論理型文法における制約解析

Constraint Analysis of Logic Grammars

杉村領一

Sugimura Ryôichi

(財)新世代コンピュータ技術開発機構

Institute for New Generation Computer Technology

ABSTRACT

This paper presents a tiny model to connect constraints on logic grammars like DCG with basic constraints in discourse. Then, we can handle disjunctive interpretations derived from natural language analysis on logic grammars in constraint logic programming. To put in the concrete, the model presented here enables us to obtain logical core constraints between disjunctive interpretations explicitly in propositional logic. Consequently, the major advantage of our model is that we can propagate logical constraints between disjunctive interpretations on discourse analysis. The computational complexity of disambiguation is improved from Π number of disjunctions to Σ number of disjunctions. Practically, a structure which has solved with logical variables can easily be shared with disjunctive interpretations, the problem in structure sharing.

1. はじめに

自然言語の解析において、文脈(談話)情報を用いずに、一文の解釈を一意に決定することは困難である。なぜなら、一つの文は種々の談話状況で異なる解釈を持つ[Barwise 83]ためである。しかし、一文の処理と談話処理を結び付ける事も難かしい。そこで、従来は文解釈内において、種々のヒューリスティクスにより一文の解釈を決定している[Nakamura 86]。

本論は、この曖昧さを、談話において扱うための基本的な枠組みを提供する。つまりDCG等の論理型文法による一文の処理と談話処理を曖昧さの処理を通じて結び付ける一手法を提案する。本手法は、文解析の複数の解釈間に成り立つ制約を制約論理プログラミングによって談話処理へ伝搬させ[Dincbas 86]、伝搬した制約と談話における制約により解釈を順次決定する事を可能にする。

本論は以下の3つの部分から構成される。

まず、2.で論理型文法の制約と談話構造の基本的な制約を定式化し、関係づける。

3以下では、2.で示した制約を用いて、文解析を行う具体的な方法を示す。

3.では、一文の解析処理の果たすべき役割を示す。具体的には、同一の部分木を重複して作らない構文解析を行ない、複数の解釈間の制約を生成する。現在、構文解析にはSAX[Matsumoto 87]を用いているが、これは本質的ではない。同一ゴールの重複計算の無い解析が、インプリメントにおいて、解釈間の制約を有効にするための必要条件である。

4.では、談話処理について述べる。具体的には、一文の解析において得られた複数の解釈間の制約と、談話構造の基本的な制約を、制約論理プログラミングによって解く。制約の解決方法としては2つの手法を示す。

第一の手法は、遅延実行制御による制約の受動的な解法である。本手法では、制約式の書き換えを能動的には行わない。

第二の手法は、能動的に、制約式の書き換えを行う。制約式の論理値が決定できない場合には、この制約を以降の文脈処理へ伝搬する。第二の手法は、項変換プログラムで実行可能である。第二の手法によれば、多環境における論理変数共有の問題[Matsumoto 87]は解決される。計算量等についても考察を行う。

2.文文法と談話の関係

2.1.論理型文法

DCG等の論理型文法では、解析は、文法として与えられた推論規則と入力文から、述語"文"を証明する証明過程と考える事が出来る。解析過程で得られた種々の部分木は、入力文と文法から得られる定理である。先ず、推論規則は以下の例のようになっている。

$$h(S_h,A,A_n) \Leftarrow b1(S_1,A,A_1) \wedge \ldots \wedge bn(S_n,A_{n-1},A_n) \wedge \quad (1)$$
$$rel(S_h,(S_1,\ldots,S_n)) \quad (2)$$
$$b_j(S_j,A,B) \Leftarrow c(A,B) \wedge B = A + 1. \quad (3)$$

(1)(2)は文法として予め与えられている推論規則である。論理変数の$S_h, S_j(1 \leq j \leq n)$は、それぞれの述語h, bj$(1 \leq j \leq n)$を真とする意味情報を示す変数である。論理変数のA, $A_j(1 \leq j \leq n)$は、それぞれの述語h, bj$(1 \leq j \leq n)$を真とする文中の単語並びの範囲を示す変数である。ただし、各単語には自然数が文の左端の単語から順に対応させられているとする。例えば、$h(S_h,A,A_n)$は、位置AからA_nの手前までをhとして解釈できる事を示す。

