

並列論理型言語による意味解析

- PAF/SFG: Parallel Analysis For Situation and Feature Grammar -

6D-7

伊藤 達雄 溝口 文雄  
(東京理科大学 理工学部)

1. はじめに

構文論としてのGPSG (Generalized Phrase Structure Grammar)[3]と、意味論としての状況意味論 (Situation Semantics)[2]を統合した自然言語の文法を記述する言語SFG (Situation and Feature Grammar)[1,6]を部分木の局所性に着目して、並列に効率よく解析する方法(PAF/SFG: Parallel Analysis For SFG)を提案する。GPSGでは本来、意味論として内包論理を用いているが、同じ言葉が状況によって異なった解釈となること(言葉の効率性)を説明することができないため、SFGでは状況意味論を用いている。各々の語い項目に持つ特有の情報(素性)が、句構造規則によって子から親へ伝播される。その規則に対応する部分木において、素性構造の単一化が親カテゴリ生成の制約として働くが、規則の局所性より、その処理が独立に、並列に実行することができる[4]。この単一化によって意味を生成することから、意味解析が並列に処理される。現在、PAF/SFGは並列論理型言語GHC上でインプリメントされている。

2. SFGにおけるGPSGと状況意味論の融合

GPSGにおける意味論、内包論理式は素性の具現化(feature instantiation)によって合成される。これに対し、状況意味論は言葉の外延の意味を重視する考えを発展させたもので、第一階述語論理による外延的モデル理論の発展形である。またその解釈は、(可能)世界にその言語の表現の外延を与えることによって関係付ける。つまり状況意味論では、その引数にあたる割り当て(assignment)を具体化することによって事態σが作られる。

そこでSFG記述では、s σをin(s, σ)と表し、グローバル素性として\*オペレータで非終端記号に付加し、不定要素にあたる論理変数によって、+オペレータで示されたローカル素性(ルール)を関係付けておく(アンカー)。よって構文解析中にGPSGの素性の具現化によって素性が具現化されたとき、事態の割り当ても具現化される。つまり構文・意味情報、発話状況、記述状況を素性として捉え、宣言的に記述できる。

```
v + subj>ind:X + loc:Loc
+ obj>ind:X + sit:Sit
* in(Sit, (Loc, (love, X, Y), 1)) --> [愛す].
```

図1: SFG記述

3. 意味生成の並列性

意味は素性の具現化によって素性構造を作る際に生成されていく。素性の具現化は部分木ごとの処理であるため、部分木の持つ独立性から並列に処理することが可能である。実際には、意味を形成する事態はすべて

で終端記号に記されているため、非終端記号となる部分木の処理は句構造に対して素性の具現化を行うだけでよい。そしてロールに対応する素性が具現化されたなら、事態の割り当てでも具現化される。つまり意味処理は部分木ごとに独立に、並列に実行され、部分的に次々と解釈が作られていく。

4. 句構造規則の並列処理

部分木の並列性から、句構造文法のカテゴリである素性集合を並列に動くプロセスとして捉えると、PAX[5]と同様のアルゴリズムを用いることが可能となる。この方法では、プロセスが並列に動くことや、一度生まれたプロセスは必要なデータをすべて読み終えるまで生き続け、再計算をすることが無いため無駄なプロセスやデータが生成されないという利点がある。そこでPAF/SFGでは、各々のカテゴリは素性構造を持ち、素性の具現化におけるユニフィケーションがプロセス生成の制約として働く。

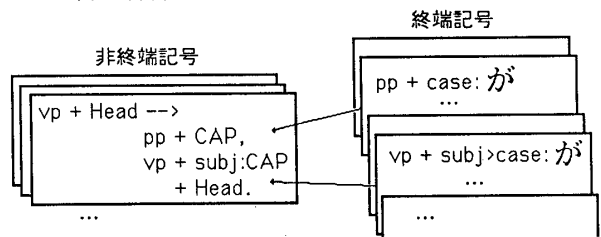


図2: SFG記述によるCAP制約の例

ところで、句構造文法のようなユニフィケーションベースの文法は、主辞(Head)と他のカテゴリのユニフィケーションという概念を中心に捕らえている。しかしGHCでは、双方向のユニフィケーションができないため、これらのユニフィケーションに方向性をもたせることで、制約と具現化を行っている。

