

4F-9

単一化に基づく構文解析における制約の選択的適用*

田代敏久 永田昌明†
ATR 自動翻訳電話研究所 ‡

1 はじめに

本報告では、素性構造で記述された制約を選択的に適用することにより単一化パーザを高速化する方法について述べる。文法記述には、無矛盾性の検査に関する記述と、情報の併合に用いられる記述がある。単一化結果は制約の適用順序に依存しないことを利用し、解析実行時に無矛盾性検査に関する制約を選択的に適用することにより、宣言的な文法記述の枠組みを保ったまま構文解析と意味解釈を分離する実験を行い、結果を報告する。

2 単一化に基づく構文解析の特徴と問題点

単一化演算は交換則(commutative)と結合則(associative)が成立する演算であるので、単一化結果は制約の適用順序に依存しない。また単一化は単調(monotonic)な演算なので、ある構成要素で成立している情報は、より大きな構造の一部となっても常に成立する。また、単一化に基づく構文解析では、統語的・意味的・語用論的信息をすべて統一的、宣言的に記述できるため、文法のモジュール性・保守性が高いという利点がある。

しかし、単一化演算は計算コストのかかるプロセスであるため、単一化に基づく構文解析には実行効率に問題がある。そこで単一化パーザの効率を上げることを目的として、従来以下のような試みがなされてきた。

- 単一化アルゴリズムの改良。単一化演算の中でも特に処理時間がかかる素性構造の複製を減らす [1]、選言的素性記述の単一化のアルゴリズムを改良する [2]、等の試みがある。
- 文法記述の改良。選言ができるだけ早く解消されるように規則を書く [3]、文法の保守性を損なわない範囲で句構造規則を詳細化し単一化の回数を減らす [4]、等の提案がある。
- CFG パーザと単一化プロセスとの関係の見直し。詳細化した句構造規則を用いた場合、素性記述を遅延評価することにより無駄な単一化が回避できる [4] との報告がある。

本研究では、CFG パーザと単一化プロセスとの関係の見直しによる実行効率の向上の研究をさらに深め、文法記述を分割し、一部の記述を遅延評価することにより、構文解析の高速化を目指す。

*A selective constraint application method in unification-based parsing

†Toshihisa TASHIRO, Masaaki NAGATA

‡ATR Interpreting Telephony Research Laboratories

```
(deflex-named 登録用紙 -1 登録用紙 n-com
  (<!m>==[[syn [[head [[pos n]
                    [form 登録用紙]]]]]
          [SEM [[PARM ?x]
                [RESTR [[ENTITY ?x]
                        [RELN 登録用紙 -1]]]]]
          [PRAG [[RESTR !EMPTY-DLIST]]]
          [semf [[anim -][time -][loc -][conc +]
                 [event -][act -] [state -]]]])
(大文字による記述は意味解釈用記述)
```

図 1: 文法記述 (辞書記述) の例

3 文法情報の性質と単一化の成功率

文法を構成する情報は、次の 2 つに大きく分類できる。

- 品詞情報、活用情報、選択制限のための意味素性情報等の句構造を組み立てる際に無矛盾性を検査するために使われる情報 (構文的制約の記述)。
- 意味情報、語用論情報等の、語彙記述に書かれた部分的情報を併合していくことにより、文 (解析の単位) 全体の情報を合成するために使われる情報 (意味解釈の記述)。

構文的制約の記述は、単一化に失敗する率が高く、構文解析の曖昧性の縮小に貢献するのに対し、意味解釈の記述はほとんど単一化に失敗しない。このことから、統語構造作成時には構文的制約の記述のみを単一化することにより無駄な部分木が作成されるのを防ぎ、統語構造作成後、まとめて意味解釈の記述を単一化して出力構造を作成する (構文解析と意味解釈を分離することにより、全体の単一化の計算量の削減が可能である)。

4 構文解析と意味解釈の分離の実現

構文解析と意味解釈の分離を実現するためには、以下の 2 つの処理が必要になる。

- 素性記述の分割 (factoring)
- 分割された素性記述 (= 制約) の選択的適用 (selective constraint evaluation)

素性記述の分割する時期としては、1) ソースファイルを分割する、2) 文法コンパイル時に分割する、3) 素性構造作成後に分割する、等が考えられるが、実現の容易さから 2) の方法をとることとした。