(2)の述語$rel(S_h,(S_1,\ldots,S_n))$は、引数$S_1,\ldots,S_n$と、$S_h$の間に何らかの関係が成り立つ場合に真となる。

(3)の$b_j(S_j,A,B)$は辞書として与えられる述語である。$c(A,B) \wedge B = A + 1$は、入力文によって、その真偽が決定される。cは表層を示す。

2.2. 複数の解釈と文脈

文解析とは、上記の推論規則(文法と辞書)を用いて入力文(長さをLとする)で与えられた解釈から証明できる、述語"文($S_文$,1,L+1)"を得る事だと考えよう。一般に、入力文1つに対して、複数の述語"文"が証明できる。複数の解釈を文脈で扱うために、述語"文"については、上記の文法へ以下のように下線で示した述語を追加する。

$$文(S_文,A,A_n) \wedge \underline{S_文 \in I} \Leftarrow b1(S_1,A,A_1) \wedge ... \wedge bn(S_n,A_{n-1},A_n) \wedge rel(S_文,(S_1,...,S_n))$$

Iは述語"文"を真とする意味情報を要素とする集合である。

次に、意味情報を談話を考慮したものに拡張する。文内でも、例えば敬語などのように、談話状況が異なる[杉村 86]。そこで、意味情報を談話を示すパラメータと意味情報の組で示す。例えば、形態的に明らかに談話が変わる場合には、文法規則は以下の様に示される。

$$h((DS_b,S_h),A,A_2) \Leftarrow b1((DS_a,S_1),A,A_1) \wedge b2((DS_b,S_2),A_1,A_2) \wedge rel((DS_b,S_h),((DS_a,S_1),(DS_b,S_2)))$$

上記の例では、述語b1とb2の間で談話状況がDS_aからDS_bへ切り替わり、全体としては、談話状況DS_bになる事を示している。

また、辞書規則については、発話において、cなる単語があり、かつ、この状況から"c"の意味S_jが得られるので、以下のように規則を記述する。

$$h((DS,S_j),A,B) \wedge rel((DS,S_j),(DU,c)) \wedge DU \models c \Leftarrow c(A,B) \wedge B = A + 1.$$

また、cは単語として発話に存在しているので、DUを発話状況とすると、$DU \models c$が成り立つ。

2.3. 談話の制約

2.1,2.2で示した規則に従い文の解釈を行ったとする。すると、述語$\underline{S_文 \in I}$に付いては、以下の制約が成立する。

公理D1

$$\exists DS \models S \text{ where } (DS,S) \underline{\in I}$$

この制約は非常に重要である。つまり、文は最低一つの意味を文脈で持つ($\models$)ということを、この制約は述べている。この制約は、例えば集合Iに2つの要素がある場合には、以下の様に書き下せる。

$$(DS \models S_1) \vee (DS \models S_2)$$

次に、本論で重要な位置をしめる推論規則を示す。

公理D2

$$(DS \models S_h) \Rightarrow (DS_1 \models S_1) \wedge ... \wedge (DS_n \models S_n) \text{ where } rel(S_h,((DS_1,S_1),...,(DS_n,S_n)))$$

公理D2は、ある状況DSである意味が成り立ち、この意味が他の状況で成り立つ部分意味から構成される場合、この部分意味もある状況で成り立つ事を示す。直感的には、文の意味が談話状況で成立するかどうかは、その部分意味がある状況で成立するかどうかを調べればよい事を示している。

ただし、この公理の意図は、述語論理の域をはみ出ている。というのは、例えば、動詞「走る」が一つの引数「誰か」を持つ述語だとすると、「誰か」が「走る」ことが成立するかを決定するには、引数の決定されない(つまり、適当な引数への値の割り当てが決まっていない)述語「走る」が成り立つかどうかを調べねばならず、この決定を、適当な引数への値の割り当てが決まっていなくても行えるという立場であるためである。これは、状況理論の基本的な立場である。

