

L T B - 形態素解析システム L A X の開発環境

久保 幸弘 * 妙泉 正隆 佐野 洋 赤坂 宏二 杉村 頌一

(財)新世代コンピュータ技術開発機構 * (財)日本情報処理開発協会

1. はじめに

本稿では、汎用日本語処理系 L T B における形態素解析システム L A X の開発環境について述べる。L A X では正規言語と等価な形態素文法[1]を形態素辞書の形で記述し、それをトランスレートして形態素解析プログラムを得る[2]。本システムでは逐次型 L A X アルゴリズムをサポートし、この形態素辞書の開発と文法記述の効率的なデバッグを行うためのツールとして、それぞれ文法辞書エディタとインスペクタを備えている。これらのツールを用いて開発された形態素解析プログラムは L T B の一環として S A X の前処理部的な位置付けとなるが、単独でも使用可能なため、日本語インタフェースの入力部としての応用も容易である。

2. 形態素解析プログラム

L A X で開発される形態素解析プログラムは図1の様に、変換された形態素辞書と解析エンジンからなる。形態素辞書は複数のクラス構成となっている。解析は二つのフェーズからなる。まず、文章が入力されると解析エンジンは形態素辞書を用いて入力文の形態素解析を行い、文節候補を求める。このとき解析は決定的に行い、解析結果のオルタナティブを縮退されたデータ構造として一度に出力する。次に、この結果と各形態素の持つ構文情報を用いて、各文節候補の構文情報を決定する。これが L A X における形態素解析結果と呼ばれるものであり、構文解析システム S A X [3, 4] の入力データとなる。

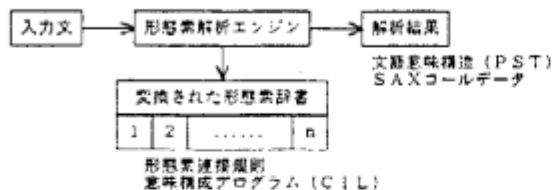


図1 形態素解析プログラムの構成

3. 形態素辞書記述形式

図2に形態素辞書記述形式を示す。大きく分けて連接規則の記述部と意味情報の記述部とに分けられる。後方連接素性には複数の状態遷移表が指定できる。'SS'以降に構文情報と意味処理を記述する。述語 in で前方形態素までに構成された P S T (C I L 部分項[5])を受け取り、共通処理を加えた上で後方接続形態素の種類による固有処理を行い述語 send で後方に P S T を送る。ここで send の代わりに call_sax と書けば必要なデータが S A X へ送られる。

各処理部はすべて C I L プログラムで記述する。

```

begin(カテゴリ名).
表層 :: 遷移表名a([状態名...]),
      && [ 遷移表名b([状態名...])... ],
      SS [ [in(i), 共通処理],
            遷移表名b(状態名, [固有処理, send(0)]),
            .... ],
....
end(カテゴリ名).
  
```

図2 形態素辞書記述形式

今回の開発ツールでは、'SS'以前の連接規則のデバッグ機能のみで、意味構成部のデバッグはサポートされていない。意味構成部のデバッグ機能は、今回作成したツールの機能を拡張する形で実現する予定である。

4. 開発環境の構成

図3に L A X の開発環境の構成を示す。マスター辞書[6]から必要な情報を取り出して形態素辞書の元を作り出す辞書トランスレータ、形態素辞書の内容を修正、追加するための L A X 辞書エディタ、解析プログラムを生成するジェネレータ、形態素解析の結果出力されるデータ構造をインスペクタとして連接チェックの再現や構文情報の構成途中のデータなど詳しい内部データを表示する L A X インスペクタなどから構成される。本辞書エディタを用いた編集は直接、辞書記述に対して行うのではなく、辞書記述を変換した中間データファイル群(オブジェクトの辞書クラスに対応)に対して行われる。これは、辞書内容の局所的な変

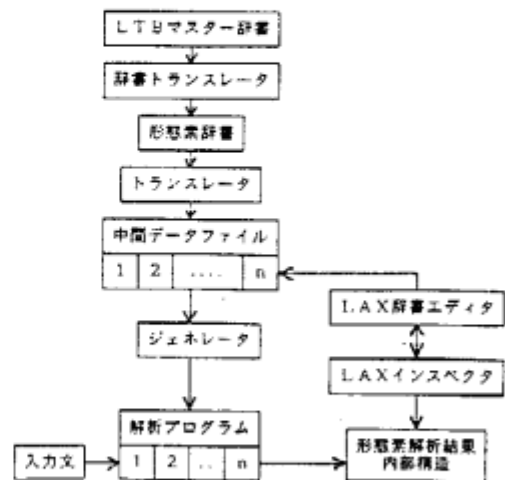


図2 L A X システムの構成

Environment of morphological analyzing system LTB-LAX
Yukihiko KUBO *Masataka YOSHIZUMI Hiroshi SANU
Kouji AKASAKA Ryoichi SUGIMURA ICOT *JIPDEC

