュールについて解説し，制約パラダイムによる統合的な自然言語処理の展望を示す．まず 2. において，談話理解システムに課せられた要請を満たすためにはある種の制約プログラミングが不可欠であることを述べ，このモジュールで用いた処理方式について簡単に解説する．3. では，照応や発話行為の処理なども含め，この処理方式による談話の処理について概説した後，それに基づく問題解決の1例を詳述する．4. ではまとめを行い，今後の研究の方向を論ずる．

2. 制約パラダイム

普通，計算機で解いている問題は，完全情報問題である．つまり，関係する情報を十分に参照することができ，解（問題になっている対象の構造）を一つに定めることができる．しかし，ほとんどの実際的な問題は部分情報問題である．すなわち，問題になっている対象の構造を唯一に決定しうるほど十分には情報を参照することができない．この情報の部分性 (partiality of information) は，たとえば，明日の天気が分からない（知識の部分性）とか，将棋のルールを知っていても必勝手順を知っていることにはならない（処理の部分性）など，日常的にごく普通に生ずる事情である．

特に，人工知能の問題はすべて部分情報問題であり，自然言語の処理はそのなかでも典型的なものである．たとえば，曖昧性は情報の部分性によって生ずるものであり，文の解析や照応の同定など，自然言語処理において人間も計算機もしばしば直面する問題である．情報の部分性が引き起こす最大の処理上の困難は，文脈依存性である．文脈とは参照可能な情報の集合であり，文脈依存性とは文脈に応じて情報処理の仕方が，したがって意味などが変化することである．たとえば，「水ちょうだい」という文を考えると，この発話が台所で行われたものであるという情報が参照可能か否か，その話者が手に花びんを持っているという情報が参照可能か否かなどに応じて文脈が異なり，それに応じてこの文の解釈も異なる．特に談話処理のような場合には，この水の例からも分かるように，どの情報が参照可能でどの情報が参照不可能かということに関する組合せ，すなわち文脈の多様性は莫大なものとなる．情報の伝達はさしあたって情報が参照可能なところからそうでないところへ向かうから，そこで生ずる情報の伝達の仕方の多様性も莫大なものとなる．つまり，きわめて多様な情報伝達のパターンが，適当な文脈を与えれば生じうる．（一つの文脈で多様な情報伝達が行われるということではない．）

このような問題を手続き型プログラミングによって処理することはできない．手続き型プログラミングは制約の内容とともにその制約を処理する際の情報の伝達の仕方を明示する方法であるから，上記のように莫大な多様性をもつ情報伝達の仕方を記述しつくそうとすると，プログラムが非常に複雑なものとなる．しかもそれは，プログラムがモジュール構造を欠くという，たちの悪い複雑さである．

制約自身は，上の例に関連していえば，水に関する化学的知識，料理に関する知識，植物に関する知識，文法的知識，話者の心理状態に関する知識というようなモジュール構造をもっているだろう．プログラムにモジュール構造をもたせ，人間の手で管理可能にするためには，制約のモジュール構造がプログラムのモジュール構造に反映されなければならない．しかし，制約の各部の内容と，それに関わる情報伝達の仕方とをからみ合わせて記述するという手続き型プログラミングのアプローチはここで破綻する．すなわち，そのようなアプローチによって制約のモジュール構造をプログラムのモジュール構造に反映できるのは，可能な情報伝達の多様性を狭く限定した場合のみであり，その場合は必要な情報処理様式の実現を諦めたことになる．手続き型プログラミングによって扱われてきた問題と談話のような問題との最大の違いは，後者においては，文脈の構造，つまり情報の参照可能性の分布が制約のモジュール構造と直交する多様なものであり，実現すべき情報伝達の可能性がきわめて多様だという点にある．その情報処理の構造は，制約のモジュール構造よりもはるかに複雑な非モジュール的なものとなり，そのような情報伝達を制約の各部の内容ごとに明示的に指定するプログラムは，実際問題として人間の手に負えないほど複雑なものとなる．

プログラムのモジュール構造を保証しつつ多様な情報伝達を実現するには，制約の内容と情報伝達の仕方とが別々に，互いからなるべく独立に抽象的な形で記述されなければならない．そこでは，各種の制約は，処理手順を捨象した制約そのものとしてプログラムされ，このプログラムの構造は制約のモジュール構造を

