

# 卒論執筆の基本と卒論用スタイルファイル rgt.sty の利用方法

龍谷大学 先端理工学部 数理・情報科学課程

a23036 藤原 和将

指導教員 藤原 和将

## 概要

本書は、卒業論文の書き方について概説したものである。論文を作成する為には、文章を作文し、作成した文を適切に組版する事によって、文書とする必要がある。本書では、論文の為の作文における基礎的な要点と、組版に使用する  $\text{T}_\text{E}\text{X}$ ( $\text{X}_\text{L}\text{A}_\text{T}_\text{E}\text{X}$ ) の初歩的な内容、 $\text{X}_\text{L}\text{A}_\text{T}_\text{E}\text{X}$  で卒業論文を記述する為の、rgt.sty (Ryukoku Graduation Thesis) の取り扱いに付いて説明する。

作文の説明では、論文の特徴と、文章を構成する要素（即ち文と段落）における諸注意を与えた。又、推敲が如何に大切かという点について説明した。推敲は、文章を読み返して改良する作業である。推敲は、実は難易度の高い作業であるので、作文と推敲の要点を、文と段落に分けて本書で概説した。そして、数式を文の要素として如何に扱うべきかについて論じた。

数式を伴った文章を組版する上で、 $\text{T}_\text{E}\text{X}$  は一般的なソフトの一つである。 $\text{T}_\text{E}\text{X}$  は文書における文章と組版命令を読み込む事で、文書の組版を実行する。従って、 $\text{T}_\text{E}\text{X}$  で文書をある程度思い通りに組み上げる為には、組版命令を習得しなければならない。本書では、 $\text{T}_\text{E}\text{X}$  に入門する方法と、自習する為に前提となる知識について解説した。一方で、卒業論文を作成する上で、直接利用しない知識は付録に収録している。又、本書では、卒業論文で要請されている書式の組版命令を収録した rgt.sty について概説した。本書の解説と、本書自体を凡例として見る事で、卒論を執筆する準備は粗方整うはずである。

付録では、文字コードと  $\text{T}_\text{E}\text{X}$  エンジンについて触れた。

2023 年度卒業論文

# 卒論執筆の基本と卒論用スタイルファイル rgt.sty の利用方法

龍谷大学 先端理工学部 数理・情報科学課程

a23036 藤原 和将  
指導教員 藤原 和将

# 目次

1	はじめに	1
2	作文	1
2.1	論文の特徴 . . . . .	1
2.2	推敲 . . . . .	2
2.3	文の構成 . . . . .	2
2.4	段落の構成 . . . . .	3
2.5	数式の取り扱い . . . . .	3
3	卒論の組版について	3
3.1	T <sub>E</sub> X の概要 . . . . .	4
3.2	T <sub>E</sub> X の入手・導入 . . . . .	4
3.3	T <sub>E</sub> X の基礎用語 . . . . .	5
3.4	rgt スタイルの利用 . . . . .	6
3.5	独自の環境 . . . . .	7
A	付録	7
A.I	文字コード . . . . .	7
A.II	T <sub>E</sub> X エンジン . . . . .	8
A.III	T <sub>E</sub> X のコーディング Tips . . . . .	9

# 1 はじめに

本文では、卒業論文を執筆の仕方について解説する。卒業論文とは卒業に必要な論文の事であり、論文とは学術的に価値がある内容が記された文書である。話を進める前に、作文の為の用語を整理しよう。

## 定義:1.1

文とは、終止符で終了する言葉の繋がりである。

文章とは、文の連なりである。

文書とは、専ら文字を用いた表現や記録である。

例えば、この PDF の記載内容をメモ帳のコピーしたものは同じ文章であるが、異なる文書である。見やすい論文を執筆するとは、

1. 解読しやすい文章を作文する
2. 判読しやすい文書を作成する

の二つの作業が必要である。本文では、第二章で作文の基礎知識について解説する。又、第三章では、文書を作成する為の組版ソフトである  $\text{T}_{\text{E}}\text{X}$  と卒業論文用の `rgt` ファイルについて説明する。

## 2 作文

この章では、論文などの、情報を精確に伝える趣旨の文章を作文する為の注意点を述べる。

### 2.1 論文の特徴

論文の特徴は、主張がある事である。論文の使命は、この主張を読み手に説得させる事にある。従って、当然のことながら、論文では主張を明示する必要がある。学術論文を構成する主張以外の文章は専ら、主張の意義や新規性、並びに主張の根拠を示す為の文章である。一方で、主張に貢献しない文は、読み手の混乱を引き起こす為、極力控えることが望ましい。

