

SATySF_I-formatter

usagrada

目次

1. formatter の install 方法	2
2. formatter の使い方	2
3. 開発者の方	2
3.1. Pull Request	3
3.2. issue	3
4. format	4

SATySF_I を使うに当たって、formatter が無いのが不便だったので、format をするためのツールを作りました。^{*1*2}

1. formatter の install 方法

以下のどちらかの方法で、入れることができます。ターミナルに以下のコマンドを打ち込んでください。^{*3}

```
cargo install --force --git https://github.com/usagrada/satysfi-formatter. ↵  
↳ git
```

```
git clone https://github.com/usagrada/satysfi-formatter.git  
cd satysfi-formatter  
cargo install --path .
```

2. formatter の使い方

```
satysfi-fmt $input -o $output
```

output を指定しなかった場合、コマンドラインの標準出力に結果が表示されます。

3. 開発者の方

release build でない場合、src/visualize.rs にある関数が呼び出されるようになってお

1 このドキュメントは format のテストも兼ねて書いています。

2 SATySF_I の文法に習熟している訳ではないため、parser を元に復元するという手法によりフォーマットを実現おり、容易にビルドが失敗するようになります。(22/3/11 現在)

3 --force は無くても入りますが、既にインストールしている場合、最新のデータにアップデートするために同じコマンドを使用できます。

り、ファイルの構造を確認できるようになってます。

```
cargo run -- $input
```

lib.rs の format を開始地点とし、コードから satysfi-parser で CST 化し、文字列に戻して結合しています。現状では、かなり愚直な実装をしている + 一部のみしか対応していない (コメントが消去される、改行入れて欲しいのに消える、etc.) ため、修正等があれば、Pull Request や Issue をお願いいたします。issue でいただく場合、期待するフォーマットのテストをいただけるとスムーズに対応ができると思います。その際、実際にそれがコンパイル可能である必要はありません。

3.1. Pull Request

実装した部分のテストケースを書いていただいてから、プルリクエストをいただけると幸いです。その際、src/tests 以下でしたら何処に書いていただいても構いません。(現状テストコード用のファイルをテキストウに書いてしまっているため、整備できていません。)

3.2. issue

以下にサンプル (src/tests/common.rs test1 と同じ) を載せておきます。r#" の内部に書かれたテキストはスペースや改行を含め全てそのまま出力されるため、スペース数改行数等の違いにより、テストが容易に落ちます。

```
#[test]
fn test1() {
    // format 前のテキスト
    let text = r#"@import: hello
@require: local

document(|title = {hello}|)'<+p{hello world}>"#;

    // 期待されるテキスト
    let expect = r#"@import: hello
@require: local

document(|title = {hello}|)'<
```

```
+p { hello world }  
  
>  
"#;  
    test_tmpl(text, expect);  
}
```

4. format

indent は 4 です。(そのうち引数で管理できるようにします) ブロックコマンド同士は改行により結合されます。