

行組版機能の試作 JIS に基づいた行組版機能の試作

4 N-4

遠瀬雅宏, 伊藤聡, 早川栄一, 並木美太郎, 高橋延匡

(東京農工大学)

1. はじめに

現在、組版処理を行なうシステムは数多くあるが、その多くは欧米で開発された組版システムをベースに考案されている。そのため、和欧文混植などの日本語文書組版処理を行なうことができないものが多い。そこで、日本語ベースの組版システムを構築する知見を得るために、行組版処理の様々なパラメータを変化できる行組版機能を作成することを考えた。

この行組版機能の特徴は、JIS X 4051「日本語文書の行組版方法」[1]で規定される行組版規則、行組版処理モデルに基づいた行組版処理を行ない、行組版処理した文書を紙面に出力することである。また、JIS X 4051 の特徴として、不可分文字列を組版処理の単位とすることが挙げられる。不可分文字列とは、それ以上分割できない文字の集合である。出力する文章を不可分文字列に分割し、その不可分文字列単位ごとに組版処理を行なうことにより、和欧文混植、禁則処理を一括した処理で行なうことができる。

本報告では、JIS X 4051 に基づいた行組版機能を試作し、ペナルティ関数で使われるパラメータが行組版処理にどのような影響を与えるのかを考察する。

2. 行組版機能の設計方針

行組版機能機能の設計方針は、次のとおりである。

(1) JIS X 4051 の各種パラメータを設定できる

JIS X 4051 の行組版処理で使用するペナルティ関数のパラメータ、ウィドウ処理の文字数、各文字クラスの設定をパラメータファイルに記述することにより、各種パラメータを自由に設定することができる。

(2) 文書の論理構造を反映させる

章、節、項、段落といった文書の論理構造を指令

語として文書中に記述することにより、段落の最終行がある文字数以下にならないようにするウィドウ処理や章や節が見出しがページの最終行に配置されることを避ける処理を行なうことができる。

(3) 指令語埋込み式にする

文書の中に文字の大きさ、書体などの指令語を混在させて記述することにより、行組版の細かな設定を行なうことができる。また、ユーザが分離禁則処理の対象語を設定できる指令語を用意する。

3. 行組版機能の全体構成

行組版機能の全体構成を図 1 に示す。

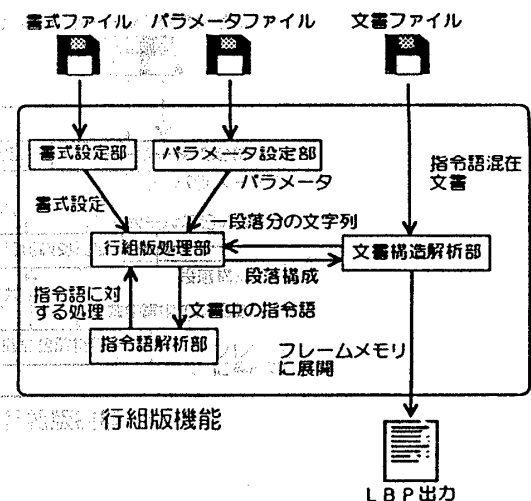


図 1 行組版機能の全体構成

書式ファイル、パラメータファイル、文書ファイルの三つの入力ファイルをこの機能に渡すことにより、行組版処理を行なう。書式ファイルには、紙のマーゲンや文書の論理構造ごとの文字の大きさ、書体の設定を記述する。パラメータファイルには、JIS X 4051 で規定される行組版処理を行なうために必要なパラメータを記述する。文書ファイルには、文書の論理構造や文字に対する修飾指令を表わす指

令語を混在させて印刷する文書を記述する。

4. 行組版処理部の概要

行組版機能では、段落単位で組版処理を行なう。段落一つ分の文字列に対して、JIS X 4051 で規定される行組版処理モデルを用いて行組版処理を行なう。JIS X 4051 の処理モデルを実現する上での特徴を次に示す。

(1) 不可分文字列リストを作成する

文字にクラスを与え、2文字間のクラス関係によって不可分文字列を作成する。そして、その連結リストを作成する。また、このときに文字のサイズ、書体、修飾といった情報を文字に付加する。

(2) 段落候補木を作成する

(1) で作成されたリストを設定した行長（以下、指定行長とする）単位に区切る。このとき、指定行長の前後2点を分割候補点とし、行頭のタグを付ける。そして、段落の最後までについて行頭のタグの二分木を作成する。1行に対して二つの分割候補点が存在する可能性があるため、N行の段落では最大 2^{N-1} 個の段落候補が作成される。

(3) 段落構成の決定

(2) で作成された段落候補の各行についてペナルティ関数と呼ばれる評価関数でペナルティ関数値を算出する。段落候補ごとにこのペナルティ関数の総和を求め、総和が最小になる段落候補を探索する。総和が同じになる段落候補については、行長調整における詰め処理の回数が多い方を採用する。

5. 行組版機能の試作と考察

本機能の中核となる行組版処理部の処理結果を考察を行なうために、一段落分の文字列に対し、行組版処理を行ない印刷するシステムを試作した。そして、JIS X 4051 で例示されるペナルティ関数で使われる三つのパラメータの A, B, C を変更したときに、それらのパラメータが処理結果にどのような影響を与えるのかを実験した。A は詰め処理罰則値、B は延ばし処理罰則値、C はウィドウ処理罰則値を表すもので、数値が小さいものほど優先される。A = 1, B = 50, C = 60 の例を図 2 に、A = 50, B = 1, C = 60 の例を図 3 に示す。

我々の研究室では、文書出力システム「JOSHO」の研究を行なっている。このシステムの特徴は、フレームメモリを採用していることである。

図 2 行組版機能の出力例 (1)

我々の研究室では、文書出力システム「JOSHO」の研究を行なっている。このシステムの特徴は、フレームメモリを採用していることである。

図 3 行組版機能の出力例 (2)

パラメータを次のように変化させた。

- ・ A < B の関係で、パラメータを変化
- ・ A > B の関係で、パラメータを変化

A < B の場合は詰め処理が優先されるために図 2 のように文書全体が詰って見える。A > B の場合は延ばし処理が優先されるので図 3 のように文書全体が空いて見える。A と B と C の間隔が大きすぎるとパラメータの数値が低い処理だけを行なう。文字間隔が開いている文書は間延びした印象がするので、A < B の方がよい。また、A と B の差が大きいと詰め処理ばかり行なう。これは、逆に文字間隔が詰った行ばかりになるので、好ましくない。JIS で例示しているパラメータの設定では、詰め処理と延ばし処理の両方が行なわれていた。

また、今回の段落構成の決定方法では段落ごとに構成を探索するので、段落構成が決定するまで処理結果がわからない。したがって、WYSIWYG 方式の文書清書システムよりもバッチ処理方式の文書処理システムに向いていると考える。

6. おわりに

今回、JIS X 4051 に基づいた行組版機能を試作した。しかし、行組版処理部だけを作成したので文書構造を反映させることはできない。今後は、文書構造解析部を作成し、文書構造を反映できるようにする予定である。

参考文献

- [1] 日本規格協会：“JIS X 4051「日本語文書の行組版方法」”，日本工業規格，1994