文法や加工手順を入力データとするテキスト加工システムＲＰＳの開発*

6Q－2

小林　要†
金沢工業大学§

1．はじめに

ソフトウェア開発において、プログラムや文書などのテキストを加工編集する作業は重要な作業である。プログラムが複雑であったりサイズが大きい場合には、テキストを自動的に加工するシステムが望まれると同時に、加工作業内容が適切な作業であることを確認し保証できることが極めて重要である。

本稿では、テキストの自動加工に用いた各種のデータを確認容易なデータとする汎用のテキスト加工システムＲＰＳを提案する。

2．システムの概要

本システムは、構文解析、木構造変換、テキスト生成の３つの基本機能を持つ。構文解析機能は、テキストの構文を、**“入力文法”**に基づいて解析し、**“役割木”**と呼ぶ構文木相当の中間データを出力する。木構造変換機能は、役割木の構造を、**“加工ルール”**に基づいて加工し、加工後の役割木を出力する。テキスト生成機能は、役割木に対応するテキストを、**“出力文法”**に基づいて出力する。これらの基本機能をどのように用いるかを表す**“加工手順”**を与えることにより、テキストの加工を行う。

加工内容は、加工手順データに記された手順と、用いられた入力文法、加工ルール、出力データを見ることにより確認することができる。

*Development of the Text Processing System RPS which inputs Grammar Data and Proccessing Instruction Data

†Kaname Kobayashi
§Kanazawa Institute of Technology

3．役割木

本システムの中間形式である役割木を以下のように定義する。視点集合Ｖ、順位パタン集合P_1、固定パタン集合P_2，順位役割集合R_1、固定役割集合R_2，副視点集合Ｓが与えられるものとする。ＶとＳは共通部分Ｍ＝Ｖ∩Ｓを持つが、他は互いに共通部分を持たない。順位役割集合には線形順序が与えられているものとする。Ｐ＝(P_1∪P_2)、Ｒ＝(R_1∪R_2)とするとき、役割木ＴをＶ，Ｐ，Ｒ，Ｓの４項関係Ｔ⊆Ｖ×Ｐ×Ｒ×Ｓで定義し、述語path(v,s,T)を以下(i)(ii)で定義する。

(i) (v,p,r,s)∈T ならば path(v,s,T)

(ii) path(v,w,T)∧path(w,s,T) ならば path(v,s,T)

役割木Ｔは以下の５条件を満たすものとする。

(1)関数従属性Ｖ→Ｐが成立する

(2)関数従属性ＶＲ→Ｓが成立する

(3)m∈M∧(v,p,r,m)∈T∧(v',p',r',m)∈T∧
(m,p",r",x)∈T ならば、v=v' ∧p=p' ∧r=r'

(4)(v,p,r,s)∈T ならば
p∈P_1∧r∈R_1 または p∈P_2∧r∈R_2

(5) どのｓ∈Ｓについても¬path(ｓ,ｓ,T)

役割木をテキストデータとして表現することもできるがその形式の説明は割愛する。

4．構文解析機能（役割木作成機能）

テキストの構文を解析するため、入力文法に基づきパーサを生成し、生成したパーサを用いて構文を解析する。解析結果は役割木で表現する。

文法はＢＮＦを少し拡張した形式を用い、**順位パタン定義文、固定パタン定義文、パタンセット定義文、語彙定義文**の４種類の文を用いて定義する。文法クラスは文脈自由文法の範囲である。

順位パタン定義文は、構文に繰り返しがある場

合の構文の記述に用いる。固定パタン定義文は、構文に繰り返しが無い場合の構文の記述に用いる。パタンセット定義文は、構文要素が複数の構文形式のどれかであることを記述する場合に用いる。語彙定義文はトークンの切り出し方を定めるために用いる。これらの記述形式についての説明は割愛する。

本システムの構文解析機能では、複数の入力文法を入力し、一方の文法で解析が失敗した場合に他の文法による解析に切り替える方式を採用し、複数の文法が混在するテキストの構文解析を可能とした。複数の文法が混在するテキストにはコメント型、名前空間型、部分文字列型の３種類の混在形態がある。**“コメント型”**の文法混在は、Ｃ言語のコメントのように、任意の場所に入り得て、かつ本来の構文木には登場しえなかった要素となる場合である。**“名前空間型”**の文法混在は、プリプロセッサ文のように、本来のＣ言語の文に新たな構文が追加された形で扱え、文法の名前で区別をつけることができる場合である。**“部分文字列型”**の文法混在は、文字列定数の中身が別文法の場合である。例えばプログラムの中でＳＱＬ文に相当する文字列定数を扱う例がそうである。

本システムでは、コメント型混在と名前空間型混在に対する処理までを構文解析機能で実現でき、部分文字列型混在の場合には、構文解析機能と木構造変換機能とを組み合わせて対処する。

現在までにＲＰＳの構文解析機能で生成するパーサには、ＬＬパーサとＬＲパーサの２種類がある。ただし、複数文法切り替えのバックトラック機能と、トークン切り出しにおけるバックトラック機能を追加し、汎用性を高めている点で従来のパーサと異なる。

５．木構造変換機能（役割木変換機能）

役割木を加工するため、加工ルールの列を与える。加工ルールを順に適用し、徐々に役割木の構造を加工する。各加工ルールは while-do 型ルールを用い、条件が成立する間、加工を継続し、条件が成立しなくなれば終了する。条件部は役割木の４項組とのマッチ・非マッチの条件の列で構成し、順にマッチさせ、マッチ変数の値を求める。マッチ変数の値が求まらない場合は、バックトラックして他のマッチ候補を探す。

マッチ変数に値が求まり、条件が成立すると、do 部分に書かれた４項組の列をマッチ変数の値を用いて構成し、役割木に順に強制追加または削除する。強制追加とは、既存役割木と矛盾する追加があれば、古い役割木の側の矛盾部分を削除して新しい４項組を追加する操作である。加工ルールの記述形式については説明を割愛する。

While-Do 型ルールを用いることにより、その適用直後の役割木において、While 条件が偽であることが証明でき、加工ルールの適用結果における成立条件を保証できる。また、役割木において各副視点の役割を識別できるので、加工箇所の意味的な指定ができ、加工ルールの再利用性が高い。

６．テキスト生成機能

テキスト生成には、役割木に出力用の文法データを与える。文法データの記述形式は入力文法の記述形式とほぼ同じとし、入力文法データを流用可能にした。テキスト生成は、役割木の順位パタンや固定パタンに対応する定義文を文字列への写像関数として用いることにより行う。

７．おわりに

プログラムなどの“文法を持つテキスト”を“加工手順データ”に基づいて自動加工する**“ソフトウェア加工システムＲＰＳ”**を開発した。文脈自由文法の範囲で、複数文法の混在した実際の様々なプログラムテキストの解析が可能となった。また、while-do 型加工ルールを用いることにより、加工後の役割木が満たす条件を保証することも可能となった。汎用性が高く、様々な言語を同時に扱えるので、言語変換や抽象化、詳細化などのソフトウェア開発における加工に利用できる。