SugarTeX

SugarTeX is a more readable LaTeX language extension and a transcompiler to LaTeX.

See PDF version of this documentation - it nicely renders all Unicode characters and LaTeX example at the end. See original markdown version here.

Contents

- Command line interfaces
- Tweaking SugarTeX
- SugarTeX replacements and operators
    - Math delimiters
    - Brackets
    - Simple pre-replacements
    - Superscripts and Subscripts
    - Regular expressions pre-replacements
    - Nullary operators
    - Prefix unary operators
        - Styles
        - Styles with special brackets
        - Greedy prefix unary operators
        - Standard prefix unary operators
    - Postfix unary operators
    - Center binary operators
        - Matrices

        - General fractions without bars
        - Greedy center binary operators
        - Standard center binary operators
    - Regular expressions loop replacements
    - Regular expressions post-replacements
    - Simple post-replacements
    - Escapable characters
- Examples

Command line interfaces

1. `sugartex`:
    - sugartex reads from stdin and writes to stdout,
    - `sugartex TO` - run Pandoc filter that iterates over math blocks,
    - `sugartex --kiwi` - same as above but with kiwi flavor,
2. `pre-sugartex`:
    - pre-sugartex reads from stdin and writes to stdout,
    - `pre-sugartex` - replace `` ` `` with `$` only,
    - `pre-sugartex --all` - replace everything with regexp,
    - `pre-sugartex --kiwi` - same as above but with kiwi flavor.

Panflute scripts are also installed so you can use it in default Panflute automation interface in metadata or in it's CLI wrapper from pandoctools:

- `panfl sugartex_panfl --to markdown`,
- `panfl sugartex_kiwi -t markdown`.

Examples. Windows:

```
chcp 65001 > NUL
set PYTHONIOENCODING=utf-8
```

```
type doc.md | ^
pre-sugartex | ^
pandoc -f markdown --filter sugartex -o doc.md.md
```

Unix:

```
export PYTHONIOENCODING=utf-8

cat doc.md | \
pre-sugartex | \
pandoc -f markdown --filter sugartex -o doc.md.md
```

Or splitting Pandoc reader-writer:

```
chcp 65001 > NUL
set PYTHONIOENCODING=utf-8

type doc.md | ^
pre-sugartex | ^
pandoc -f markdown -t json | ^
sugartex --kiwi | ^
pandoc -f json -o doc.md.md
```

Tweaking SugarTeX

SugarTeX is written in python and has a tweakable architecture. As you can see in this filter tweaks can be made in between:

```
sugartex = SugarTeX(delay=True)
...
sugartex.ready()
```

Attributes of instance of `SugarTeX` class can be changed. See them in defining of `SugarTeX` class and in it's `__init__` method here. List of attributes:

- `.brackets`
- `.brackets_types`
- `.simple_pre`
- `.superscripts`
- `.subscripts`
- `.regex_pre`
- `.null_ops` (class `NullOps`)
- `.pref_un_ops` (class `PrefUnOps`), including:
  - `.styles` (class `Styles`)
  - `.other_styles` (class `OtherStyles`)
  - `.pref_un_greedy` (class `PrefUnGreedy`)
- `.postf_un_ops` (class `PostfUnOps`)
- `.bin_centr_ops` (class `BinCentrOps`), including:
  - `.matrices` (class `Matrices`)
  - `.bin_centr_greedy` (class `BinCentrGreedy`)
- `.loop_regexps`
- `.regex_post`
- `.simple_post`
- `.escapes`

SugarTeX replacements and operators

Many replacements use amsmath macros.

Math delimiters

In default use-case SugarTeX first preprocesses text replacing `˴` with `$` (modifier letter low grave accent U+02CE). Can be escaped: `\˴`

SugarTeX Completions for Atom:

- `˴` ← `_``.

Brackets

Independently replace brackets:

- `˳(` ⟶ `\left({` and `)˳` ⟶ `}\right)` (modifier letter low ring U+02F3),
- `˳˳(` ⟶ `\bigl(` and `)˳˳` ⟶ `\bigr)`,
- `˳ˌ(` ⟶ `\Bigl(` and `)˳ˌ` ⟶ `\Bigr)`,
- `ˌˌ(` ⟶ `\biggl(` and `)ˌˌ` ⟶ `\biggr)`,
- `ˌˌ(` ⟶ `\Biggl(` and `)ˌˌ` ⟶ `\Biggr)` (modifier letter low vertical line U+02CC).

