

依存構造解析のための SAX のデバッグ環境

福本文代 赤坂宏二

弘田直人† 伊藤哲†

(財) 新世代コンピュータ技術開発機構 日本情報科学研究所 (株)†

1 はじめに

構文意味解析システム SAX は、ICOT がこれまでの自然言語処理研究で得られた要素技術の 1 つであり、DCG(Definite Clause Grammars)で記述された文法を基に上昇型の解析を行なう構文意味解析システムである [1]。SAX は本来 2 型と呼ばれる記述形式を用いた文法を基に解析を行なうが、日本語の係り受け関係のための書き換え規則を宣言的に記述することを目的に 0 型を含めて解析できるように拡張されている。本稿ではこの拡張記述形式を用いて実現された係り受け解析の、デバッグ環境について述べる。

2 SAX 上での依存構造解析

日本語を解析する手法としては句構造および依存構造があるが、句構造は SAX で扱う文法が論理型文法であることから、実装は容易に行なえる。そしてそのためのデバッグ環境もすでに用意されている。一方依存構造を論理型文法で実現する場合には、係り側から受け側への依存関係が認識された後も、なお両者が係り受け関係の書き換え対象として残しておく必要がある。そこで SAX の文法記述形式は 0 型、つまり文法規則の左辺に任意個のカテゴリーを記述することができるように拡張されている [2]。以下にこの拡張形式を用いて記述された依存関係のための係り受け規則の例を示す。

```
word(S11,M1,C1,P1),bar(M3,C3,P3),
  word(S22,M2,C2,P2)
→ word(S1,M1,C1,P1), word(S2,M2,C2,P2)
{ ... check(S1,S2,S11,S22,M1,M2,M3,
           C1,C2,C3,P1,P2,P3), ... }
```

S1: 依存関係成立前の係り側の構文情報
 S2: 依存関係成立前の受け側の構文情報
 S11: 依存関係成立後の係り側の構文情報
 S22: 依存関係成立後の受け側の構文情報
 C1,C2,C3: 等位関係を示す情報
 M1,M2,M3: 依存関係を示す情報
 P1,P2,P3: 優先度得点を示す情報

図 1: 係り受け文法規則

A debugging tool based on a dependency structure analysis built around the SAX system
 Fumio FUKUMOTO Konji AKASAKA Naoto HIROTA †
 Satoshi ITO †
 ICOT ISL †

図 1 は、隣り合う文節の左側の文節が右側の文節を修飾する場合に対応した書き換え規則である。規則中、左辺の第 1 カテゴリーの word は係り側、第 2 カテゴリーの word は受け側の文節を表す。また規則中 bar のもつ引き数には依存関係が成立した時、どの文節がどの文節に係っているかを示す情報および依存関係の尤もらしさを表す優先度得点が蓄積される。

3 依存構造解析のためのデバッグ環境

本文法では、

- 図 1 の係り受け規則より、文法規則が適用されたときに新たなノードカテゴリーは生成されない (図 1 の規則中 bar のもつ引き数には依存関係成立後、どの文節がどの文節に係っているかを示す情報が蓄積されているため、ここでは新たに生成されたノードカテゴリーとは見なさない)
- 文法規則が適用された後、本文法では、どの文節がどの文節に係っているかを示す情報がカテゴリーの引き数で表現されることから、この引き数を参照することにより依存関係を示すアークを表現している

という 2 点に特徴がある。そこでこの点に着目し、従来の SAX のデバッグ環境を利用し新たに依存構造解析のための SAX のデバッグ環境を開発した。

4 デバッグ環境の実現

従来の SAX のデバッグ環境は、大きく 2 つ、すなわち解の探索履歴 (ストリーム) を用いた静的デバッグ環境と、解析実行をトレースする動的デバッグ環境から構成されていた [3]。ところが文法開発の面から見ると、トレース途中では、それまでに構成された構文木を視覚的に見ることができないため、文法開発の効率の面で問題があった。そこで今回、依存構造解析のためのデバッグ環境の開発に先立ち、両環境を一つにまとめ、トレース時に、その時点までの解の探索履歴を用いて、すでに解析の終了している部分についての構文木の構成、表示、ストリームのインスペクトなどの静的デバッグが行なえるようにした。依存構造解析のためのデバッグ環境は、この統合環境を用いて開発されている。具体的には、解析のトレース機能に依存関係の情報を視覚化するビジュ

