所属

タイトル

著者名

2024 年 4 月 16 日

概要

概要(アブストラクト)です

キーワード: キーワード 1, キーワード 2

目次

第 1 章
序論

Typst は markdown like なコーディングで pdf, ポスター, スライド等のドキュメントを作成できます. Rust 言語で書かれており, コンパイルが速いのが特長です.

1.1 Typst は優秀だ

こんな感じで `@ss8843592` or `#cite(<ss8843592>)` と引用できます

こんな感じで [1] or [1] と引用できます

1.1.1 エレガントに書ける

数式

```
$ mat(1, 2; 3, 4) $ <eq1>
```

と書くと

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \tag{1.1}$$

式 (1.1) を書くことができます.

関数を作れば

図 1.1 イメージ

図 1.1 を表示できますし,

表 1.1 テーブル [2]

t	1	2	3
y	0.3s	0.4s	0.8s

表 1.1 も表示できます.

第 2 章
先行研究

卒論や修論や学会の予稿等の作成においては Typst [2] の使いやすさから置き換わるのではないかと思います(半分願望).

図 2.1 Typst + git [2]

第 3 章
定義

Typst では関数定義が簡単であるので定理の書き方などをカスタマイズできます.

3.1 定義例

thmbox 関数を作ってカスタマイズをできるようにしました.

```
#let theorem = thmbox(
 "theorem", //identifier
 "定理",
 base_level: 1
)

#theorem("ヲイラ-")[
 Typst はすごいのである.
] <theorem>
```

定理 3.1: (ヲイラ-)
Typst はすごいのである.

```
#let lemma = thmbox(
  "theorem", //identifier
  "補題",
  base_level: 1,
)

#lemma[
  Tex はさようならである.
] <lemma>
```

補題 3.2:
Tex はさようならである.

このように, 定理 3.1 , 補題 3.2 を定義できます .
カッコ内の引数に人名などを入れることができます. また, identifier を変えれば, カウ

ントはリセットされる. identifier 毎にカウントを柔軟に変えられるようにしてあるので, 様々な論文の形式に対応できるはずです.

```
#let definition = thmbox(
 "definition", //identifier
 "定義",
 base_level: 1,
 stroke: black + 1pt
)
#definition("Prime numbers")[
 A natural number is called a _prime number_ if it is greater than $1$ and
 cannot be written as the product of two smaller natural numbers.
] <definition>
```

定義 3.1:
Typst is a new markup-based typesetting system for the sciences.

定義 3.1 のようにカウントがリセットされています.

```
#let corollary = thmbox(
  "corollary",
  "Corollary",
  base: "theorem",
)

#corollary[
  If $n$ divides two consecutive natural numbers, then $n = 1$.
] <corollary>
```

Corollary 3.2.1:
If n divides two consecutive natural numbers, then $n = 1$.

base に identifier を入れることで Corollary 3.2.1 のようにサブカウントを実現できます.

```
#let example = thmplain(
  "example",
  "Example"
).with(numbering: none)

#example[
  数式は\$\$ で囲む
] <example>
```

例:
数式は $$ で囲む

thmplain 関数を使って plain 表現も可能です.

参考文献

[1] S. Hussain, S. Bai, と S. Khoja, *Content MathML(CMML) conversion using LATEX Math Grammar (LMG)*, *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, Vol. 0 (2019), pp. 1–5

[2] L. Mädje, A Programmable Markup Language for Typesetting, 2022