そのまま反映する．その制約を，制約の形式のみに依存し内容に依存しないインタプリタが処理する．システムに内在化されている情報は常に制約の形で表現されることになるから，その処理とは，対象の構造を決定すること（情報の部分性によりそれは不可能である）ではなく，制約から他の形の制約への変換である．この処理は，制約のなかから矛盾する部分を除くと同時に，問題になっている対象の構造を徐々に特定してゆくようなものとなろう．たとえば Prolog の場合は，プログラム節の本体 (body) におけるリテラルの順序が情報伝達の方向を規定しているから，制約の内容と情報伝達の仕方とを分けて記述するという要請は満足されない．また，Prolog の計算は主として対象（変数）の構造を決定する（具現化する）ことであり，プログラムの変換ではない．

談話理解のような部分情報問題を扱うためには，ある種の制約プログラミングが必須であることが分かった．しかし，既存の制約プログラミング[3),4),10),12)]で扱っているのが実数や有理数やブール代数のような領域に関する制約であるのに対し，談話理解で扱うべき制約は，談話構造や意味構造などの組合せ的 (combinatorial) な構造をもつ対象の領域に関するものであるから，われわれが必要としているのは別種の制約プログラミングである．

DUALS-III の問題解決モジュールでは，**依存伝播** (dependency propagation)[6)-8),15)] という計算方式を用いて談話の処理を行っている．依存伝播においては，組合せ的な対象の領域として Herbrand 空間を考え，制約をその上の1階論理プログラムとして表現し，そのプログラムに対するある種のプログラム変換として計算を定式化する．この方法は，記号的な側面では一般の導出原理 (resolution principle) による定理証明にほぼ相当するものであり，これによって上記のような情報伝達の多様性を捉えることを目指した．

ただし，情報伝達の多様性とは，任意の文脈において情報処理が無制限に行われるということではなく，適当な文脈を与えれば可能であるような情報の伝達が全体として多様であるということである．おのおのの文脈においては，処理の範囲は厳しく制限されなければならない．そのために，制約のどの部分に対して処理を行うかを制御するためのヒューリスティクスが必要となる．一般的な導出法による計算は効率が悪いといわれているが，それは適切な制御構造を欠いているからであった．DUALS-III の問題解決モジュールでは，**依存関係** (dependency) と**ポテンシャル・エネルギ** (potential energy) というものを用いて，この処理のヒューリスティクスを与えた．

依存伝播においては，制約はプログラム節の集合として表現される．また，Prolog と違い，プログラム節が Horn 節である必要はないし，また，閉世界仮説は必ずしも前提されない．プログラム節の中に**先頭節** (top clause) と呼ばれる特別な節がただ一つあり，これは Prolog の問合せ (query) に対応する．計算の目的は，Prolog と同様，先頭節の否定（存在限量化されたリテラルの連言）を証明すること，または先頭節の否定が正しいことの abductive な説明[9)]を与えることである．

リテラル，束縛，および束縛の否定をひっくるめて**要素制約** (atomic constraint) と呼び，変数，述語，関数記号をひっくるめて**対象** (object) と呼ぶ．**依存関係** (dependency) とは，連言によって結ばれた二つの（同一かもしれない）要素制約の間での対象の共有関係のことである．たとえば，$p(X, X)$, $p(X, Y) \wedge q(Y)$, および $p(X, Y) \wedge p(A, B)$ にはおのおの依存関係がある．

依存関係によって，その依存関係を解消する処理が起動される．その際，制約のその部分の周辺にしばしば新たな依存関係が生じ，こうして依存関係の伝播にともなって処理が進行する．そのときの各処理は，展開/畳込み (unfold/fold) プログラム変換[13)]のような変換，または要素制約同士の単一化（導出法の議論においては，リテラルの間の単一化を普通は因子化 (factoring) というが，ここでは束縛の間の単一化も含めて考えているので，いずれも単一化ということにする．）である．依存関係が処理を起動するという実行制御が，制約の内容から独立に抽象的に規定されるものであることに注意されたい．

たとえば，(1)のような先頭節は述語 memb の定義(2)に従って(3)のように変換され，このとき，新述語 p は(4)のように定義される．

```
(1)  :-memb(X, [a, b]).
(2)  memb(A, [A|_]).
     memb(A, [_|S]):-memb(A, S).
(3)  :-p(X).
(4)  p(a).
     p(b).
```

変数 X に関する依存関係が解消されていることに注意されたい．また，(5)が先頭節とすると，これは(6)に変換される．

(5) :-memb(X, [a, b]), memb(X, [b, c]).