以上の公理D1,D2と解析のための推論規則を用いて如何に、文が処理されるかを、3以降で述べる。

3.構文解析

構文解析は同一の部分木を重複して生成しない方法で行う。これは、計算機上で同一の述語が複数真であると、無意味な曖昧さが述語"文"について作られるためである。まず、以下のようにSTを構文解析中にできた部分木(定理)の集合とする。

$$ST = \{Sti \mid Stiはiをインデクスとする部分木\}$$

必要な条件は以下の通りである。

$$\forall i \forall j (Sti = Stj \Rightarrow i = j) \text{ where } Sti \in ST \wedge Stj \in ST$$

同一の部分木を重複して生成しない構文解析は、論理型構文解析システムSAX、BUP[Matsumoto 84]、LangLAB[今野 86]等で既に実現されている。

4. 談話における制約処理

以上、3で示した条件を満たす構文解析手法により、例えば英文、"You saw Jon on that"を、2で示した形式の文法と辞書を用いて解析し、図1の様な解析結果を得たとする。また、構文解析で成り立つ述語を、(XY)で以下の規則を用いて簡略に示す。

$$(DX \models Y) \Leftrightarrow YX$$

例えば、$(DS \models S_{16})$は$S_{16}S$と示す。

結果としては、以下の述語が公理D1、公理D2により、文脈内部で成立つ。

公理D1より

$(DS \models S_{13}) \vee (DS \models S_{15}) \vee (DS \models S_{17}) \vee (DS \models S_{19})$

公理D2より

$(DS \models S_{13}) \Rightarrow (DS \models S_2) \wedge (DS \models S_{12})$
$(DS \models S_2) \Rightarrow (DS \models S_1)$
$(DS \models S_{12}) \Rightarrow (DS \models S_3) \wedge (DS \models S_{11})$
$(DS \models S_{11}) \Rightarrow (DS \models S_6) \wedge (DS \models S_{10})$
$(DS \models S_6) \Rightarrow (DS \models S_5)$
$(DS \models S_{10}) \Rightarrow (DS \models S_9)$
$(DS \models S_9) \Rightarrow (DS \models S_7) \wedge (DS \models S_8)$
$(DS \models S_{15}) \Rightarrow (DS \models S_2) \wedge (DS \models S_{14})$
$(DS \models S_{14}) \Rightarrow (DS \models S_4) \wedge (DS \models S_{11})$
$(DS \models S_{17}) \Rightarrow (DS \models S_2) \wedge (DS \models S_{16})$
$(DS \models S_{16}) \Rightarrow (DS \models S_3) \wedge (DS \models S_6) \wedge (DS \models S_{10})$
$(DS \models S_{19}) \Rightarrow (DS \models S_2) \wedge (DS \models S_{18})$
$(DS \models S_{18}) \Rightarrow (DS \models S_4) \wedge (DS \models S_6) \wedge (DS \models S_{10})$

発話より

$(DS \models S_1) \Rightarrow (DU \models You)$

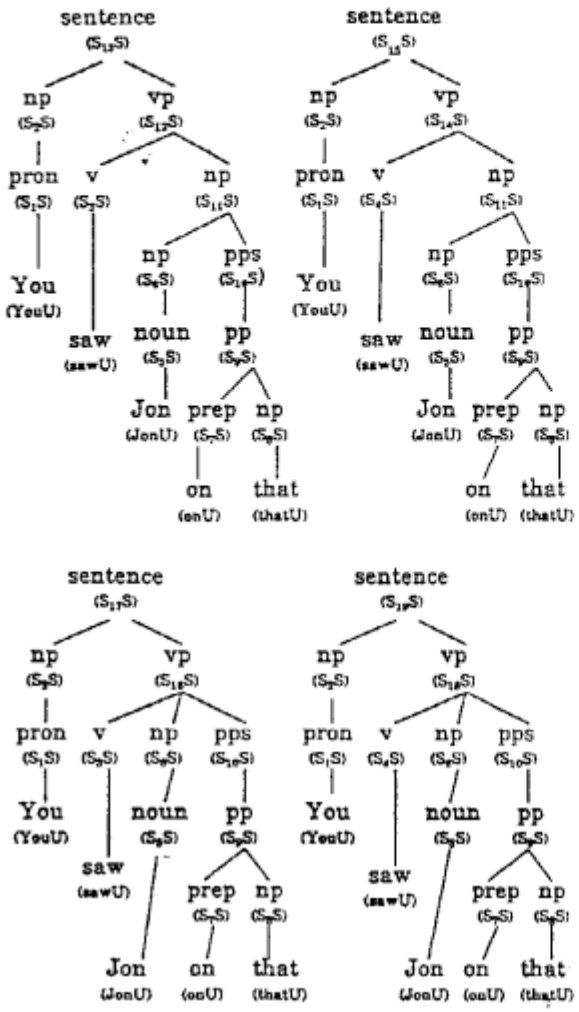

図1 "You saw Jon on that"