Instead of `(` and `)` can be other brackets:

- `[` ⟶ `[` and `]` ⟶ `]`,
- `(` ⟶ `(` and `)` ⟶ `)`,
- `{` ⟶ `\{` and `}` ⟶ `\}`,
- `|` ⟶ `\vert` (box drawings light vertical U+2502, for math in markdown tables),
- `|` ⟶ `\vert`,
- `‖` ⟶ `\Vert` (double vertical line U+2016),
- `˂` ⟶ `.` and `˃` ⟶ `.` (modifier letter low left/right arrowhead U+02F1/U+02F2),
- `⟨` ⟶ `\langle` and `⟩` ⟶ `\rangle` (mathematical left/right angle bracket U+27E8/27E9),
- `⌊` ⟶ `\lfloor` and `⌋` ⟶ `\rfloor` (left/right floor U+230A/U+230B),
- `⌈` ⟶ `\lceil` and `⌉` ⟶ `\rceil` (left/right ceiling U+2308/U+2309.

SugarTeX Completions for Atom:

Use these shortcuts for fast Unicode typing in Atom:

- ◌̥ ⟵ `\&`,
- ◌̥ ⟵ `\_o\small`,
- ◌̍ ⟵ `\_'\small`.
- ◌̀ ⟵ `\|`,
- ◌̀◌̀ ⟵ `\||`,
- ◌̬ ⟵ `\_<`,
- ◌̬ ⟵ `\_>`,
- ◌̬◌̬ ⟵ `\_<>`,
- ⟨ ⟵ `\<\big`,
- ⟩ ⟵ `\>\big`,
- ⟨⟩ ⟵ `\<>\big`,
- ⌊ ⟵ `\lfloor`,
- ⌋ ⟵ `\rfloor`,
- ⌈ ⟵ `\lceil`,
- ⌉ ⟵ `\rceil`.

Simple pre-replacements

- ∛ ⟶ `3√` (cube root U+221B),
- ∜ ⟶ `4√` (fourth root U+221C),
- ◌ ⟶ `\,` (thin space U+2009).

SugarTeX Completions for Atom:

- ◌ ⟵ `\,` (thin space),
- ◌ ⟵ `\],[` (thin space),
- √ ⟵ `\^1/2`,
- ∛ ⟵ `\^1/3`,
- ∜ ⟵ `\^1/4`.

Superscripts and Subscripts

Groups of superscript Unicode characters like `¹²³` are replaced with `^{123}`. Unless they are escaped with `\` or followed by `√`:

- `\¹²³√` ⟶ `¹23√` (square root U+221A),
- `\¹²³` ⟶ `¹^{23}`,
- `¹²³ᵃᵇᶜ` ⟶ `^{123abc}`.

Same is for groups of subscript Unicode characters:

- `\₁₂₃` ⟶ `₁_{23}`.
- `₁₂₃ₖₗₘ` ⟶ `_{123klm}`.

List of supported characters can be found in the beginning of the SugarTeX source code.

UPDATE

Now `‹›` and `⌜⌝` from Styles with special brackets end up inside `_{}`/`^{}`, like: `A‹ₐₑ›` ⟶ `A_{‹ae›}`. Does not work if there are non-subscript/superscript characters inside `‹›`/`⌜⌝`, like: `A‹ᵃe›` ⟶ `A‹^{a}e›`.

SugarTeX Completions for Atom:

- `₁` ⟵ `\_1`,
- `ₐ` ⟵ `\_a`,
- `¹` ⟵ `\^1`,
- `ᵃ` ⟵ `\^a`.

Regular expressions pre-replacements

Nothing. But can be tweaked.

Nullary operators

Big operators replacements:

- ∑ ⟶ `\sum` (n-ary summation U+2211),
- ∑⁚ ⟶ `\sum\nolimits`,
- ∑⁝ ⟶ `\sum\limits` (braille pattern dots-48 U+2888).