(6) :-X=b.

これらの例から分かるように，特にホーン節論理の場合，依存伝播によって対象の構造の可能性が絞られていく．

ホーン節論理で閉世界仮説を置いた場合，変数に関する依存関係があるということは，制約のその部分が充足不能かもしれないということであり，逆に，すべての述語が本体にそのような依存関係のない定義節をもつならば，先頭節の否定は充足可能である（有限失敗をもたない）．したがって，閉世界仮説付きのホーン節論理の場合，変数に関する依存関係を解消するというのは，非常によい処理のヒューリスティックである．依存伝播が処理する制約はホーン節論理に限定されず，しかも閉世界仮説を用いるわけでもないが，その場合でもこれは依然としてかなりよいヒューリスティックと考えられる．また，述語の共有をも依存関係とみなすのは，たとえば :-p, $\neg p$. のような矛盾や :-p, p. のような冗長性を処理するためである．

このヒューリスティックにおいて，依存関係に加えポテンシャル・エネルギなるものを措定することにより，ポテンシャル・エネルギの高い依存関係が処理を起動するという実行制御を考える．ポテンシャル・エネルギは，制約の各部分に割り当てられたアナログ量であり，**活性度** (activation) と**仮説コスト** (assumption cost)[9] という二つの成分の和である．活性度は注意 (attention) の度合ないしは重要度[5] のようなものを表し，制約のあらゆる部分に割り当てられる．一方，仮説コストは要素制約に割り当てられ，その要素制約の成立の疑わしさを表現する．

制約は要素制約，対象，節などからなるネットワークとみなせる．このネットワーク上で活性拡散 (spreading activation)[5),16),17] が生じ，依存関係を伝わって上記の活性度が伝播する．活性拡散はネットワーク中の関連する部分を同定する方法である．

拡散のもとになるのは，依存関係によって生ずる活性度である．変数に関する依存関係は，二つの引数位置がその引数の束縛に対して課する制限の強さに応じた活性度を生む．また，述語およびファンクタに関する依存関係は，二つの要素制約の仮説コストの差が大きいほど大きな活性をもつ．これら二つの要素制約を単一化すると，その結果できる要素制約の仮説コストは，もとの要素制約の仮説コストの低いほうに一致し，こうして全体としてポテンシャルが下がる．依存関係を解消する処理のほかに，ポテンシャルの高い要素制約が実行または展開されるという処理もある．詳細は次章の例題に譲る．

3. 談話と計算

DUALS-III の問題解決モジュールは，依存伝播に基づいて，照応，問題解決，質問応答などを統一的に処理する．すなわち，照応のためだけの特別な計算機構とか，問題解決のための専用の手続きなどはない．以下では，談話の処理のさまざまな側面が制約の処理として実現される仕方を概説した後，問題解決のある具体例について詳述する．

会話の進行も制約の変換として扱われる．先頭節（の否定）のなかに要素制約 listen(ξ) がある場合，これは，DUALS-III が，入力文を受け取ることによって外界の情報を取り込もうとしている，ということを意味する．ξ は，その入力文中に含まれる照応表現の指示対象となり得る対象や事実を要素とする集合である．listen(ξ) を処理することによって入力文 s が解析されると，この要素制約はその文の命題内容などに相当する制約に変換される．

その結果の要素制約のなかには，listen(η) または speak(ϕ, η) の形の要素制約が含まれており，η は ξ の要素（の一部）の他に，文 s によって導入された対象を要素とする．ここで，speak(ϕ, η) は，DUALS-III が ϕ という内容にあたる発話を行う，という制約である．その発話のなかでは，η の要素を指示対象とする照応表現を用いることができる．