$(DS \models S_3) \Rightarrow (DU \models saw)$
$(DS \models S_4) \Rightarrow (DU \models saw)$
$(DS \models S_5) \Rightarrow (DU \models Jon)$
$(DS \models S_7) \Rightarrow (DU \models on)$
$(DS \models S_8) \Rightarrow (DU \models that)$
$DU \models You \quad DU \models saw \quad DU \models Jon$
$DU \models on \quad DU \models that$

以下、上記制約の処理方法について述べる。

まず、以降の議論では、上記の各述語を図1で用いた簡略化により、例えば一つの述語$DU \models that$をthatUとして示される一つの論理変数として用いる。これにより、構文解析で得られた、談話における制約は全て以下の様に命題論理式として扱える。

$(S_{13}S \vee S_{15}) \vee S_{17} \vee S_{19}S$

$S_{13}S$	$\Rightarrow S_2S \wedge S_{12}S$	S_2S	$\Rightarrow S_1S$
$S_{12}S$	$\Rightarrow S_3S \wedge S_{11}S$	$S_{11}S$	$\Rightarrow S_6S \wedge S_{10}S$
S_6S	$\Rightarrow S_5S$	$S_{10}S$	$\Rightarrow S_9S$
S_9S	$\Rightarrow S_7S \wedge S_8S$	$S_{15}S$	$\Rightarrow S_2S \wedge S_{14}S$
$S_{14}S$	$\Rightarrow S_4S \wedge S_{11}S$	$S_{17}S$	$\Rightarrow S_2S \wedge S_{16}S$
$S_{16}S$	$\Rightarrow S_3S \wedge S_6S \wedge S_{10}S$	$S_{19}S$	$\Rightarrow S_2S \wedge S_{18}S$
$S_{18}S$	$\Rightarrow S_4S \wedge S_6S \wedge S_{10}S$		
S_1S	$\Rightarrow YouU$	S_3S	$\Rightarrow sawU$
S_4S	$\Rightarrow sawU$	S_5S	$\Rightarrow JonU$
S_7S	$\Rightarrow onU$	S_8S	$\Rightarrow thatU$

YouU sawU JonU onU thatU

(1)受動的な制約処理

第一の手法は、遅延実行制御による制約の受動的な解法である。本手法では、制約式の書き換えを能動的には行わない。

まず、論理型言語CIL[Mukai 87]で実現されている遅延実行制御による命題論理の制約解決機構について説明する。表1にCILの真理値表を示す。Uは値の定まらない変数である。例えば、CILでは以下のように命題論理の制約を解決する。

表1 CIL真理値表

∧	1	0	U	∨	1	0	U	¬	
1	1	0	U	1	1	1	1	1	0
0	0	0	0	0	1	0	U	0	1
U	U	0	U	U	1	U	U	U	U

> constr(and(A, B), false), A = true.
A = true, B = false,
> constr(and(A, B), false).
A = Unbound, B = Unbound

$A \Rightarrow B$は、$\neg A \vee B$としてCILで定義されている。

以上の機能を用いて、上記命題論理で表現された制約を定義できる。

ただし、上記の公理1から得られる制約、つまり、$S_{13}S \vee S_{15}S \vee S_{17}S \vee S_{19}S = T$からは、何ら論理的な帰結は自動的には得られない。ここに、受動的な解法と、能動的な解法の異なりが現れる。ただし、例えば$DS \models S_3$が偽であることが分かれば、$S_{13}S \vee S_{17}S = F$となる。