アル表示機能を加えることによって実現した。図2にデバッグウインドウを示す。

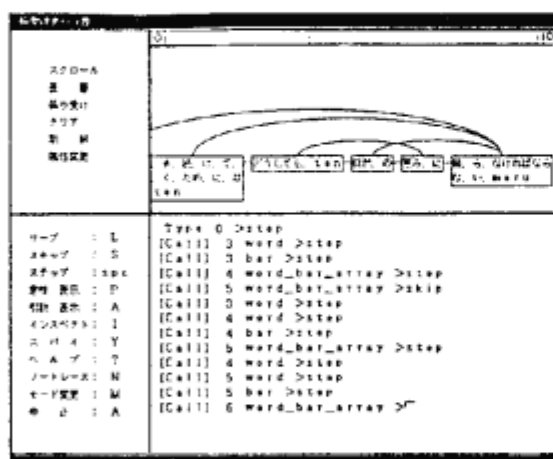


図 2: 依存関係解析のためのデバッグウインドウ

ウインドウは、

- 上部の依存関係表示用のグラフィックウインドウ
- 下部のトレースウインドウ

より成る。以下ではそれぞれについて述べる。

4.1 グラフィックウインドウ

グラフィックウインドウは、トレース途中でそれまでに構成された依存関係の情報を視覚化するものである。ウインドウには入力列が各列ごとにボックスで囲まれて表示（ボックス表示）される。ボックス表示は入力列の各カテゴリ名で行なうが、形態素解析の出力結果を入力とする場合は、変換文字列が表示される。構文意味解析結果の出力である依存関係の表示は、解析途中に係り受けに関係するカテゴリから生成された ESP メソッドの呼び出し時、終了時、および 0 型を用いて記述された文法規則の左辺の評価終了時にユーザの指示により、各カテゴリの持つ引数をもとに行なう。そして修飾関係は実線、等位関係は破線の円弧で表わされる。依存関係（係り受け）ウインドウで用意されているコマンドを以下に示す。

- スクロール … スクロールバーを用いて依存関係表示の左右スクロールを行なう。
- 表示 … 入力列のボックスをクリックすることにより、その表示構造を表示する。
- 依存関係 … リーシュカテゴリの引数情報から、修飾関係、等位関係を円弧で表示する。
- クリア … 円弧を全て画面から消去する。
- 別解 … 解析の出力結果が複数存在する場合、解

析終了時に、別解の係り受け関係を表示する。

属性変更 … 表示のフォントサイズ、およびボックスのサイズを変更する。

4.2 トレースウインドウ

トレースウインドウには、構文意味解析の過程が表示される。具体的には、各カテゴリから生成した ESP メソッドの呼び出し、終了、補強項の評価、文法規則の適用状況などが表示される。

トレースコマンドは制御用として、リープ、スキップ、ノートレース、中止がある。意味表示は意味構成用の引数情報をブリティブプリントする。また、引数表示は文法カテゴリから変換したメソッドの引数のうち、もとのカテゴリのもつ引数のみを表示する。一方インスペクトはストリームや差分リストを含む全ての引数を SIMPOS のインスペクタで表示する。スパイはスパイポイント設定メニューでスパイポイントの変更を行なうことができる。スパイエントリには、文法カテゴリ、文法規則、補強項、入力列（入力文字列）が指定できる。モード変更はモードメニューでトレース項目、リーシュ項目の指定および表示の深さや長さのレベルの変更を行なう。

5 おわりに

SAX 上での依存関係解析のためのデバッグ環境について述べた。今後は、実際の文法開発を通しての実証と同時に、トレース途中で単語列の範囲を指定することにより、範囲内での解析数および係り受け関係の表示ができるなど、随時拡張を行なっていく予定である。また本デバッグ環境は、トランスレート後に動作されるものであるが、トランスレート前の文法開発支援として、文法規則内に 1 度しか表れない変数のチェックおよび未使用のカテゴリチェック機能などを備えた支援環境の構築も検討していく予定である。

参考文献

- [1] Y. Matsumoto and R. Sugimura, "A parsing system based on Logic Programming", *Proceedings of the International Joint Conference of Artificial Intelligence*, 1987.
- [2] 杉村領一、福本文代, 「論理型文法による依存関係解析」 情報処理学会, 談話理解モデルとその応用シンポジウム, 89-100, 1989
- [3] 山崎重一郎、弘田直人、赤坂宏二, 「LTB における構文解析システム SAX について」 情報処理学会第 37 回全国大会, 1080-1081, 1988