

2 S-6

単一化文法の並列処理の一提案

菅井 猛 溝口文雄
(東京理科大学 理工学部)

1. はじめに

GPSGのような单一化文法[1]を扱うようなパーザを作ることは、大量のデータを扱わなければならない。PAF/SFG(Paralell Analsysis For Situation and Feature Grammar)[2]では、構文的な処理だけでなく、状況意味論[3]に基づく、意味的なデータ構造も扱かっているので、素性構造の組み合せ的な爆発が起こりかねない。この解決策として、並列処理という点からの考察を行なう。この時、並列処理を行なうと、单一化文法のユニフィケーション[4]がどのようなものがものが適当であるかの検討を行なう。使用言語は、单一化文法と相性がいいことと並列処理を扱えるということから、並列論理型言語GHCを用いて、その言語の枠組み内でインプリメントしてある。

2. 単一化文法による並列処理

PAF/SFGでは、構文解析のデータフローにレイヤードストリームを採用しており、素性構造はパーザの中に埋め込まれた形となっている。図1にPAF/SFGの動詞の終端記号の文法の記述例を示す。図1の規則はトランストレータによって、図2のような、二分木の生成述語を含むをDCG形式に変換される。これは、内部的には、図3のように素性構造の二分木として埋め込まれた形になっている。

```

verb
+ type:subob      + sem:love
+ subj>gr:subj    + subj>sem:Subj
+ obj>gr:obj      + obj>pcase:wo
+ obj>sem:Obj     + obj2>sem:0
+ obj2>gr:obj2    + obj2>sem:0

+ subj>in>sit:Sit + subj>in>loc:Loc
+ obj>in>sit:Sit  + obj>in>loc:Loc

+ in>sit:Sit      + in:loc:Loc
+ in>pol:Pol
- subcat:[sub,obj]
* anchor(love,subj,Subj)
* anchor(love,obj,Obj)
* in(Sit,Loc,[love,subj,obj],Pol)

-->[愛す].

```

図1：終端記号の記述例

図1の+オペレータはローカル素性、*オペレータはグローバル素性を示している。ローカル素性とは、カテゴリそれぞれが固有に有する素性である。一方、グローバル素性は、すべてのカテゴリが共有する素性で

あり、どのレベルからも素性の検索が可能になっている。この時、それぞれの句構造規則に対し、素性集合の処理を行なわねばならなく、大きな計算量を要する。

句構造規則内の素性構造は、その句構造の中で、変数を含まない順序付きの二分木に変換される。この二分木への変換は、素性構造を探索する際に、探索空間を削減するものである。

subcat[sub, ob]は、下位範疇化の原則を表したものである。この規則は、M→D Hにおける句構造について、「Mのsubcatの値は、Hのsubcatの値から、Dと等しいものを取り除いたものに一致する」という制約で表せる。この他に、HFC(Head Feature Convention), CAP(Control Agreement Principle), FFP(Foot feature Principle)という单一化の原則を句構造規則中に埋め込んでいる。

統語的あいまいさは、素性構造の（二分木）の組み合せによって表せる。この時、部分木が、ボトムアップに作られながら、部分的な値の割り当てが成立する。

```

word(愛す,In,Out) :- true !
make_obi([type:subob, sem:love,
          subj>gr:subj,           subj>sem:var(c),
          obj>gr:obj,             obj>pcase:wo,
          obj>sem:var(d)          obj2>gr:obj2,
          obj2>sem:0,              obj2>sem:0,
          subj>in>sit:var(e),     subj>in>loc:var(f),
          obj>in>sit:var(e),     obj>in>loc:var(f),
          in>sit:var(e),          in>pol:var(g)],H),
          in>pol:var(g)],H),
          verb(In,H,[sub,obj],verb(愛す),
          [situation(anchor(love,subj,var(c))),
          situation(anchor(love,obj,var(d))),
          situation(anchor(var(e),var(f),[love,subj,obj]),
          var(g))),Out].

```

図2：トランストレータによって変換されたもの

3. 値の割り当ての実現

GHCでは、変数の具体化の方向性が決まっており、組み込み述語をそのまま使うことはできない。また、ある文脈として、1文1文をとられた場合、文脈間にまたがる共有変数の扱わなければならない。そこで、次のような解決策をとる。

図1の大文字で始まるSubj, Obj, Loc, Polなどは変数であり、共有変数によって素性構造の整合性を保っている。図2は、句構造規則内の共有変数が、同じグランドアトムとなっている。二分木への変換では、