Head Feature Conventionは主辞である娘から母親へ、Control Agreement Principleは主辞以外の娘カテゴリから主辞へ、Subcat Feature Principleは娘から母親へ、という方向性を持つとすることによって、句構造を生成することが可能となる。SFG規則の素性処理は、次のようにGHCプログラムに変換される。

主辞以外の娘カテゴリは、素性構造から素性値の取り出し(get\_obt)、主辞は素性値を素性構造に置けるか判定(put\_obt)とする。また素性値の定まっているチェックは制約として働くため、素性構造に対するその素性値の判定とする。

以上のように単一化を素性構造の受渡しに置き換え、GHC上で実現している。ここでカテゴリ生成の制約として働くのはput\_obt述語であり、すべての制約が充足されたときのみ、新しいプロセスを生成する。

```

vp2( [ pp( In, LF, Env) | Tail], LF1, Env1, Out) :-
    true !
    get_obt(all:CAP),
    put_obt(subj:CAP, LF1, LF2, Env2, true, J),
    goal(J, vp, In, LF2, Env2, Outg),
    vp2(Tail, LF1, Env1, OutT),
    merges([Env, Env1, Env2], Env3),
    merge(Outg, OutT, Out).
    
```

図3:変換された制約処理プロセス

このユニフィケーションによって素性の値を具現化し、意味生成を行う。ところが、論理変数を用いているためor並列におけるor分岐点で、他の枝にまで影響を与えてしまう(環境のコピーの問題)。そこでストリームによる環境の分割を用いる。

5. ストリームによる環境の分割

新しいプロセス(母親カテゴリ)は、別の環境で処理されなければならない。そこで、プロセス(カテゴリ)ごとにそれぞれ別の環境を与え、ユニフィケーションの際に、変数をそのまま具体化せずに、変数と束縛する値との対として解析木の中に流し、or分岐点ではストリームを分割する。そして最終的に文の解析が終了したとき、それぞれの持つストリームが各々の環境となる。

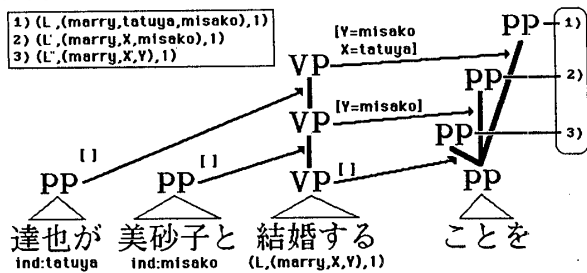


図4:ストリームによる環境の分割

6. 日本語の解析例

例題:「美砂子が私が泣いたと言った」

上文を係り受けのみで考えると、後置詞句「美砂子が」と「私が」は「泣いた」と「言った」という2通りの動詞句に各々係る可能性があるが、まず係り受けが交差する事はないので  $2 \times 2 - 1 = 3$  通りの構文解析木が考えられる。さらに動詞が主語を1つしか持たないことを考慮にいれると1通りに絞られる。しかしながら「私」という単語が発話者と同定することから、最終的に1)美砂子が泣いたということを美砂子が言った、2)発話者が泣いたということを美砂子が言った、という2通りの解釈が得られる。

```

NP + speaker:X
+ ind:X --> [私]
    
```

図5:SFGによる同定の例

このような文を PAF/SFG では、全ての可能な構文解析木を得てから解釈を絞るのではなく、無駄なプロセスを生成しないように、効率よく解析することが可能である。まず句構造規則に従った解析では、構文的な係り受けは絶対に交差せず、またSUBCAT素性の原理により、適当な係り受けしか認めない。し

