論理型プログラミング環境における文法開発支援ツール

木下　聡　　佐野　洋
㈱東芝　総合研究所　　(財)新世代コンピュータ技術開発機構

1．はじめに

近年，自然言語処理応用システムの中には，機械翻訳システムのように，既に製品化の段階に達しているものもあり，処理の対象も従来のように教科書にのせられているような例文的な文から，マニュアルや科学技術論文などの実文に移ってきている．ところで，そのような実文を解析していくと，当然のことながら，解析できない文があるため，文法を修正する必要が生じることになる．しかしながら，文法の規模が大きくなると，その修正作業は極めて困難なものとなり，文法開発支援ツールが必要となってくる．

ところで，ＩＣＯＴでは，研究開発の効率化を目的として，「汎用日本語処理系（Language Tool Box 以下ＬＴＢと略す）」の開発が行われており，その中のソフトウェアの１つとして，構文解析システムＳＡＸ［松本86］がある．ＳＡＸでは，ＤＣＧで記述した文法規則をを論理型プログラムに変換して直接動作させることで，高速な解析が可能となっている．しかしながら，文法そのものを開発する環９という点から見ると，以下のような問題がある．

１）文法の数が増加した場合に文法規則を管理することが困難である．

２）処理が横型探索で行われるため，処理のトレースが困難であり，文法のデバッグが難しい．

３）文法規則の一部を修正した際に，それを用いて実験を行うまでの時間が長い．

本研究では，上記の問題を解決するための文法開発支援環境を開発することをその目的とし，文法管理ユーティリティーとデバッグ支援ユーティリティーを設計し，逐次型推論マシンＰＳＩ－Ⅱの上に実現した．

2．文法管理ユーティリティー

文法管理ユーティリティーは，文法規則のチェックや，文法規則の書換えを行うためのユーティリティーである．文法は，カテゴリー宣言部と文法本体から構成され，それぞれ別のファイルに定義される．カテゴリー宣言部は，文法本体で定義される文法規則中で使用される文法カテゴリーを宣言するためのもので，文法本体は，下の例に示すように，本来のＤＣＧ形式の文法規則に比べ，文法規則の左辺にルール番号が付加されている（この形式を以降「拡張ＤＣＧ形式」と呼ぶ）．

```
100:: sentence([sentence,NP,VP]) -->
      np(N1,NP), vp(V1,VP), !check1(N1,V1)|.
```

2．1　文法のチェック

指定された文法ファイル中の各文法規則のチェックを行う．チェックする項目には以下のものがある．

１）シンタックス
文法規則が定める形式にしたがって正しく記述されているかをチェックする．

２）文法カテゴリー
文法規則中に現れる文法カテゴリーが，カテゴリー宣言部で宣言されているものかチェックする．

３）文法規則中の変数
文法規則中に現れる変数の中で，ただ１カ所のみに現れる変数があるかをチェックする．

４）ルール番号
同一のルール番号が付与されている文法規則がないかチェックする．

チェックを行っている際にエラーが見つかった場合には，単に処理を中断したり，続行したりする他，エディターを呼び出し，直ちにエラーの修正を行い，その箇所から処理を再実行することも可能である．

2．2　文法規則の書換え

文法規則の書換え機能には次の２つがある．

①拡張ＤＣＧ形式で記述した文法を，指定された書換えパターンにしたがって書換える．②文法規則にルール番号を付与したり，付与してあるルール番号を削除する．

①は，ある条件を満足するような文法規則を，別の規則に機械的に書き替えて文法を修正する際に用いる．システムのエディターなどの文字列の置き換えなどでは対応できないような置き換えも可能である．②は，ＤＣＧ形式で記述されている文法を，拡張ＤＣＧ形式にしたり，その反対に，拡張ＤＣＧ形式で記述された文法をＤＣＧ形式にしたりするためのものである．

3．デバッグ支援ユーティリティー

１章で述べた（２）の問題を解決するために，縦型探索（depth first）の解析を行う実行環境を整備した．さらに，（３）の問題を解決するため，文法規則の追加や削除，修正が行えるようにした．

3．1　文法トランスレーター

文法トランスレーターは，拡張ＤＣＧ形式で記述された文法を，ボトムアップ・パーザーとして動作するＥＳＰのプログラムの形に変換する．このボトムアップ・パーザーはＢＵＰ［松本83］の初期のバージョンに相当する．この変換機能には，文法全体を一度に変換する一括変換のほか，文法規則単位での追加や削除，修正を行うことが可能である．

文法の部分的修正が可能となったことで，文法の修正を行ってから解析の実験を行うまでの所要時間は大幅に減少

Grammar Development Tool under Logic Programming Environment
Satoshi KINOSHITA and Hiroshi SANO
Toshiba Corporation, ICOT

している．約２００個の文法規則からなる文法を変換し，パーザーのプログラムとして解析に使用できる状態になるまでの時間は約１７０秒であったが，ルール１個を修正するのに要する時間は，わずか７秒に抑えられている．その結果，文法規則の追加，修正，削除などの作業に要する時間は，従来に比べ１／２４に減少したことになる．この比率は文法規則の数が増加すると，さらに大きくなるため，この効果は文法開発の上で極めて大きいといえる．