## 2.2 推敲

推敲とは、一度作文した文章を読み返して改善する作業である。推敲の際は、以下について確認する。

- 現在の文が論文の主張にどう貢献しているか
- 段落の構成に支障はないか
- 文の構成に支障はないか
- 誤字、脱字、語彙、「てにをは」に支障はないか

段落と文の構成については次節に述べる。

**推敲は、論文を書く上で必要不可欠である。** 作文をしていると、論文の主張を見失ったり、後述の文や段落の規則を破りがちである。この為、どこかのタイミングで必ず読み返しながら、上記の点を一通り確認しなければならない。

## 2.3 文の構成

文を構成する上で、**主語と述語は整合しなければならない。** 文を書き下す際は、主語が読み手である場合を除いて、主語を明示すべきである。又、文を締め括る際は、主語に対応した述語を配置しなければならない。

**文を構成する単語の意味を、書き手は把握していなければならない。** 単語の誤用は百害あって一利無しであるので、辞書を引くか、自分が把握している言葉に置き換えるべきである。

直前の文章を鑑みて、**必要な場合は、文頭に適切な接続詞や節を配置すべきである。** 但し、接続詞は必ずしも配置する必要はない。接続詞はややもすると濫用しがちになり、読み手の理解の妨げとなる為、接続詞の使用は最低限度に留めるべきである。

**文の長さは、出来る限り短くすべきである。** 長い文は、読み手によって読み辛いだけでなく、書き手にとっても混乱を来し易くなる。この為、文が長くなってしまう場合は、以下の対策を検討する。

- 不要な文言を削除する
- 同一文内の重複した部分を集約する
- 分割する
- 短い語彙に言い換える
- 前後の文との配置を転換し、複数の文単位での集約をする

## 2.4 段落の構成

段落とは意味内容が纏まった文章の連なりである。段落の趣旨を明示する為に、段落の最初と最後の文では、段落の趣旨を可能な限り直接扱うべきである。最初の文で趣旨に触れる理由は、段落を読む上での方向を読者に示す事で、内容の理解を助ける為である。一方で、最初と最後の段落で趣旨が変わってしまう段落は、意味内容の纏まりがなく、段落としての体を成していない。従って、段落の趣旨は先頭と末尾で明示するべきである。

一方で、段落の趣旨を段落内で説明しない段落については、上記の規則は適用されない。例えば、項目を列挙する事が趣旨の段落は、段落の趣旨を段落内で説明しない段落である。この場合、段落の趣旨を先頭で明記していれば、末尾で趣旨に立ち返る必要はない。

## 2.5 数式の取り扱い

数式は文の一部を構成する。数式を利用する上での主な注意点は以下である。

- 文の切れ間には句読点を付ける。
- 新しい文字の利用は最低限に留める。
- 対象毎に、極力別の文字を利用する。
- 対象の種類と、文字の種類を極力一致させる。
- 新しい文字を断りなく利用しない。但し、直後に断る場合はよい。
- 自然言語で書いた場合に、却って読者の混乱を生む場合は、論理式を用いる。

### 註:2.1

セミナーの場合、板書の時間を削減する目的で論理式の利用が推奨される場面が多い。一方、文章の場合は読み易さが重視される為、自然言語での説明が好まれる場面が多い。

## 3 卒論の組版について

この章では、 $\text{X}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$  を用いて卒論の文書を成す方法について述べる。 $\text{T}_{\text{E}}\text{X}$  にはエンジンと呼ばれる種類があるが、 $\text{X}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$  はその一つである。エンジンについては付録で触れる。

### 3.1 T<sub>E</sub>X の概要

T<sub>E</sub>X は、文章と組版の為の命令を読み込み、文書を作成するソフトである。T<sub>E</sub>X に読み込ませる為の文章を書き込んだテキストファイルをソースファイルという。T<sub>E</sub>X にソースファイルを入力し、解釈させる（この作業をコンパイルという）ことで、始めて文書を得ることができる。

T<sub>E</sub>X がソースファイルの読み込みに成功すると、組版の結果をファイルで出力する。X<sub>Y</sub>L<sub>A</sub>T<sub>E</sub>X の場合、出力結果は PDF ファイルとなる。他のエンジンの場合は付録で扱う。X<sub>Y</sub>L<sub>A</sub>T<sub>E</sub>X では、PDF に直接出力する為、PDF ファイルに対する命令群も用意されている。