制約パラダイムにおいては，言語の使用を考える際，言明の命題内容のみに注目するのではなく，おのおのの発話の「意味」を自然に発話行為 (speech act) として，つまり，主張，質問，約束などの行為として捉えることになる．たとえば，ある発話が質問であるとは，聞き手がその質問の答に当たる発話を行うべきだという制約がその発話によって生成されるということである．同様に命令は，聞き手がある一定の行為を行うべきだという制約となる．このような制約は，行為を対象として捉えたうえで，その行為の構造に関する制約として表現できる．ただし，DUALS-III においては，実際に行為を遂行するのは文の入出力に近い手続きモジュールなので，問題解決モジュールにおいて行為を明示的に対象化しているわけではない．

たとえば入力された文が質問だった場合，問題解決モジュールはその質問に答えるべきだという制約を課

されることになる．その質問の答を求めるための問題解決も，制約の変換である．質問の命題内容にあたる要素制約は仮説コストが高く，主張の命題内容にあたる要素制約は仮説コストが低い．したがって，これらの要素制約の間の依存関係のポテンシャルが高まる．これによって，これらの要素制約の間で単一化が起こると，答が求まってポテンシャルが下がることになる．むろん，後に示すように，その際に行われる計算は要素制約同士の単一化だけとは限らず，リテラルの展開や実行が行われることもある．

照応の処理も制約の変換として実現される．したがって，照応の処理と問題解決とを協調させつつ並列的に行うこともできる．たとえば，「彼は何を買ったのですか」という質問に答える際，代名詞「彼」に関する照応の処理と，何を買ったかを求める推論とは，同時に，かつまったく同じ処理方式によって行われる．そもそも二つの処理ははっきり分けられるわけでもない．すなわち，「彼」の指示対象を同定する際にも，買われた物を探索する際にも，だれかが何かを買ったという情報が用いられており，先行文脈中で「買った」に対応する部分を見出すことが，同時に両方の処理の一部となっている．しかも，こうした協調関係は明示的にプログラムされるのではなく，一様な処理機構から自然に生ずる．

以下では問題解決の一つの具体例について少し詳しく述べる．DUALS-III が「人間が，この地球の上で生き続けていくためには，どうしても，自然の恵みに頼らなければならない.」という文で始まる文章を処理する場合を考えよう．この文章を読んだ後，DUALS-III に「人間はどこで生きていますか.」という質問をしてみる．「地球に生きている」などというのが期待される答であろうが，上記の第1文から得られる情報からこの答を論理的に演繹することはできない．にもかかわらず，DUALS-III は「地球に生きている」と答えることができる．いかにしてこの答に至るかを以下で述べる．

初め，先頭節（の否定）のなかには下のような要素制約がある．（他にもあるが全部は示さない．以下同様に適宜省略する.）

(8) Sit＝ground
(9) hold (Sit, earth)
(10) Listen (Objs0)

DUALS-III の意味表現は状況意味論 (Situation Semantics)[1] および状況理論 (Situation Theory)[2] に基づいている．Sit は現実に対応する**状況** (situation) であり，この状況に，earth（「地球」の指示対象）などの**個物** (individual) が存在し，**事実** (facts) が成立する．Sit が ground に束縛してあるのは，他の対象と無闇に単一化させないためである．対象 ϕ が状況 ζ に存在することも，事実 ϕ が状況 ζ で成立することも，ともに要素制約 hold (ζ, ϕ) で表現する（このように「もの」と「こと」を統一的に扱っている点は元祖状況理論と異なる）．

まず DUALS-III は，入力された上記の第1文を読み込んで解析する．この処理は，上の要素制約 listen (Objs0) の仮説コストが高く，したがってポテンシャルが他の要素制約に比べて十分高いことによって生ずる．述語 listen は手続きとして定義されており，これを処理するということは，文解析モジュールを呼び出すということに相当する．この処理によって，listen (Objs0) が次のような要素制約群に変換される．

(11) hold (Sit, Purpose)
(12) hold (Sit 1, Continue)
(13) presuppose (Sit 1, Sit 3)
(14) hold (Sit 3, Loc 1)
(15) hold (Sit 3, Live)
(16) Purpose＝infon (purpose, [Sit 1, Sit 2], yes)
(17) Continue＝infon (continue, [Sit 3], yes)
(18) Loc 1＝infon (loc, [earth, Live], yes)
(19) Live＝infon (live, [human], yes)
(20) Sit 1＝sit (Sit, Purpose, 1)
(21) Sit 2＝sit (Sit, Purpose, 2)
(22) Sit 3＝sit (Sit 1, Continue, 1)
(23) Listen (Objs1)