３．２　文法デバッガー

文法デバッガーは，線型探索のボトムアップ解析をトレースし，文法の不具合を発見するための環境を提供することを，その目的としている．

（１）　トレース・モード

トレースのモードとして，ライン(line)・モード，ビジュアル(visual)・モード，ノートレース(no trace)・モードの３つのモードを持つ．図１にビジュアル・モード時のデバッガーの画面を示す．画面には以下の情報が表示される．

ａ）現在までの解析で使用されている文法規則
- 一番下の規則が，現在パーザーによって適用されている文法規則である．
- 文法規則の右辺のカテゴリーは，そのカテゴリーに対応する構造が見つかるにしたがって，表示が反転する．

ｂ）木構造
- 解析途中の木構造が適宜表示される．
- 画面内に木構造をおさめるため，ある程度大きくなった構造は，簡略化して表示する（『構造のホロフラスティング表示機能』と呼ぶ）．

ｃ）処理の局面
- デバッグモデルにおける処理の局面（後述）を表示する．

ｄ）マウスメニュー

なお，ライン・モードでは，ＥＳＰのデバッガーの画面構成に準じ，処理の局面の表示とマウスメニューからなる．また，ノートレース・モードでは，解析が成功または失敗するまで一切トレース情報を表示しない．

（２）　デバッグモデル

文法デバッガーのデバッグモデルは，PrologのＢＯＸモデルをベースとし，次の特徴を持つ．
- 文法規則レベルで扱う．
- Left corner parsing による解析に適したトレース制御コマンドを利用者に提供する．

本デバッグモデルにおける処理の局面は基本的に２種類ある．まず第１は，解析の種となる左隅の文法カテゴリーが得られた時に，それをもとに１段上の構造を作ろうとする局面であり，これをEXPANDの局面と呼ぶ．この局面では，得られた文法カテゴリーが文法規則の右辺の最初のカテゴリーとなっている文法規則を捜す．もう一つは，上で得られた文法規則をもとに，右辺の残りのカテゴリーを１つ１つ順番に解析していく時の局面である．この局面をSEARCHの局面と呼ぶ．

EXPANDとSEARCHの局面は，PrologのＢＯＸモデルのCALLの局面に対応しており，EXIT，FAIL，REDOに対応する局面をそれぞれ持っている．

（３）　スパイポイントの設定

スパイポイントの設定方法は，ＥＳＰのデバッガーにおける方法に準ずる．本デバッガーでは，文法規則に対するスパイの他，文法カテゴリーに対してのスパイも可能である．

４．おわりに

本研究では，自然言語処理システムの中核である文解析システムのための文法の開発を支援するためのツールについて述べた．このツールは，論理型プログラミング環境の元で実現されており，

①記述した文法規則をチェックする機能
②文法の修正から解析実験までのターン・アラウンド・タイムの大幅な削減
③文法の不具合箇所を容易に発見するための豊富な実行のトレース機能

を実現した．今後このツールを活用して，実験で利用する文解析システムの文法を開発していくとともに，ツールの利用を通じてさらに使いやすいものにしていきたい．

参考文献

[松本83] 松本他：“ＢＵＰトランスレーター文法規則から構文解析プログラムの自動生成ー”，電子技術総合研究所彙報 Vol.47，No.8(1983)．

[松本86] 松本他：“論理型言語に基づく構文解析システムＳＡＸ”，コンピュータ・ソフトウェア Vol.3，No.4(1986)．

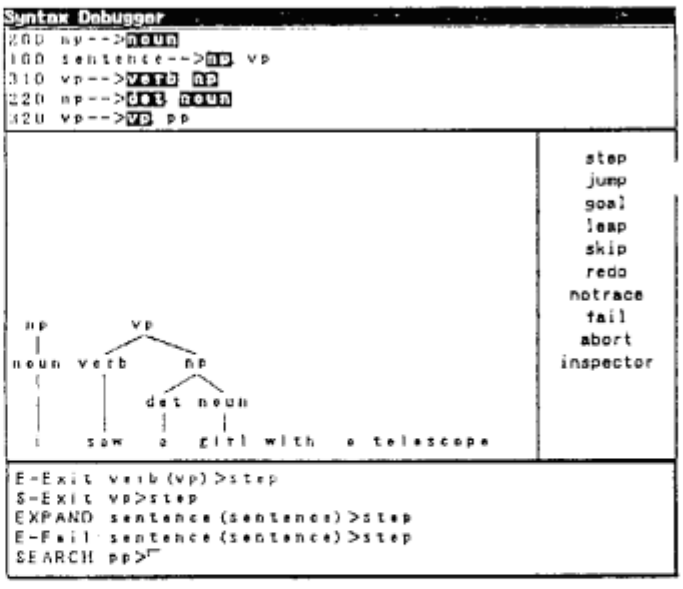

図１　文法デバッガーの画面構成（ビジュアル・モード時）