T<sub>E</sub>X に入力するソースファイルに、組版の為の命令を直接書き下す点は、Microsoft Word とは大きく異なる。Microsoft Word や Google Document は、最終的に出力される文書を直視しながら文書編集が出来る様になっている。上記の様なソフトにおいても、組み版命令群が記録されたファイルがあるが、基本的にユーザーは組み版命令を直接触れることはない。一方で、上記のソフトでは、実際の組み版命令とユーザーが直視しているインターフェースを繋ぐ必要がある為、T<sub>E</sub>X の様なソフトに比べて演算機の容量をより消費する必要がある。

T<sub>E</sub>X の利点の一つは、ソースファイルを直接扱える点にある。ソースファイル自体はテキストファイルである為、編集の為のコストが非常に小さい。その代わり、T<sub>E</sub>X 自体には、グラフィカルなインターフェースや、バックアップ機能、添削機能といったものは無い点に注意する必要がある。

### 3.2 T<sub>E</sub>X の入手・導入

T<sub>E</sub>X に入門する為の最大の難関は T<sub>E</sub>X の導入であった。この理由の一端は、T<sub>E</sub>X は元々 Unix 用のソフトウェアであり、Windows OS や Mac OS の為に作成されたプログラムではない点が挙げられた（その昔は、T<sub>E</sub>X を動かす為の機械が高価だった事などもある）。一方現在では、手頃な Windows OS や Mac OS への導入方法が用意されている。特に、各 OS のパッケージマネージャーが T<sub>E</sub>X に対応している場合が多い為、端末ソフトの操作ができれば導入は容易である。因みに、Windows OS の場合、WSL(Windows Subsystem for Linux) によって仮想的に Linux が使用できるので、Linux 付属のパッケージマネージャーも利用できる。筆者の知る限り iOS には未対応である。（部分的な対応の試みはある）

一方、現在はオンライン上で T<sub>E</sub>X を利用するサービスが存在する。overleaf は上記のサービスの一つであり、PC に T<sub>E</sub>X がインストールされていなくとも、オンライン環境であ

れば  $\text{T}_{\text{E}}\text{X}$  を利用する事ができる. 又, overleaf では,  $\text{T}_{\text{E}}\text{X}$  の編集環境も整っており,  $\text{T}_{\text{E}}\text{X}$  への入門を簡単にしている. 藤原研究室でも, overleaf の利用を推奨している. (overleaf を利用した場合, 定期的に藤原研究室の共用 PC に, 資料のバックアップが作成される.)

### 3.3 $\text{T}_{\text{E}}\text{X}$ の基礎用語

$\text{T}_{\text{E}}\text{X}$  に入力するソースファイルを用意する為には,  $\text{T}_{\text{E}}\text{X}$  の文法を理解しなければならない. 以下に  $\text{X}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$  の基本事項について述べるが, これだけでは不十分なので, 適宜必要なものは調べる事.

#### 3.3.1 数式モード

$\text{T}_{\text{E}}\text{X}$  には数式モードと呼ばれる, 文章を解釈する仕組みがある. 数式モードにおいては, 数式を記入する為の命令を  $\text{T}_{\text{E}}\text{X}$  が受け付けるだけでなく, アルファベットの入力も全て変数として, 数式用のフォントが適用される. 数式モードには文中式と独立行式があり, 文中式は  $\$$  で挟まれる部分に適用される. 一方, 独立行式は  $\backslash[, \backslash]$  によって挟まれる部分に適用される.

#### 3.3.2 コマンド

$\text{T}_{\text{E}}\text{X}$  におけるコマンドとは,  $\backslash$  (バックスラッシュ) で始まる文字列で表される  $\text{T}_{\text{E}}\text{X}$  への命令である. 環境によっては,  $\backslash$  ではなく,  $\yen$  マークが表示される場合がある.

$\text{T}_{\text{E}}\text{X}$  のコマンドには引数を取るものと取らない物がある. 引数を取らないコマンドは, 組版命令だけでなく, キーボードにない文字の入力に用いられる. 例えばコマンド  $\backslash\alpha$  は数式モード中において  $\alpha$  を出力するためのコマンドである. 又, コマンドの中には引数を取るものもある. 例えば, ソースファイルの書き出しに配置される  $\backslash\text{documentclass}\{\}$  コマンドは, 大まかな組版命令が定義されているクラスファイルを読み込むコマンドである. 又, 組版命令を追加するスタイルファイルを読み込む為のコマンドは,  $\backslash\text{usepackage}\{\}$  である.