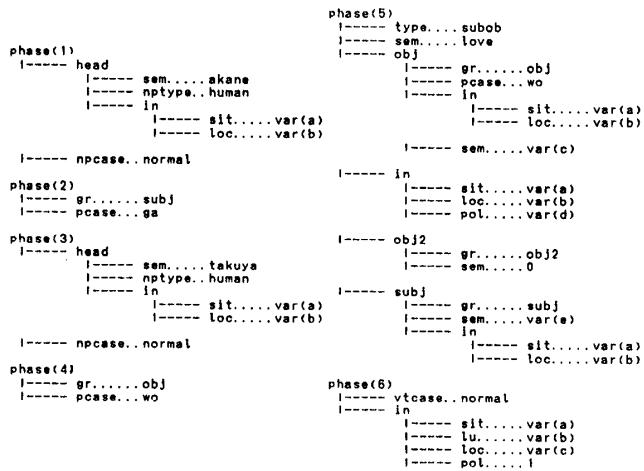


図3：二分木の構造

句構造文法規則内の素性集合の変数を $\text{var}(1)$, $\text{var}(2)$, $\text{var}(3), \dots$ のように、グランドアトムで表現されるように変換する。これにより、部分的に情報が付加された不定項がすべてグランドアトムで表現することができる。例えば、図1の大文字で始まるSubj, Obj, Loc, Polなどは変数であり、共有変数によって素性構造の整合性を保っている。これらの変数は[5][6]の中の考え方を句構造規則用に書き直したものである。これは、句構造規則内の変数をグランドアトムとして考え、グローバルな変数の管理を行なっている。

```

value_assign([value(11)=lu(1),var(7)=var(12),var(9)=1,
             value(10)=var(8),var(7)=var(3),var(8)=var(4),
             var(6)=akane,var(7)=var(1),var(8)=var(2),
             var(5)=takuya],
            []
           [in(var(1),var(2),[present,male,takuya],1),
            in(var(3),var(4),[present,female,akane],1),
            anchor(love,subj,var(5)),
            anchor(love,obj,var(6)),
            in(var(7),var(8),[love,subj,obj],var(9)),
            loc(t_overlap,var(10),var(11))]) - Output.

Output = [in(var(12),var(4),[present,male,takuya],1),
          in(var(12),var(4),[present,female,akane],1),
          anchor(love,subj,takuya),
          anchor(love,obj,akane),
          in(var(12),var(4),[love,subj,obj],1),
          loc(t_overlap,var(4),lu(1))].

```

図4：値の割り当ての例

図4は最終的な値の割り当てを定義する述語を示す。この述語の第一引数は、グローバルな変数の割り当てを表すリスト、第二、第三引数は、ローカルな素性集合とその出力結果（ここでは空リスト）、第四引数は全体の素性集合、第五引数は値の割り当ての出力結果となる。これによって、`anchor/3`, `in/4`にグローバルな変数の値が伝わり、全体の解釈が定まる。図4のOutputは、値の割り当てが終了したときの出力結果であ

る。

表1は、JPSGパーザの部分と状況生成のための値の割り当ての部分との解析結果である。なお、JPSGパーザでは、解析木と全体の素性集合が出力される。この結果から、状況生成の値の割り当ての方が約3倍程度、大きくなっている。

表1：状況の生成のための値の割り当ての結果

形態素数	Red.	Sus.	Cyc.
文番号① 6	3 0 5 2 9 6 3 4	4 3 3 2 9 1 3	9 3 2 3 9
文番号② 8	4 2 0 1 1 7 9 4 5	4 7 8 5 4 5 2	1 0 8 3 0 2
文番号③ 12	2 0 1 3 6 7 7 8 6 1	1 5 6 3 2 3 8 1 7	2 2 1 5 7 1

(上段: JPSGのパーザ、下段: 状況生成のための値の割り当ての例)

(Red., Sus., Cyc. はそれぞれ、リダクション数、サスペンド数、サイクル数を示す)

[例文]

①ヒロミがタクヤを訪ねた。

②タロウが文明の先端を味わう。

③タクヤはヒロシにアカネがプロポーズされたことを教えた。

4.まとめ

単一化に基づく句構造を並列に実行する方法とそれによる変数管理の方法を示した。これにより大規模な素性集合の記述も可能になる。これからは、さらに効率のよい変数管理の方法が課題である。今後は、この点を改善していきたい。

参考文献

- [1]Stuart M. Shieber: An Introduction to Unification-Based Approaches to Grammar, Center for Study of Language and Information, (1986)
- [2]伊藤達雄, 溝口文雄: 並列論理型言語による意味解析(Parallel Analysis for SFG), 情報処理学会第38回全国大会, 6D-7(1989), pp289-290
- [3]Jon Barwise : The Situation and Logic, Center for Study of Language and Information, (1989)
- [4]Kevin Knight : Unification ,A Multidisciplinary Survey, ACM Computing Surveys, Vol. 21 , No. 1, (1989)
- [5]A. Martelli and U. Montanari:An Efficient Unification Algorithm, ACM Trans. on Programming Languages and Systems, Vol4, No2, (1982)
- [6]J. W. Lloyd:Directions for Meta-Programming, Proc. of International Conference on Fifth Generation Computer Systems, pp. 609-617, 1988