Supported symbols for limits:

- ⁝, ⁚ ⟶ `\limits` (braille pattern dots-48/dots-17 U+2888/U+2841),
- ⁚, ⁝, ⁝ ⟶ `\nolimits` (braille pattern dots-23/dots-56 U+2806/U+2830).

Supported big operators:

- ∑ ⟶ `\sum`,
- ∏ ⟶ `\prod`,
- ∫ ⟶ `\int`,
- ∬ ⟶ `\iint`,
- ∭ ⟶ `\iiint`,
- ⨌ ⟶ `\iiiint`,
- ∮ ⟶ `\oint`.

Who knows what I was thinking about by adding them here instead of Regular expressions replacements…

SugarTeX Completions for Atom:

- ⁝ ⟵ `\:\big`,
- ⁝ ⟵ `\:\small`,
- ∑ ⟵ `\sum`,
- ∏ ⟵ `\prod`,
- ∫ ⟵ `\int`,
- ∬ ⟵ `\iint`,
- ∭ ⟵ `\iiint`,
- ⨌ ⟵ `\iiiint`,

- `∮` ← `\oint`.

Prefix unary operators

Styles

Text inside standard brackets (`()`, `[]`, `{}`) with special prefix is replaced with style operator. For example:

`[ʳtext]` or `[^{r}text]` ⟶ `\mathrm{text}`.

First SugarTeX finds opening part like `[^{r}` then searches for the first non-escaped closing part `]` that is not inside `{}` or `‹›` – SugarTeX counts opening and closing `{}‹›` (`‹›` would later be replaced with `{}` so both are counted together). For example:

`(ʳsome{te)(t})` ⟶ `\mathrm{some{te)(t}}`.

List of available styles:

- `{ʳtext}` / `{^{r}text}` ⟶ `\mathrm{text}` (**math regular**),
- `{ⁱx}` / `{^{i}x}` ⟶ `\mathit{x}` (**math italic**),
- `{ᵇx}` / `{^{b}x}` ⟶ `\mathbf{x}` (**math bold**),
- `{ᵝx}` / `{^{β}x}` ⟶ `\boldsymbol{x}` (**math bold italic**),
- `{ᵐtext}` / `{^{m}text}` ⟶ `\mathtt{text}` (**math monospace**),
- `{ᶜA}` / `{^{c}A}` ⟶ `\mathcal{A}` (**math calligraphic**,
  no cyrillic support, see Monotype Corsiva),
- `{ᵗtext}` / `{^{t}text}` ⟶ `\text{text}` (**text**),
- `{ᵗⁱtext}` / `{^{ti}text}` ⟶ `\textit{text}` (**text italic**),
- `{ᵗᵇtext}` / `{^{tb}text}` ⟶ `\textbf{text}` (**text bold**),
- `{ᵗᵝtext}` / `{^{tβ}text}` ⟶ `\textit{\textbf{text}}` (**text bold italic**),
- `{⃗x}` / `⟦x}` ⟶ `\mathbf{x}` (**vector bold notation**,
  combining right arrow above U+20D7, first one is ‘space’ + ⃗),
- `{⠒x}` / `{⠒ x}` ⟶ `\mathbf{x}` (**vector bold notation**,
  braille pattern dots-45/dots-12 U+2818/U+2803 [right upper 2/left upper

2]),

- `{⃛A}` / `{⃜A}` ⟶ `\mathbf{A}` (**matrix bold notation**, braille pattern dots-124/dots-1245 U+280B/U+281B).

SugarTeX Completions for Atom:

- `⃗` ⟵ `\^->`,
- `⃛` ⟵ `\^:`,
- `⃜` ⟵ `\^::`,
- `⃝` ⟵ `\^:.\rot`.

Styles with special brackets

- `‹ᵝtext›` / `‹^{β}text›` ⟶ `\textit{\textbf{text}}` (**text bold italic**),
- `‹ⁱtext›` / `‹^{i}text›` ⟶ `\textit{text}` (**text italic**),
- `‹ᵇtext›` / `‹^{b}text›` ⟶ `\textbf{text}` (**text bold**),
- `‹text›` ⟶ `\text{text}` (**text regular**, single left/right-pointing angle quotation mark U+2039/U+203A),
- `˽text˽` ⟶ `\mathrm{text}` (**math regular**, modifier letter begin/end high tone U+02F9/U+02FA).