数式モードにおいて, ギリシャ文字や数学の文字様記号 ( $\mathbb{R}$  など) の入力は, 基本的にコマンドを用いて行う. 一方で,  $\text{X}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$  は Unicode 準拠なので, unicode-math スタイルを使用する事で, コマンドを使用せずに直接 Unicode の文字様記号を入力できる. Unicode などの文字コードについては付録で述べる. 例えば

$$\int_{\mathbb{R}} \langle \tau + \xi \rangle^{-\alpha} \langle \tau - \xi \rangle^{-\alpha-\beta} d\xi$$



を入力する場合、通常は数式モードにおいて

```
\int_{\mathbb{R}} \langle \tau + \xi \rangle^{-\beta} \langle \tau - \xi \rangle^{-\alpha - \beta} d\xi
```

の様に入力する必要があるが、Xe<sub>La</sub>TeX と unicode-math を併用した場合は、

$\int_{\mathbb{R}} \langle \tau + \xi \rangle^{-\alpha} \langle \tau - \xi \rangle^{-\alpha - \beta} d\xi$  とできる。

### 3.3.3 環境

TeX には begin コマンドで始まり、end コマンドで終了する環境と呼ばれる文法がある。環境においては、指定された組版規則が適用される。例えば、TeX で文章を入力する為には、document 環境を利用する。この為、TeX の文章には、`\begin{document}` と `\end{document}` が必ずある。加えて、場合分けや行列を扱う為の数式モードで利用される環境や、数式モードを特定の組版で行う為の環境もある。

### 3.3.4 プリアンブル

TeX の文章は document 環境に格納されるが、文章を開始する前に組版命令を行いたい場合は、document 環境の前に命令を記載する。この document 環境の前の部分をプリアンブルという。プリアンブルは基本的にはコマンドのみが受け付けられ、その他の文字列が認識された場合、TeX は異常停止する。文章全体の組版の基礎を規定するクラスファイルや、追加の組版命令が記載されたスタイルファイルは、プリアンブルにコマンドを記載する事で読み込まれる。

## 3.4 rgt スタイルの利用

rgt.sty は卒論の為に作成されたスタイルファイルである。rgt.sty は、卒論の為の基本的な間隔調整や命題の為の環境、数式番号の取り扱いなどの規則で構成されている。この章では、rgt.sty の利用方法について述べる。

### 3.4.1 独自のコマンド

- `\Nendo:` 引数には、卒業論文を提出する年度を記載する。
- `\Myouji:` 引数には、筆者の名字を記載する。
- `\Namae:` 引数には、筆者の名前を記載する。
- `\ShidouMyouji:` 引数には、指導教員の名字を記載する。
- `\ShidouNamae:` 引数には、指導教員の名前を記載する。

`\Shokukai:` 引数には, 指導教員の職階を記載する.  
`\GakusekiNum:` 引数には, 筆者の学籍番号を記載する  
`\title:` 引数には, 論文題目を記載する.  
`\maketitle:` 引数なし. 題目ページを作成する.  
`\Mokuji:` 引数なし. 目次ページを作成する.

### 註:3.1

通常の  $\text{T}_{\text{E}}\text{X}$  では筆者氏名を指定するコマンドは `\author` である. 卒論では, 名字と名前の間に間隔調整を行う為, ここでは独自のコマンドを利用している.

## 3.5 独自の環境

`abstract:` 通常の概要ではなく, 卒論用の概要ページを作成する. `\maketitle` コマンドの前に記載する.  
`Definition:` 定義を行う為の環境.  
`Theorem:` 定理を記述する為の環境.  
`Proposition:` 命題を記述する為の環境.  
`Lemma:` 補題を記述する為の環境.  
`Corollary:` 系を記述する為の環境.  
`Remark:` 註を記述する為の環境.

## A 付録

### A.I 文字コード

現在のコンピュータは2進数しか扱う事が出来ない為, 文字を2進数に符号化する必要があり, 符号化の方法を文字コードという. 文字コードは, 残念ながら統一されておらず, 歴史的経緯から特に日本語には多くの文字コードが混在している. 主な文字コードを挙げる.

■**ASCII** 情報交換用米国標準コード. 古い文字コードであり, 128 文字 (0 から 127 番) しか登録されていない. 内訳は改行や削除などの制御用の文字, 英語のアルファベット, そして記号である. 基本的に後発の文字コードは, ASCII の文字コードを拡張する様に設計

されている。

■SJIS アスキー社と Microsoft 社によって開発された日本語用文字コード。なぜか、ASCII で `\` が登録されている 5C(92 番) に `¥` が登録されている為、他の文字コードで書かれたテキストファイルを読み込む際には、`¥` として文字化けしてしまう。