(2)能動的な制約処理

前記の命題論理の制約の解法には種々の手法が適応

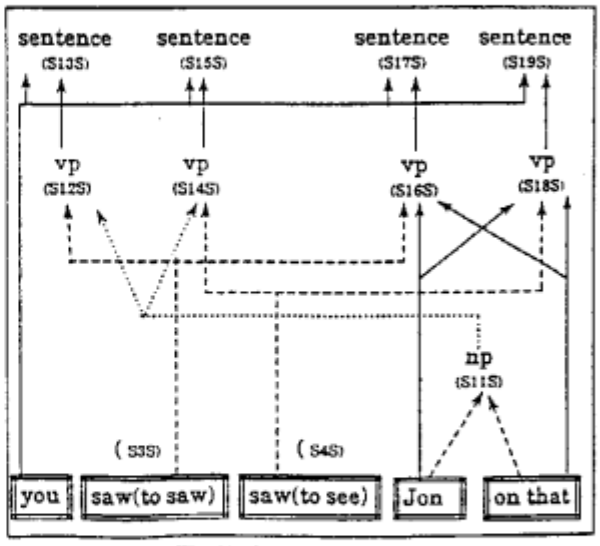

図2 談話上の論理的帰結

可能である。

第1の手法としては、命題論理式の各論理変数に総当たり的に値を割り当てる手法があるが、これは、計算量が$O(2^n)$(nは変数の数)になるので、用いたくない。

第2の手法は、上記の命題論理の制約が、Prologと同様のクローズのみから与えられている事を利用する。例えば、以下の用なプログラムで実行可能である。

```
go(V,U):-
    assert(':-'(s,s13s)), assert(':-'(s,s15s)),
    assert(':-'(s,s17s)),assert(':-'(s,s19s)),
    assert(youU),assert(sawU),assert(jonU),
    assert(onU),assert(':-'(s1s,youU),
    assert(':-'(s3s,sawU)),assert(':-'(s4s,sawU),
    assert(':-'(s5s,jonU)),assert(':-'(s8s,thatU)),
    assert(':-'(s13s,(s2S,s12s))),assert(':-'(s2s,s1s)),
    <中略>
    go1([s1S,s2,s3,s4,s5,...,s16s,s17s,s18s,s19s],V,U).
go1([],[],[]).
go1([H|R],V,U) :-retract(':-'(H,B)),
    (s, assert(':-'(H,B)),U=[H|UR],V=VR
    ;assert(H),V=[H|VR],U=UR),go1(R,VR,UR).
```

本プログラム実行により、述語goの第1引数Vへ真となる論理変数の名前が入り、第2引数Uへ値の決まらない論理変数の名前が入る。本手法の欠点は、例えばs2s⇒s1sでs2s=真のとき、自動的にs1s=真とできない点である。そこで、上記のassert部分をCILの命題論理制約記述に置き換える方法が、インプリメントでは有効である。計算量は詳細は省くがPrologの場合$O(n\times m)$(mは命題論理式の総数)で、CILの場合$O(n\times m')(m'\leqq m)$である。現在の所、本手法を我々は用いている。

第3の手法はブーリアングレブナーベースのソルバを用いる手法である。本手法でも計算時間は$O(n\times m)$になる。詳細は[Sakai 88]を参照されたい。

つぎに、解かれた制約の意味を考察する。

命題論理の制約を解くと、以下の帰結が新たに得られる。値の決まらない変数間の関係は図2へ示す。

$$(DS\models S_1)\wedge(DS\models S_2)\wedge(DS\models S_5)\wedge(DS\models S_6)\wedge$$
$$(DS\models S_7)\wedge(DS\models S_8)\wedge(DS\models S_9)\wedge(DS\models S_{10})=\text{真}$$

①値の決まった命題論理値に対応する述語

文文法の制約と、公理D1,D2だけから、値が決まる事に注意して欲しい。談話処理は、これら値の真となる命題論理値に対応した述語を真とするように談話を形成する必要がある。また、対応した述語内部の意味構成を行っても多重環境による論理変数共有は起こらない。例えば、SAXでは以下の様に意味構成を行っている。makeはボトムアップに意味を組み上げる関数である。