Sit 1, Sit 2 および Sit 3は現実の状況 Sit よりも下位の，すなわち抽象的な状況である．これらの要素制約の間の関係を**図-2** に示す．

このとき，DUALS-III に対して「人間はどこで生きていますか.」という質問をすると，同様に(23)が処理され，下のような要素制約に変換される．

(24) hold (Sit, Loc 2)
(25) hold (Sit, Live)
(26) Loc 2＝infon (loc, [Where, Live], yes)

述語 presuppose は，下のように定義されているとする．

```
(27) presuppose(S 1, S 2) :-sub_sit(S 2, S),
                            project(S 1, S).
     project(sit(S 1, Infon, ArgPlace), S) :-
```

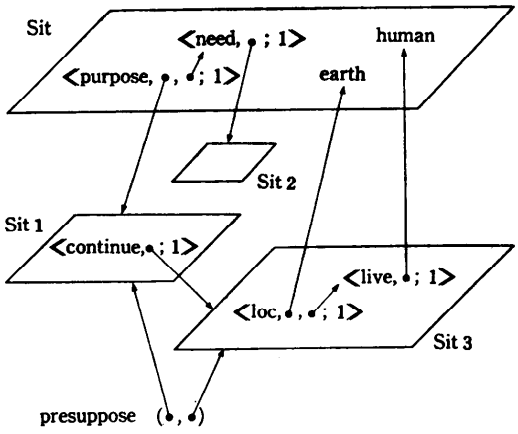

図-2 「人間がこの地球の上で生き続けてゆくためには,…」に対応する意味構造

project (S 1, S).
project (S, S).

ここで sub_sit (S 2, S) は，S 2 が S の部分状況であること，すなわち，hold (S 2, σ) なるすべての σ に関して，σ のコピー τ が hold (S, τ) を満たすことを意味する．ただし，σ が構造をもつ場合，その構造中に hold (S 2, α) なる α が現れるならば，τ の構造のなかの対応する位置には α のコピーが現れるものとする．(13)は，Sit 3 が Sit 1 または Sit の部分状況であるという意味である．この要素制約は，動詞「続ける」に付随する前提 (presupposition) であり，「続ける」の次のような語彙項目から得られる．

(28) lexicon ("続け", Category) :-
⋮
index (Category, Fact),
situation (Category, Sit),
subcat (Category, [Complement]),
situation (Complement, Sit 0),
Fact = infon (continue, [Sit 0], Pol),
hold (Sit, Fact),
presuppose (Sit, Sit 0).

前記のように，質問の命題内容にあたる要素制約 (ここでは(11)～(22)) と主張の命題内容にあたる要素制約 (ここでは(24)～(26)) との単一化が起動される可能性が高い．この例の場合には，(18)と(26)を単一化することができる．しかし，(24)と(25)をただちに他の要素制約と単一化させることはできない．たとえば，Sit と Sit 3 とは単一化不可能なので，Loc 1 = Loc 2 だからといって(14)と(24)を単一化することはできない．ここで(24)および(25)からのエネルギの伝播を考慮すると，これらに繋がったさまざまな要素制約が活性化されていると考えられる．こうして，(13)と(22)の間の依存関係のポテンシャルが高まる．この依存関係がもっともポテンシャルが高いとすると，(13)が展開され，sub-sit (Sit 1, Sit 3) が実行されて，以下のような要素制約ができる．

(29) hold (S, Loc 1)
(30) hold (S, Live)

上述のとおり，(13)は Sit 3 が Sit または Sit 1 の部分状況だということであるが，どちらであるかはそれだけでは分からない．しかし，もしも Sit 3 が Sit の部分状況であるとすれば，上の二つの要素制約をそれぞれ(24)および(25)と単一化することにより，ポテンシャルを下げることができる．一方，もし Sit 3 が Sit 1 の部分状況だとすれば，(24)と(25)のポテンシャルを下げることはできない．したがって，前者のほうが全体としてポテンシャルが低くなるので，より望ましいことになり，前者が選択される．こうして，DUALS-III は「地球に生きている」という答を返すことができる．(この間，実際には計算の途上でいくつかの新しい述語が定義されたのだが，繁雑なので詳細には触れない．)