■EUC Unix や Linux で主に使用されていた文字コード。日本語用の EUC-jp だけでなく、韓国語用や中国語用の EUC もある。

■ISO2022-jp 今もメールで使用されている文字コード。

■UTF-8 全ての言語を一つも文字コードで統一する事を目指した文字コード規格 Unicode 規格のうち、8bit 単位の 1-4byte の可変長の文字コード。卒業論文を作成する上でも使用を推奨する。UTF-8 では、数学用の記号や異体字なども登録されている。但し、UTF-8 に割り当てられた文字を網羅しているフォントは、筆者の知る限り存在しない。

#### 註:A.1 異体字

異体字とは、同じ漢字で字形が異なるものであり、例えば、高 (u+9ad8) と高 (u+9ad9) などがある。困った事に、日本語には大量に異体字があり、Unicode の 16 進数 4 桁の番号が振られていない字も多数ある上、入力方法も文字コードの様に統一されていない。Unicode では異体字に対応する為、異体字セクタと呼ばれる制御用の文字を利用する仕組みが用意されている。一方、文字コードで符号化規則が定まっても、フォントが対応した字形を収録していない事もある。

## A.II T<sub>E</sub>X エンジン

T<sub>E</sub>X にも種類がある。X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X を利用する理由は Unicode に対応しているからであるが、ここで他の T<sub>E</sub>X についても紹介する。ここで挙げる以外にも、T<sub>E</sub>X の種類はあるが、ここでは割愛する。

■(plain) T<sub>E</sub>X 狭義には、T<sub>E</sub>X とはこの plain T<sub>E</sub>X を指す。plain T<sub>E</sub>X は始祖の T<sub>E</sub>X であり、Knuth 教授によって、自分の論文を綺麗に出版する為に開発された。plain T<sub>E</sub>X の文法は上記のコマンドとは異なっており、環境を利用しない。加えて、plain T<sub>E</sub>X の出力は、PDF でなく、DVI ファイルと呼ばれる独自形式であり、PDF で出力する為には、DVI ファイルを PDF ファイルに変換する必要がある。

■**L<sup>A</sup>T<sub>E</sub>X** T<sub>E</sub>X の独自のコマンド体型を、より一般向けに Lamport 氏によって開発されたエンジン。出力は DVI ファイル。

■**pT<sub>E</sub>X/pL<sup>A</sup>T<sub>E</sub>X** 日本語に対応させた T<sub>E</sub>X と L<sup>A</sup>T<sub>E</sub>X。現在日本人が日本語の文章を作成するときの標準エンジン。出力は DVI ファイルである。幾つかの文字コードに対応しているが、内部的には sjis か EUC である為、Unicode 等の独自機能を利用するのは工夫が必要である。又、DVI ファイルはフォントを埋め込み出来ない為、ゴシックフォントや明朝フォントの指定をする為には、DVI ファイルを変換する際にフォントを指定する命令を追加する必要がある。Unicode 対応した upT<sub>E</sub>X/upL<sup>A</sup>T<sub>E</sub>X も存在する。

■**PDFT<sub>E</sub>X/PDFL<sup>A</sup>T<sub>E</sub>X** 標準出力を DVI ではなく、PDF に変更した T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X。PDF を直接出力する為、T<sub>E</sub>X ソースコードからフォントの指定などが出来るようになった。日本語は初期状態では対応していないので、日本語を使用する場合は相応の工夫が必要となる。又、縦書きの「ルビ」など、日本独特の組版規則の適用は困難。

■**X<sub>Y</sub>T<sub>E</sub>X/X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X** Unicode を直接扱えるようにした T<sub>E</sub>X。Unicode には日本語が登録されている為、日本語を表示するだけなら簡単にできるが、日本語の組版処理の規則は初期状態では適用されない。この為、日本語を利用するためには、bxjsarticle クラスを利用する必要がある。フォントの利用は、fontconfig というソフトを経由する。

■**LuaT<sub>E</sub>X/LuaL<sup>A</sup>T<sub>E</sub>X** Unicode の対応に加え、Lua 言語による命令を受け付けるようにした T<sub>E</sub>X。Lua を利用して、グラフの描画などが可能となる。

### A.III T<sub>E</sub>X のコーディング Tips

1. 数式番号なしの単独独立数式を記入する場合、 $\$ \$ \$ \$$  を利用する案内もあるが、 $\$ \$ \$ \$$  は plain T<sub>E</sub>X 用のコマンドであるので、余白などに微妙に影響する為、 $\backslash[.\backslash]$ を用いる。
2. T<sub>E</sub>X では独自コマンドを作成できる。一方で、独自コマンドは他人がソースを読む際に混乱き来すので、極力使用しない。