$$rel(S,(S_1,...,S_n))\wedge S\Rightarrow S=make(S_1,...,S_n)$$

②値の決まらない命題論理値に対応する述語

談話処理において、順次これらの値を決定する必要がある。どのように、決定するか詳細な議論は別途報告したいが、おおまかには、以下のようになる。

(A)語彙の曖昧さについて

例えば、$((DS\models S_3)\Rightarrow(DU\models saw))\wedge((DS\models S_4)\Rightarrow(DU\models saw))\wedge(DU\models saw)$より、$(DS\models S_3)\vee(DS\models S_4)$となるが、談話がどの言語で営まれ、どの分野の談話であるかなどの情報を用いて、$(DU\models saw)\Rightarrow\neg(DS\models S_3)$とする方法が有る。従来の分野情報とは論理的にはこのDUのレヴェルの規則を指す。$\neg(DS\models S_3)$が分かれば、$DS\models S_{13}$と$DS\models S_{17}$は偽になる。

(B)談話依存の曖昧さについて

本枠組みではDUのレヴェルから順に図2の上方向に各述語の真偽を決定してゆける。例えば、図2で$S_{11}S$が偽である事が分かると、$S_{17}S$と$S_{19}S$が残り、$S_{13}S$と$S_{15}S$は消え去る。$S_{11}S$が偽である事を導くには、談話状況の間の規則を研究して行く必要がある。談話状況の間の規則については、まだまだ多くの課題があるが、最低限以下の公理系を用いても良いと考えている[Fenstad 85]。

$$DS\models A\wedge B \quad\Leftrightarrow (DS\models A)\text{ and }(DS\models B)$$
$$DS\models A\vee B \quad\Leftrightarrow (DS\models A)\text{ or }(DS\models B)$$
$$not(DS\models A) \quad\Leftrightarrow (DS\models\neg A)$$

5. おわりに

文解析と談話構造の関係を曖昧さの処理の観点から整理した。今後は、本手法をベースに談話構造の研究を行って行きたい。

[謝辞]

本研究の機会を下さいましたICOT渕所長、内田第2研究室室長、吉岡2研究室室長代理に感謝致します。

[参考文献]

[Barwise 83] Barwise, J. and Perry, J., Situations and attitudes, MIT Press, Cambridge, 1983

[Dincbas 86] Dincbas, M., Constraints, Logic Programming and Deductive Databases, Proceedings of France-Japan Artificial Intelligence and Computer Science Symposium 86, pp. 1-27, ICOT, 1986

[Fenstad 85] Fenstad, J.E., Halvorsen, P.K., Langholm, T., Benthem J., Equations, Schemata and Situations: A framework for linguistic semantics, CSLI Report No. CSLI-85-29, 1985

[Kaplan 82] Kaplan, R.M., & Bresnan, J., Lexical-Functional Grammar: A Formal System for Grammatical Representation, in Bresnan, J., ed., The Mental Representation of Grammatical Relations, MIT press series on cognitive theory and mental representation, 1982

[今 野 86] Konno, S. and Tanaka, H., Processing Left-extraposition in Bottom-up Parsing System, Computer Software Vol.3 No.2 (Tokyo, 1986) pp. 19-29

[Matsumoto 84] Matsumoto, Y., Kiyono, M. and Tanaka, H., Facilities of the BUP Parsing System, Proc. of 1st International Workshop on Natural Language Understanding and Logic Programming, Rennes, 1984

[Matsumoto 87] Matsumoto, Y., and Sugimura, R., A Parsing System based on Logic Programming, IJCAI 87

[Mukai 87] Mukai, K., A System of Logic Programming for Linguistic Analysis Based on Situation Semantics, Proceedings of the Workshop on Semantic Issues in Human and Computer Languages, Half Moon Bay, CSLI, 1987

[Nakamura 86] Nakamura, J., Solutions for Problems of MT Parser – Methods Used in Mu-Machine Translation Project, Coling'86, pp.133-135, 1986

[Sakai 88] Sakai, K, Sato Y., Boolean Gröbner Bases, ICOT Technical Memo No.488, 1988

[杉村 86] 杉村領一, 日本語の時間表現と状況意味論, ソフトウェア科学会, Vol.3 No.4, pp46-54, 1986,

[Sugimura 87] Sugimura, R.,Miyoshi,H. Mukai,K., Constraint analysis on Japanese Modification, Second international Workshop on Natural Language Understanding and Logic Programming,1987