

SATySF_Iで卒論を書いた話

SATySF_I Conf 2022

pickoba

2022.09.24

自己紹介

- pickoba (修士 1 年)
- 言語 : C#, TypeScript, Scala
- SAT_YSF_I 歴 : そろそろ 1 年
- 公開している SAT_YSF_I 関連のもの
 - VS Code 拡張 [SAT_YSF_I Workshop](#)
 - 擬似コード組版ライブラリ [SAT_YSF_I Algorithm](#)
 - GitPod 向けテンプレート [SAT_YSF_I GitPod Template](#)

今日のテーマ SAT_YSF_I で卒論を書いた話

SAT_YSF_Iとの出会い

- SAT_YSF_Iに出会ったのは、去年の10月
- 9月にあった卒論の中間報告で L^AT_EX + Beamer を使い苦しんだ
- 静的型付き言語は昔から好きだった
 - もっと使いやすい組版言語はないのか ⇒ SAT_YSF_I
- 第一印象は「括弧多いな、自然に書けるようになるのかな」
 - すぐに慣れ以後どっぷり浸かる

とりあえず使ってみる

- とりあえず研究室内の発表で $\text{SAT}_{\text{Y}}\text{SF}_{\text{I}}$ + $\text{SL}_{\text{Y}}\text{DIF}_{\text{I}}$ を使ってみた
 - 今使っているテーマは元々その時に作成したもの
- 執筆体験が良い
 - $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ のスライドと遜色ないものが作れる
 - ADT とパターンマッチで木構造の絵が描けることに感動

⇒ 卒論を $\text{SAT}_{\text{Y}}\text{SF}_{\text{I}}$ で書きたい

卒論を SATySF_Iで書くために

- 学科の規則的には問題なさそう
 - PDF で出力できれば OK
 - ページ数以外の規定なし
- 既存の環境 (VS Code) は長文を書くにはつらそう
 - 保存時にビルドを自動でしてほしい
 - monaqa さんの [Language Server](#) を使いたい

⇒ SATySF_I Workshop の開発へ

SATySF_I Workshop の開発

wraikny さんが作成されていた既存の VS Code 拡張をベースに、以下の機能追加・改善を実施
12 月に SATySF_I Advent Calendar で公開

- ビルド機能
 - ショートカットキーによるビルド・保存時のビルド
- 型チェック機能
 - 保存時やタイプ時に `satysfi` コマンドを呼び出し出力をパースして表示
 - 元からあったもののパフォーマンス等を改善
- Language Server のサポート
 - 単に受け口を作っただけ

その他 lint や format の自動化、テスト追加なども（詳細は [Qiita 記事](#) 参照）

卒論を SATySF_Iで書く

卒論は実際に SATySF_I Workshop を使って執筆した

使用させていただいたライブラリ

- [abenori/satysfi-class-jlreq](#) ... クラスファイル
- [monaqa/satysfi-easytable](#) ... 表組
- [monaqa/satysfi-enumitem](#) ... 箇条書き
- [monaqa/satysfi-figbox](#) ... 画像挿入
- [namachan10777/BiByFi](#) ... 文献管理
- [puripuri2100/satysfi-code-printer](#) ... ソースコード挿入

SATySF_Iで卒論を書いてよかったこと

- エラーメッセージがわかりやすい
 - 静的型に守られているという安心感
- 「ちょっとした拡張」がやりやすい
 - その文書限りのコマンドを作る精神的ハードルが低い
- Language Server が快適
- Lua^AT_EX よりコンパイルが速い

ちょっとしたコマンドの例

画像にキャプションを付けるコマンド

```
let with-caption caption figbox =  
  vconcat ? :align-center [  
    figbox;  
    gap 10pt;  
    textbox caption;  
  ]
```

のようなものを定義しておくと

```
+fig-center(  
  include-image 400pt `satysfi.jpg`  
  |> with-caption {\SATySFI; のロゴ}  
);
```

のように使える



Fig.1 SAT_ySF_Iのロゴ

SATySF_Iで卒論を書いて大変だったこと

問題 1 ライブラリの種類が少ない

⇒ その場その場でライブラリを作りつつ執筆

- figbox に自動で番号付けされるキャプションを付けたい ⇒ 前述の方法を拡張して作成
- 擬似コードを書きたい
 - その時点では enumitem を利用して作成

```
let-block +While cond inner =  
  '<  
    +EnumitemAlias.item({\bold{while}\ #cond; \bold{do}})(inner);  
    +EnumitemAlias.item({\bold{end while}})<>  
  >
```

- 後に独立したライブラリとし、テーマの切り替えなど機能追加 ⇒ [satysfi-algorithm](#)

SATySF_Iで卒論を書いて大変だったこと

問題 2 ビルドが終わらない

文章を書き進めていくに連れ、ビルドが `satysfi` コマンドの実行 1 回では終わらなくなってしまった

⇒ 浮動な画像（ページのヘッダに配置される画像）を多く挿入していたのが原因

一度安定すれば `satysfi-aux` ファイルを元に 1 回で組版できるが、文章の前方を編集するとやり直しになってしまうことも

まとめ

- SATySF_I を使うと分かりやすいエラーメッセージと Language Server による支援を存分に受けられる
- ライブラリが少ないのは大変だが、「困ったときは自分で作る」の精神があれば意外と乗り切れる
- （学部・学科の規定等があれば要確認）

卒論は SATySF_I で書ける！

SATySF_I関連の近況

- SATySF_I Workshop は鋭意開発中
 - そろそろまたリリースします（型チェック機能周辺の改善）
 - SATySF_I 0.1.0 の対応も
- SATySF_I Algorithm も 0.1.0 対応させたい
 - L^AT_EX の algorithmicx 向けのコードをテキストモードで吐けるようにしたい