SugarTeX Completions for Atom:

- `‹` ⟵ `\<`,
- `›` ⟵ `\>`,
- `‹›` ⟵ `\<>`,
- `˽˽` ⟵ `\^r\small`.

Greedy prefix unary operators

- `{∈ smth}` / `˹∈ smth˺` ⟶ `\begin{cases} smth\end{cases}` (**piecewise**, element of with long horizontal stroke U+22F2).

```
``
```

```
  ｡|x|｡ = {∈  x｡ ‹if› x≥0 ¦
             -x｡ ‹if› x<0 }
``
```

SugarTeX finds non-escaped `{∈` or `‹∈` first then searches for non-escaped `}` or `›` that is not inside `{}` or `‹›` – SugarTeX counts opening and closing `{}‹›` (`‹›` would later be replaced with `{}` so both are counted together).

SugarTeX Completions for Atom:

- `∈` ⟵ `\-e`,
- `∈` ⟵ `\-E`.

Standard prefix unary operators

- `‹matrix a` ⟶ `\begin{matrix} a`
  (left-pointing curved angle bracket U+29FC),
- `👻 A² a` ⟶ `\vphantom{A^2} a`
  (**invisible characters that adjust height**, ghost U+1F47B),
- `→‾ text a` ⟶ `\xrightarrow{text} a`
  (**arrow with text above that adjusts to the text length**, rightwards arrow U+2192, top square bracket U+23B4),
- `←‾‹long text› a` ⟶ `\xleftarrow{{long text}} a`
  (leftwards arrow U+2190).

SugarTeX finds non-escaped `‹ *` first (for example) then searches for a place before non-escaped `}`, `›`, space, newline or end of the string that is not inside `{}` or `‹›` – SugarTeX counts opening and closing `{}‹›` (`‹›` would later be replaced with `{}` so both are counted together).

SugarTeX Completions for Atom:

- `‹` ⟵ `\<\big2`,
- `›` ⟵ `\>\big2`,
- `‹›` ⟵ `\<>\big2`,
- `👻` ⟵ `\ghost`,

- ¯ ⟵ `\^^`,
- ¯ ⟵ `\^]\rot`,
- → ⟵ `\->`,
- ← ⟵ `\<-`.

Postfix unary operators

- `a x⃗` ⟶ `a \vec{x }` (**vector**,
  combining right arrow above U+20D7),
- `a x⃕` ⟶ `a \overrightarrow{x }` (**arrow above**,
  combining right harpoon above U+20D1),
- `a x^` ⟶ `a \widehat{x}` **warning**: works only if the next character after `^` is `}`, `>`, newline or end of the string,
- `a xˆ` ⟶ `a \hat{x}` (modifier letter circumflex accent U+02C6),
- `a x¯` ⟶ `a \bar{x}` (macron U+00AF),
- `a x‾` ⟶ `a \overline{x}` (overline U+203E),
- `a ẋ` ⟶ `a \dot{x}` (dot above U+02D9),
- `a ẍ` ⟶ `a \ddot{x}` (diaeresis U+00A8),
- `x + y+z⏞` ⟶ `x + \overbrace{y+z}`
  (top curly bracket U+23DE),
- `x + {y + z}⏟` ⟶ `x + \underbrace{{y + z}}`
  (bottom curly bracket U+23DF),
- `a x_` ⟶ `a \underline{x}`
  warning: works only if the next character after `_` is `}`, `>`, newline or end of the string (modifier letter low macron U+02CD),
- `a matrix〉` ⟶ `a \end{matrix}`
  (right-pointing curved angle bracket U+29FD),

SugarTeX finds non-escaped `*〉` first (for example) then before it searches for a place after non-escaped `{`, `<`, space, newline or start of the string that is not inside `{}` or `<>` – SugarTeX counts opening and closing `{}<>` (`<>` would later be replaced with `{}` so both are counted together).

