プログラマブル テキスト コンバータ

4J－7

今津信一　新林満　姜力　山本俊行　筧捷彦
早稲田大学理工学部情報学科

1　はじめに

あるフォーマットにしたがって書かれているテキストファイルを別の形のフォーマットに直すという作業を行なう時、ある種のツール類が存在している。しかしこれらのほとんどは、決められたフォーマットの間のテキストでしか変換ができない。変換したいフォーマットに適したものがある場合は良いがいつもそうであるとは限らない。

たとえば構造に関するマークアップなどは明示されていない文章があったとする。しかしこの文章に章の番号などが振られていたとすれば構造を知ることができる。つまり、構造に関するマークアップが明示されていないものも、定型に収まっていれば構造を知ることはできるということである。そこで、変換元のテキストの構造を定義し、変換の方法を指定すれば望む結果が得られるようなテキストコンバータを作成した。

2　実現方法

このコンバータでは、

- 解析の方法を指定したファイル（gpi ファイル）
- 出力の方法を指定したファイル（gpo ファイル）

の２つのファイルを利用者が作成する。これらは変換の対象となるファイルに対して行なう処理の方法を記述したものである。

Programmable text converter
Shin'ichi Imazu, Mitsuru Shinbayashi, Jiang Li, Toshiyuki Yamamoto, Katsuhiko Kakehi
Department of Information & Computer Science, School of Science & Engineering, Waseda University

以下の説明では、文書の構造を解析する部分を**解析部**、解析された結果を用いて指定された通りに変換して出力する部分を**出力部**と呼ぶ。

3　解析部

3.1　gpi ファイル

変換の対象となるテキストをツリー構造に変換する方法を記述したファイル。

BNF に似た書式を用いてテキストの構造を示しており、主な書式としては

- 正規表現を定義する

  定義名 := 置換内容

があり、置換内容の部分に書かれるものには

1. 正規表現

   '[1-9][0-9]*(e|E)?[0-9]*'
2. 定義されたものを 0 回以上繰り返す

   { 定義名 定義名 ... 定義名 }

などのものが挙げられる。なお、正規表現を定義した定義名を特にノードネームと呼ぶ。

ツリー構造の最上部は top というノードネームで予約されている。

3.2　動作

gpi ファイルに従って解析を実行するにあたって、解析部は変換対象のテキストの先頭からマッチングを図っていく。マッチングが継続していれば、マッチングを図るテキスト上のインデックスを先に進めていき、マッチングしなかった場合は、インデックスをマッチングが継続していた位置まで戻すということを繰り返す。マッチングを

繰り返した結果、一番始めに予約されているtopがマッチングが継続した状態で終了した場合、入力解析成功ということになる。

解析に成功したら、解析した結果の中間ファイル（ツリー構造で表された変換元のファイル）を出力部に渡す。この際、ツリー構造の各ノードにはノードネームが入り、そのノードネームでマッチングした文字列はその下にぶら下がっている。

4 出力部

4.1 gpo ファイル

出力部で使用されるgpoファイルは、ツリー構造に変換された元の文書に対しての出力の方法を記述したファイルである。これはgpiファイルと同じようにBNFに似た書式を用いて記述されている。

ツリー構造を記した中間ファイルのノードの部分についているノードネームに対する基本的な書式は、

- ノードネームに対する出力指定

  ノードネーム := 出力指定

となり、該当する定義名ごとに出力指定を書くようになっている。

また、変数を扱うこともでき、出力指定の中に代入や演算、出力などの命令を記述できる。

4.2 動作

出力部では、ツリー構造で渡されたデータをテキストファイルへ変換して出力する。その際に出力部が行なう必要があることは

- 任意の「ノード以下の部分」を任意の位置に出力すること
- 任意の「文字列」を任意の位置に出力すること

の二つである。

渡されたツリー構造に従ってノードネームにぶら下がっている文字列をそのまま出力するのであれば、すべてのノードに対し

- 自分にぶら下がっている文字列を出力する。
- 自分のノードの下を解析する。

と定義すれば文字列が行きがけ順で出力できる。このケースが一番多いと考えられるので、何も指定しない場合のデフォルトはこうなるようになっている。

中間ファイルでのツリー構造と出力する際の順番が違う場合は、ノードネームで順番を決める方法と、キーとなるノードによってソートする方法の2つを使用できる。また、同じノードを2度以上出力する場合は出力定義もそれぞれに設定できる。

5 おわりに

このコンバータでは構造を定義する部分と出力を指定する部分とに分けたため、変換の対象となるファイルが同じならば出力の指定を変更するだけでいろいろな形に変換できる。

また、gpiファイルでは変換元の文書の構造のみを記述しており、意味の解析は一切行なっていない。つまりこのコンバータは、テキストファイルであって構造をgpiファイルに書くことができるものならば処理することができるということである。よって普通の文書に限らずいろいろなデータを加工することも可能である。今後はそのような用途についても考え、gpiファイルとgpoファイルの書式の拡張をしていくことになる。

参考文献

[1] MARTIN BRYAN 著 / 福島 誠 訳. *SGML* 入門. アスキー出版局, 1991

[2] 宮田重明・芳賀敏彦. *Tcl/Tk* プログラミング入門. オーム社, 1995

[3] A. V. Aho, R. Sethi, J. D. Ullman . *Compilers Principles, Techniques, and Tools.* Addison Wesley, 1986

[4] 斉藤孝. *UNIX yacc* と *lex* の使い方. HBJ 出版, 1992