2025 年度 特別研究論文

卒論執筆と typst スタイルの活用法

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、文を作文し、 文章を構成し、 文章を適切に組版(配置）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、 Typst で卒業論文を組版する為の、rgt.typ（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。 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、 Typst は比較的新しいソフトである。 現在、数式を伴った文章の組み版ソフトの主流は TeX である。 然し、TeX の独特の文法や、TeX の組み版の複雑さから、 Typst の様な新しい組版ソフトが登場した。 Typst には新しさ故に未熟さも否めないが、 TeX と比べて基本的に簡便となる様に設計されている。

本書で、説明するスタイルファイルは、 卒業論文の組版を行う為に、 Typst 独特の文法を学生が一切触れずに済むように用意したものである。 特に、Typst の更新速度を鑑みて、現在のバージョン 0.13.1 に対応した基本的なコマンドのみで作成している。

2025 年度 特別研究論文

卒論執筆と typst スタイルの活用法

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

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、

- 解読しやすい文を作文する
- 意味内容が連綿とした文章を作成する
- 判読しやすい文書を作成する

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、Typst の基礎的な内容について解説する。

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、**書き手は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。

書き手は、**文を構成する単語の意味を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るべきである。**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。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。** 最初の文で趣旨を触れる理由は、 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、項目を列挙する事が趣旨の段落では、 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、 口頭での説明が無く、 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. Typst
この章では、Typst を用いて卒論の文書を成す方法について述べる。 Typst の version は 0.13.1 である。

3.1. 概要
Typst は、Markdown の様な文法で文書を記述する事ができる組版ソフトである。 [1] では、 次のように説明されている。

> Typst is a new markup-based typesetting system for the sciences. It is designed to be an alternative both to advanced tools like LaTeX and simpler tools like Word and Google Docs. Our goal with Typst is to build a typesetting tool that is highly capable and a pleasure to use.
>
> — https://typst.app/docs/

> Typst は科学分野向けの新しいマークアップ型組版システムです。 これは LaTeX のような高機能ツールと、Word や Google ドキュメントのようなより簡易なツールの 両方の代替 となることを目指して設計されています。 私たちの目標は、Typst を 高度な能力を備えつつ、使っていて楽しい組版ツール にすることです。
>
> — ChatGpt による翻訳

Typst の導入方法は、公式ドキュメントを参照してほしい。 オンラインでも利用できる。 また初等的な文法も、公式ドキュメントを確認してほしい。 というのも、Typst は比較的新しい組版ソフトであり、 仕様変更の為か情報の陳腐化が見られるからである。

3.2. Typst の数式
Typst の数式は、所謂 TeX の文法とは異なり、エスケープしない点が特徴である。 以下に、数式のコマンドの違いを見てみよう。 基本的に、Typst の入力の方が LaTeX の入力よりも簡便である。

例 3.1:
$f\left(\int_X g d\rho\right) \leq \int_X f \circ g d\rho$

LaTeX

```
$
f\left(\int_X g d \rho\right)
\leq \int_X f \circ g d\rho
$
```

XeLaTeX + unicode-math

```
$f \left( ∫_X g d ρ \right) ≤ ∫_X f ∘ g d ρ$
```

Typst

```
$f( integral_X g d rho) <= integral_X f compose g d rho$
```

Typst + Unicode

```
$f(∫_X g d ρ) ≤ ∫_X f ∘ g d ρ$
```

例 3.2:

$(1\ \ 2)\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}\begin{pmatrix}1\\ 2\end{pmatrix}$

LaTeX

```
$
\begin{pmatrix}
  1 & 2 \\
\end{pmatrix}
\begin{pmatrix}
  1 & 2 \\
  3 & 4 \\
\end{pmatrix}
\begin{pmatrix}
  1 \\
  2
\end{pmatrix}
$
```

Typst

```
$mat(1,2) mat(1, 2; 3, 4) mat(1;2)$
```

3.3. 本スタイルで用意されているコマンド

3.3.1. 概要と表紙

このスタイルファイルでは、卒業論文の表紙を作成するためのコマンドを用意している。次の書式で、作成する。

```
#show: thesis.with(
title: [
卒論執筆とtypstスタイルの活用法
],
author: (
  id: "a23036",
  family-name: "藤原",
  given-name: "和将",
),
supervisor: (
  family-name: "藤原",
  given-name: "和将",
),
```

```
affiliation: (
  university: "龍谷大学",
  department: "先端理工学部",
  course: "数理・情報科学課程",
),
academic-year: "2025",
abstract: [
  概要はここにかく。
],
)
```