文法コンパイラは、素性構造記述言語で書かれた文法記述ファイルを読み込み、解析機構内部のデータ構造 (ク

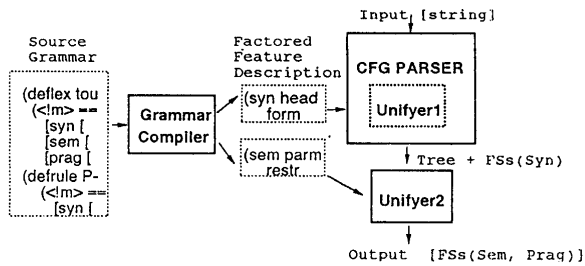


図 2: システム概要

ラフ構造)を作成する。この文法コンパイラ内部の文法記述を経路方程式の束に変換するモジュールに変更を加え、記述の分割を実現した。

一方、制約(素性記述)の評価方法としては、従来、1)句構造規則を適用するたびに評価する(早期評価戦略)、2)句構造規則のみを適用して統語構造を作成し、最後にまとめて評価する(最終評価戦略)、の2つの戦略が提案、実装されていた[4]。

しかし、この従来の戦略には、1)早期評価戦略は、無駄な部分木の生成を押さえるのにはあまり役立たない意味解釈の記述を解析途中で評価してしまう、2)最終評価戦略は、解析中は句構造規則の制約しか用いないため、無駄な部分木の生成を防ぎ切れない場合がある、という欠点がある。

そこで、ここで提案する選択的評価戦略では

- 句構造規則を適用するたびに、構文的制約の記述を評価し、無駄な部分木の生成を避ける。
- 統語構造が作成された後、意味解釈の記述を評価し、出力構造を作成する。

ことにより、従来の戦略の欠点を補う。図2に構文解析のシステムの概要を示す。

5 実験および考察

制約の選択的適用の有効性を確認するために、次のような解析実験を行なった。まず種々の素性記述評価戦略の比較を行なうために、a)早期評価戦略(以下、early-modeとする)、b)最終評価戦略(以下、late-modeとする)、c)選択的評価戦略(以下、selective-modeとする)の3つの戦略を用いて同一のテスト文を解析し、解析時間を測定した。また、文法の性質の違いによる効果の変化を見るために、1)語彙数約700、文法規則数約160で、接尾辞等の詳細な解釈は行なわない文法(以下、文法1とする)と、2)語彙数約1200、文法規則数約180で、接尾辞等を細かく解釈する文法(以下、文法2とする)の2種類の文法を用意した。

解析対象は、「国際会議の申し込み」をタスクとする5つのサンプル会話の中から無作為に抽出した6文(平均13文字)である。パーザはCommon Lispでインプリメントされている。解析時間の比較を表1、2に示す。

文法1を用いた場合、early-modeと比較すると約2倍の高速化を達成しているが、late-modeには若干劣っ

表 1: 解析時間の比較(文法1)

単一化評価モード	early	late	selective
解析時間の比(early = 100)	100.0	50.9	53.1

表 2: 解析時間の比較(文法2)

単一化評価モード	early	late	selective
解析時間の比(early = 100)	100.0	93.1	55.5

ている。これは、1)文法1が句構造規則の詳細化等の人手による最適化が進んでいるため、句構造規則だけで充分無駄な部分木の作成を防ぐ能力がある、2)selective-modeでは出力構造を得るために、2回グラフ構造を作成する必要があり、これがオーバーヘッドとなっている、ことが原因と思われる。

一方、文法2の場合は、文法1と比較すると、1)語彙数の増加により、解析中に曖昧性が発生しやすい、2)接尾辞等を細かく解析するため、評価すべき素性記述の量が多い、といった特徴がある。そのため、late-modeを用いた場合、句構造規則のみでは多くの無駄な部分木が解析途中で発生してしまい、early-modeに対する優位性が損なわれている。これに対し、selective-modeを用いた場合、解析中に必要最低限の文法記述が評価されているため、無駄な部分木の発生は抑制され、late-modeと比較して約1.7倍の高速化が達成できた。

実験結果から判断して、この制約の選択的適用は、文法の規模が大きい場合に特に効果があるといえる。文法が小規模な場合には、熟練した文法開発者が句構造規則を詳細化したうえでlate-modeを用いることにより、充分効率を高められる。しかし、文法が大規模化するにつれ、人手による最適化には限度があるので、自動的(機械的)な最適化手法が必要になると予測される。その場合、この制約の選択的適用のような、静的に記述された知識の効率的な適用方法の研究が重要になるとと思われる。

6 おわりに

本報告では、制約の選択的適用により、単一化に基づく構文解析の効率を向上させる手法を提案し、これを実験により示した。

参考文献

- [1] Tomabechi, H., "Quasi-Destructive Graph Unification with Structure-Sharing," Proc of COLING-92, 1992
- [2] Kasper, R. "A Unification Method for Disjunctive Feature Descriptions", in Proc of ACL-87, 1987
- [3] 永田・久米・小暮: 単一化に基づく枠組みにおける日本語対話文解析用文法の記述とその計算的側面, NL 研資, 90-NL-76-1, 1990.
- [4] Nagata, N. "An Empirical Study on Rule Granularity and Unification Interleaving. - Toward an Efficient Unification-Based Parsing System.", in Proc. of COLING-92, 1992.