

計算論か集合論か Web に関する何か

Hiromi ISHII

筑波大学数学専攻 博士後期二年

Monday 5th March, 2018

~~計算論か集合論か Web に関する何か~~

Hiromi ISHII

筑波大学数学専攻 博士後期二年

Monday 5th March, 2018

SAT_YSFI を使ってみた

Hiromi ISHII

筑波大学数学専攻 博士後期二年

Monday 5th March, 2018



参照先：<http://bit.ly/2I21pS0>

自己紹介と今日の話題

- 石井大海（いしい・ひろみ）
- 筑波大数学専攻 博士後期二年
- 今日の話 : SATySFI (<https://github.com/gfngfn/SATySFI>)
 - ▶ 謙訪敬之氏が開発中の新しい組版システム.
 - ▶ L^AT_EX や InDesign の競合
 - ▶ 未踏プロジェクトにも採択された今大注目のプロダクト

SAT_YSF_I の特徴

- エラーのわかりやすさと拡張の容易さが売り
 - ▶ 静的型付き関数型言語である
 - ▶ L^AT_EX は結構直接的な項書き換え系，手の込んだマクロを実装しようとすると簡約過程に関する知識が不可欠
 - ▶ SAT_YSF_I は具体的に型やパターンが合わない部分を提示
- 数式の組みにも対応.

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

コード例

```
+listing {  
    * 石井大海 (いしい・ひろみ)  
    * 筑波大数学専攻 博士後期二年  
    * 今日の話 : \emph{\SATySFi;}  
      (\url(`https://github.com/...`));  
    ** 謙訪敬之氏が開発中の \emph{新組版システム} .  
    ** \LaTeX; や InDesign の競合  
}
```

ところで

この
スライドも
SATYSFI 製
です

Steamer: Making Slides with SAT γ SFI

- SAT γ SFI はまだ黎明期でスライドの標準パッケージはない
 - ▶ じゃあ作ろう！
- ネタに困っていた 3/2(金) 17:30 頃、思い立って実装開始
 - ▶ コードやサンプルを読みつつ 3/3(土) 05:00 AM 頃には最低限の機能が実装完了
 - ▶ 深夜なのに諏訪氏に直接訊くとすぐに答えが返ってくる
- 「拡張・実装が容易」という看板はある程度真実といえそう

SATySFI のプログラミング

- 基本的に ML ベースの文法
- 型推論が強力
- 高階関数がちゃんと関数として振る舞う（項書換えの挙動を意識する必要がない）ので楽
- 黎明期なのでバグもまだある（それはそう）

コード例

```
let-inline ctx \textsuperscript inner =
let ctx-sscr =
  ctx |> set-font-size (get-font-size ctx *' .7)
    |> set-manual-rising (get-font-size ctx *' .3)
in read-inline ctx-sscr inner
```

ちなみにこのなんちゃって自動シンタックス・ハイライトも自分でつくりました。

コード例 2 : 高階関数

```
let font-size-modify ctx scale inner =
  let ctx-size =
    ctx |> set-font-size (get-font-size ctx *' s-
cale)
  in read-inline ctx-size inner

let-inline ctx \small = font-size-modify ctx .75
let-inline ctx \LARGE = font-size-modify ctx 5.
let-inline ctx \Large = font-size-modify ctx 3.
```

SATySFI の型システム

- インラインボックスを組合せてブロックボックスを作る.
 - ▶ インラインボックス = 文字組み方向の文字列 + α . 文字列の折り返し情報やフォント情報などを持つ.
 - ▶ ブロックボックス = 段落等. 分割されページに割付.
- 文脈によってボックスの幅・高さやフォントが決まる.
- 他の基本型：文字列， 整数， 浮動小数点数， 長さ， インラインテキスト， 数式など.
- 関数型とバリエントと直積型を持つ.

書く上で覚えればいいこと

- インライン・テキスト ≈ 文章の断片 = { } で囲まれた奴
- ブロック・テキスト ≈ 段落の文章 = < > で囲まれた広い奴
- 文字列 ≈ 何の付加情報もない生の文章 = ' ' で囲まれたやつ
- 整数 = 1, 5, 0, -2 など. 演算は普通の + - * < などを使う.
- 浮動小数点数 = 1., .24, 2.5 など. 整数とは厳密に区別され, 演算は後ろに . を付けた +., *. などを使う.
- 長さ = 文字のサイズや間隔を表す = 12pt, 3mm とか. 演算には ' が末尾につく.

型と数値にまつわるエトセトラ

- 数の型が分かれている！これは素晴らしい！
 - ▶ T_{EX} は長さやカウンタを数値に流用 → エラーの温床！
- ML の伝統を受け継いでおり、 上述の通り整数・浮動小数で違う演算子を使う必要がある。
 - ▶ 好みの問題（私は Haskeller なので数値の演算はオーバーロードしたいが、それはそれでつらい）
- リスト型や option 型（Haskell の Maybe）型などもある
 - ▶ これも T_{EX} では項書き換えで頑張るしかない

エラー例

```
! [Type Error] at line 18, character 31 to line 20, character 5:  
this expression has type  
  (|keywords : string list|),  
but is expected of type  
  (|keywords : string list; primitives : string list|).  
This constraint is required by the expression  
at line 15, character 6 to line 17, character 9.
```

(ちゃんとバリエントの部分型付けもやってくれる)

とはいえ SATySFI も発展途上

- フォントと文字情報のマップには未対応（これから）
 - ▶ 文字のコピーや検索はまだできない
- 一部フォント（ヒラギノ, Latin Modern 等）扱いやページ送り clear-page にバグ
- ブロック埋めるやつやパターン網羅性チェックとか欲しい
- (コンパイラを拡張する時に構文木が型付きだと嬉しい)
- (~~ML-like 構文が嫌い！ Haskell 風にしろ！ とか処理系も Haskell で書かせろ！~~)

まとめ

- SATYSFI は L^AT_EX のような組版言語
 - ▶ 関数型言語で、項書換え系の T_EX 族より拡張が容易
 - ▶ このスライドの機能も大枠は半日、機能は三日で作れた
 - ▶ エラーメッセージもわかりやすい（一部未対応）
- まだまだ発展途上
 - ▶ 構文の説明がもう少しほしい
 - ▶ 機能は足りないけど開発者は意欲的
 - ▶ 緩やかなコミュニティが形成されつつある
 - ▶ 貢献するなら今！

御清聴
ありがとうございました