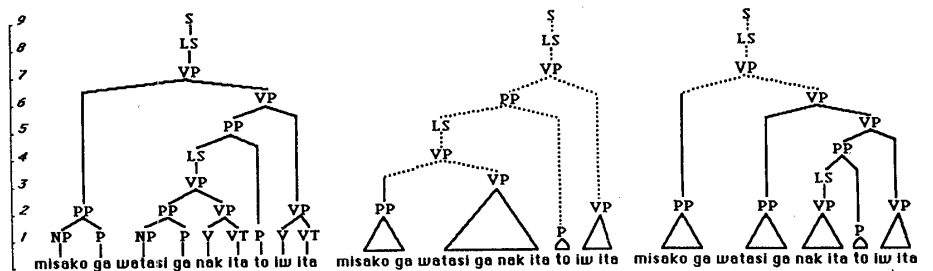


図6:探索空間の減少例

かも句構造規則の適応は部分木ごとに行われるため、独立した部分木の処理は独立に、並列に実行される。つまりPAF/SFGでは単一化による制約によって、探索空間を絞り、2通りの解釈を持った1つの正しい解析木のみを得る事ができる。この例の場合、部分木の数が20あるため、逐次型処理系では最低20ステップ必要となるが、並列に部分木を処理するため、9ステップしか必要とされない。しかも素性による制約によって、点線の部分木は生成されないため、PAXに比べ探索空間も小さくなる(図6)。なお、△部の解析はレイードストリームにより、共有されている。

以上のようにPAF/SFGは、GPSGの利点と並列処理の利点の両方を兼ね備えた自然言語解析を行ない、さらにその解釈が状況意味論の型で取り出せる。

```

in(A, (B, (t_precede, C, B), 1))
1) in(A, (C, (t_precede, F, C), 1))
   in(A, (C, (say, misako, cry(misako), D), 1))
   in(A, (C, (refer, E, misako), 1))
   in(A, (F, (cry, misako), 1))
   -----
2) in(A, (B, (t_precede, C, B), 1))
   in(A, (C, (t_precede, F, C), 1))
   in(A, (C, (say, misako, cry(D), E), 1))
   in(A, (C, (refer, D, misako), 1))
   in(A, (F, (cry, D), 1))
   -----
    
```

図7:解析結果

7. おわりに

並列論理型言語上で、GPSGと状況意味論を統合することによって、意味解析を並列に実行する方法PAF/SFGを考案した。その際、カテゴリを並列に動くプロセスとしたことで、PAXのアルゴリズムをそのまま用いることができ、また環境を分割することによって多環境を表すことができた。

統語的情報や意味的情報や状況を素性として表わし、制約として並列に処理することによって探索空間を小さくしている。これによって効率よく解釈を取り出す自然言語解析が可能となった。

参考文献

[1]安食, 溝口: 状況と素性具現化に基づく自然言語解析, ソフトウェア科学会 第2回大会論文集, 2A-1, 1985.  
 [2]Barwise, J. and Perry, J.: Situation and Attitudes, MIT Press, 1983.  
 [3]Gazdar, G.: Phrase Structure Grammar, in P. Jacobson and G. Pullum, The Nature of Syntactic Representation, pp. 131-186, D. Reidel, Dordrecht, 1982.  
 [4]伊藤, 溝口: 句構造文法を用いた並列構文解析, 日本ソフトウェア科学会第5回大会論文集, A2-3, 1988.  
 [5]Matsumoto, Y.: A Parallel Parsing System for Natural Language Analysis, to be presented at Third Int. Logic Programming Conf., London(1986).  
 [6]長瀬, 溝口: 知識表現言語SCOREを用いた談話理解システム, 情報処理学会第34回全国大会, 3W-4, 1987.