3.3.2. 定理環境

このスタイルファイルでは、以下の定理環境を用意している。

定義 3.3: サンプル
定義のサンプル

```
typst
  #definition(title:"サンプル")[
    定義のサンプル
  ]
```

定理 3.4:
定理のサンプル

```
typst
  #theorem[
    定理のサンプル
  ]
```

命題 3.5:
命題のサンプル

```
typst
  #proposition[
    命題のサンプル
  ]
```

補題 3.6:
補題のサンプル

```
typst
  #lemma[
    補題のサンプル
  ]
```

系 3.7:
系のサンプル

```
typst
  #corollary[
    系のサンプル
  ]
```

註 3.8:
註のサンプル

```
typst
  #remark[
    註のサンプル
  ]
```

例 3.9:
例のサンプル

```
typst
  #example[
```

```
    例のサンプル
]
```

共通して、以下の特徴がある：

- title 属性を追加できる
- 番号は、level 1 header の子カウンターとなっている
- 番号は、環境で共通している

3.4. 本スタイルで変更されているコマンド

3.4.1. `quote` コマンド

サンプルコード

```
#quote(attribution: "帰属")[
  引用文
]
```

サンプル出力

> 引用文
>
> — 帰属

定義文

```
set quote(block: true)
show quote.where(block: true):
block.with(
  stroke: (left: 2pt + blue, rest:
none),
)
```

3.4.2. `link` コマンド

サンプルコード

```
#link("https://typst.app/docs/")[
  Typstの公式ドキュメント
]
```

サンプル出力 Typst の公式ドキュメント

定義文

```
  show link: set text(fill: blue)
```

3.4.3. `raw` コマンド

サンプルコード

````
  ``` `latex
  This is not \TeX !
  ``` ` ```
````

サンプル出力

```
  This is not \TeX !
```

定義文

```
  set raw(block: true)
  show raw.where(block: true): block.with(
    fill: rgb("#f6f8fa"),
    inset: (x: 12pt, y: 8pt),
```

```
    radius: 4pt,
  )
```

3.5. 参考文献の書き方

参考文献はのリストは、csl ファイル等を用いなければならない。 ここでは、 `rgt.csl` を用意して用いている。

```
<?xml version="1.0" encoding="utf-8"?>
<style xmlns="http://purl.org/net/xbiblio/csl" class="in-text" version="1.0">

  <info>
    <title>Ryukoku Graduation CSL</title>
    <id>http://example.com/styles/minimal-numeric-accessed</id>
    <category citation-format="numeric"/>
    <updated>2025-07-16T00:00:00+00:00</updated>
  </info>

  <citation>
    <layout delimiter=", ">
      <text prefix="[" variable="citation-number" suffix="] "/>
    </layout>
  </citation>

  <bibliography>
    <layout delimiter=" ">
      <text prefix="[" variable="citation-number" suffix="] "/>
      <group delimiter=", ">
      <names variable="author">
        <name/>
      </names>
      <text variable="title" font-style="italic"/>
      <text variable="URL"/>
      <text variable="note"/>
      </group>
    </layout>
  </bibliography>

</style>
```

また今回の `bib` ファイルは以下のもの

```
@webpage{okumura,
  author  = {Okumura, Haruhiko},
  title = {Typst入門},
  year  = {2025},
  url   = {https://okumuralab.org/~okumura/misc/241111.html},
  note  = {Accessed: 2025-07-16}
}

@webpage{typstdocs,
  title        = {Typst Documentation},
```

```
  url          = {https://typst.app/docs/},
  note     = {Accessed: 2025-07-16}
}
```

3.6. 2025 年 7 月現在での注意

1. Typst は比較的新しい組版ソフトである為、仕様変更が頻繁に行われる。このため、
   - 生成 AI によるコード生成の精度は低い
   - 公式ドキュメント以外の情報が陳腐化している可能性がある
   - 気づくと新機能が追加されている
2. Typst の外部ファイル参照について、現在のところ、
   - `import` コマンドは相対パスでしか利用できない
   - `import` コマンドでシステム環境変数を利用できない
   - texmf の様なパッケージ管理システムがない
   - `import` コマンドで、github などのリモートファイルを読み込む場合は、キャッシュ生成時のみに読み込まれる
     - このため、github のレポジトリが更新されても、自動で更新されない
3. Typst の組み版では、今のところ、
   - 段落の字下げが完全にできない
     - 特に最初の段落はハードコーディングされてしまっている
   - 日本語の組み版規則に完全に準拠しきれてない [2]
     - 理系ではあまり使用する機械は感じない。

4. 付録

4.1. 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 1-4bite 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 4.1: 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）と髙（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、異体字セレ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いても、フォントが対応した字形を収録していない事もある。

参考文献

[1] *Typst Documentation*, https://typst.app/docs/, Accessed: 2025-07-16

[2] Haruhiko Okumura, *Typst* 入門, https://okumuralab.org/~okumura/misc/241111.html, Accessed: 2025-07-16