この推論過程は，部分情報問題を扱ううえで重要な意義をもついくつかの処理の側面を実現している．第1に，これは一種のデフォルト推論である．すなわち，ポテンシャルが低いほうの選択肢をデフォルトとして選んでいる．DUALS の現在の版では，上の例題を処理する場合に，推論速度を上げるため，選択しなかったほうの可能性は単に捨てているが，それを捨てずに記憶しておくこともできる．それによって，いわゆる非単調推論などを行うことも可能である．デフォルト推論は不完全な情報によって判断を下す際に，非単調推論はそうして下した判断を棄却して信念を改変するのに必須である．

第2に，活性拡散によって，当面の問題に無関係な知識の部分が正しく無視されている点に注意されたい．上記の説明は無関係な知識には触れていないが，たとえテキスト中の文を大量に読み込んで上記の質問には無関係な知識が大量に蓄えられた後でも，DUALS は，以前とほぼ同じ応答時間で同じ質問に答える．すなわち，大量の無関係な知識を単に無視することができる．

第3に，上の例において会話の含意 (conversational

implicature) が処理されている．つまり，質問応答において，DUALS-III が質問者に情報を提供するのみならず，質問文が暗黙のうちに含意している内容がDUALS-III の知識に逆に流れ込んでいる．その含意とは，DUALS-III が質問の答を知っている，というものであり，質問の内容に関する要素制約(24)～(26)が DUALS の信念に直接付加され，しかも仮設コストが高いという状態に対応する．すなわち，この状態は，それらの要素制約を仮設コストの低い他の要素制約と単一化することによってポテンシャルを下げるべきだということを意味し，これは，その質問の答を導出しうるべきだという会話の含意とみなすことができる．

ここで重要なのは，依存伝播が情報の流れる方向を可能性としては限定していないということである．この性質により，部分情報問題において生ずる多様な情報の流れを実現することができる．たとえば，質問文から情報を得ることを禁じてはいないということによって，上の会話の含意の処理が自然に生じている．このような処理は，従来の手続き型の方法論においては明示的なプログラミングを必要とする．これは，制約プログラミングが談話理解のような部分情報問題を扱ううえで見通しのよい方法を与えることを例証している．

4. おわりに

談話理解実験システム DUALS-III，特に，談話処理において中心的な役割を果たしている，問題解決モジュールについて述べた．まず，談話理解のような部分情報問題を扱うためにはある種の制約プログラミングが必要であることを指摘し，そのようなプログラミングの方法として，依存伝播という計算理論を提示した．次に，それが DUALS-III において談話の処理にどのように用いられているかを解説し，制約パラダイムによって談話理解に必要なさまざまの処理が自然に実現されることを示した．

しかし，問題解決モジュールにおいて制約パラダイムを用いるというだけでは，情報の部分性に対処するうえではまだまだ不十分である．DUALS に限らず，既存の自然言語処理システムはすべて，基本的には手続きパラダイムに基づいているため，文解析，推論，文生成など，それぞれの処理について，専用の特別な知識(文法・辞書など)や計算機構を用いている．前にも指摘したように，こうしたシステムは，情報伝達の多様性を犠牲にしてシステム全体のモジュラリティを保つか，あるいは多様な情報伝達を実現しようとしてモジュラリティを維持できなくなり，プログラムがいたずらに複雑化するか，いずれかの状態にある．後者の方向を押し進めてもプログラムの複雑さがたちまち開発したり保守したりできる範囲を超えてしまい，いずれにせよ，実現すべき情報伝達を完全に実現するのは不可能である．たとえば，構文解析の途中で意味的情報を参照しようとする場合，実際にはどのような意味情報を参照すればよいかが文脈によって変化するから，本気で手続き的にこれをやろうとすると，構文解析をやっているのやら意味処理をやっているのやら分からなくなる．すなわち，制約のモジュール構造をプログラムに反映できず，システムがモジュラリティを失って複雑化し，人間の手に負えなくなってしまう．

プログラムを人間が管理できるものにするためには，制約のモジュラリティをプログラムの構造に反映させねばならないということと，自然言語処理のような部分情報問題においてそれをやるためには制約プログラミングが必須だということは，談話処理のなかで問題解決や照応に限った話ではない．統語的制約や形態的制約を含む制約全体にわたるモジュラリティをプログラムに反映させるには，文解析や文生成も含め，自然言語処理全体を制約パラダイムによって統一的に実現する必要がある．DUALS の次の版である DUALS-IV においては，すべての制約を依存伝播によって統一的に処理することを目指している．

制約パラダイムは，各文脈において処理の手順が予測し難いほど情報伝達の多様性が大きいような問題を扱う際にその力を発揮する．しかし，このような問題領域においては，たとえば実行過程のトレースをみてもそれを人間が簡単には理解できないなどの事情により，従来のデバッグの方法がそのままでは使えない．これは談話処理のような問題を扱う際の情報処理そのものに内在する困難であり，仮に手続き型プログラミングによって同様の情報処理を実現した（そんなことが可能とは思えないが）としても，同じ問題に直面することに変わりはない．いずれにせよ，談話処理のような問題を計算機で本格的に扱ううえでは，処理方式や制約の内容のみならず，効率的なデバッグの方法を求めることも重要な課題であり，DUALS に関しては，それも含めて談話理解の研究を進めてゆく予定である．

参考文献

1) Barwise, J. and Perry, J.: Situations and Attitudes, MIT Press (1983).
2) Barwise, J. and Etchemendy, J.: The Lier: An Essay on Truth and Circular Propositions, MIT Press (1987).
3) Colmerauer, A.: An Introduction to Prolog III, unpublished manuscript (1987).
4) Dincbas, M., Simonis, H. and Van Hentenryck, P.: Solving a Cutting-Stock Problem in Constraint Logic Programming, Proceedings of the 5th International Conference of Logic Programming, pp. 42-58 (1988).
5) Hasida, K., Ishizaki, S. and Isahara, H.: A Connectionist Approach to the Generation of Abstracts, in Kempen, Gerard (ed.), Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics, Martinus Nijhoff Publishers (Kluwer Academic Publishers), pp. 149-156 (1987).
6) Hasida, K. and Ishizaki, S.: Dependency Propagation: A Unified Theory of Sentence Comprehension and Generation, Proceedings of the 10th IJCAI, pp. 664-670 (1987).
7) 橋田浩一, 白井英俊: 条件付単一化, コンピュータ・ソフトウェア, Vol. 3, pp. 28-38 (1986).
8) 橋田浩一: 依存伝播, 第29回プログラミング・シンポジウム論文集, pp. 147-158 (1988).
9) Hobbs, J., Stickel, M., Martin, P. and Edwards, D.: Interpretation as Abduction, Proceedings of the 26th Annual Meeting of ACL, pp. 95-103 (1988).
10) Sakai, K. and Sato, Y.: Boolean Gröbner Bases, ICOT Technical Memo No. 488 (1988).
11) Sugimura, R., Hasida, K., Akasaka, K., Hatano, K., Kubo, Y., Okunishi, T. and Takizuka, T.: A Software Environment for Research into Discourse Understanding Systems, Proceedings of the FGCS '88, pp. 285-295 (1988).
12) Sussman, G. and Steele, G., Jr.: Constraints—A Language for Expressing Almost Hierarchical Descriptions, Artificial Intelligence, Vol. 14 (1980).
13) Tamaki, H. and Sato, T.: Unfold/Fold Transformation of Logic Programs, Proceedings of the Second International Conference on Logic Programming, pp. 127-138 (1984).
14) Tomabechi, H. and Tomita, M.: Application of the Direct Memory Access Paradigm to NL Interfaces, Proceedings of the 12th COLING, pp. 661-666 (1988).
15) Tuda, H., Hasida, K. and Sirai, H.: JPSG Parser on Constraint Logic Programming, Proceedings of the European Chapter of ACL '89 (1989).
16) Waltz, D. and Pollack, J.: Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation, Cognitive Science, Vol. 9, pp. 51-74 (1985).
17) Ward, N.: Issues in Word Choice, Proceedings of the 12th COLING, pp. 726-731 (1988).

(平成元年7月3日受付)