In combination with styles:

When combining **one-character** postfix unary operators with styles the order in which operators are applied changes:

`[ᵇx⃗]` ⟶ `\vec{\mathbf{x }}`

SugarTeX Completions for Atom:

- `⃗` ⟵ `\^->`,
- `⃗` ⟵ `\^->\har`,
- `̂` ⟵ `\^\small`,
- `̄` ⟵ `\^_\small` (macron),
- `̄` ⟵ `\^-\small` (macron),
- `̄` ⟵ `\^_` (overline),
- `̇` ⟵ `\^.`,
- `̈` ⟵ `\^..`,
- `̃` ⟵ `\^}\rot`,
- `̰` ⟵ `\_}\rot`,
- `̲` ⟵ `\_`,
- `〈` ⟵ `\<\big2`,
- `〉` ⟵ `\>\big2`,
- `〈〉` ⟵ `\<>\big2`.

Center binary operators

Matrices

Family of `*matrix` amsmath macros is given by `¦̈` operator (broken bar U+00A6, braille pattern dots-124 U+280B):

`˹[a ˳b ¦̈ c ˳d]˺` ⟶

`\begin{bmatrix}a ˳b¦c ˳d\end{bmatrix}` ⟶

`\begin{bmatrix}a &b\\c &d\end{bmatrix}`

All brackets:

- `˹a ˳b ¦˙˙ c ˳d˺` ⟶ `...matrix...` (**no brackets**, modifier letter low left/right arrowhead U+02F1/U+02F2),
- `{a ˳b ¦˙˙ c ˳d}` ⟶ `...Bmatrix...` (**curly brackets**),
- `˹(a ˳b ¦˙˙ c ˳d)˺`/`{(a ˳b ¦˙˙ c ˳d)}` ⟶ `...pmatrix...`,
- `˹[a ˳b ¦˙˙ c ˳d]˺`/`{[a ˳b ¦˙˙ c ˳d]}` ⟶ `...bmatrix...`,
- `˹|a ˳b ¦˙˙ c ˳d|˺`/`{|a ˳b ¦˙˙ c ˳d|}`/ `˹|a ˳b ¦˙˙ c ˳d|˺`/`{|a ˳b ¦˙˙ c ˳d|}` ⟶ `...vmatrix...` (box drawings light vertical U+2502, for math in markdown tables),
- `˹‖a ˳b ¦˙˙ c ˳d‖˺`/`{‖a ˳b ¦˙˙ c ˳d‖}` ⟶ `...Vmatrix...` (double vertical line U+2016).

SugarTeX finds non-escaped binary operator separator `¦˙˙` first then:

- searches for a place after non-escaped `{` or `˹` that is not inside `{}` or `˹˺`,
- searches for a place before non-escaped `}` or `˺` that is not inside `{}` or `˹˺`,
- it also figures out bracket type properly,
- this way it finds two arguments (SugarTeX counts opening and closing `{}˹˺`, `˹˺` would later be replaced with `{}` so both are counted together).

SugarTeX Completions for Atom:

- `˳` ⟵ `\&`,
- `˳` ⟵ `\_o\small`,
- `|` ⟵ `\|`,
- `‖` ⟵ `\||`,
- `˹` ⟵ `\_<`,
- `˺` ⟵ `\_>`,
- `˹˺` ⟵ `\_<>`,
- `¦` ⟵ `\\`,
- `¦` ⟵ `\|/2`,
- `˙˙` ⟵ `\^:.\rot`.

General fractions without bars

Fractions works almost the same as Matrices - they add brackets and stack arguments: first arg is atop of the second arg. But with dome differences:

- they use `¦:` or `¦:` as a separator (broken bar U+00A6, braille pattern dots-45 U+2818 / dots-12 U+2803),
- cannot handle more than one line break (so two args only),
- they use `\genfrac` amsmath macro,
- they can have size modifiers after `¦:`:
  - `ᵈ`/`^{d}` - display mode,
  - `ᵗ`/`^{t}` - text mode,
  - `ˢ`/`^{s}` - smaller,
  - `ˣˢ`/`^{xs}` - extra small,
- left and right brackets can be different.

Examples:

- `˹(x¦:ᵗy)˺`,
- `˹[x¦:y]˺`,
- `{x¦:y}` (**curly brackets**),
- `˹x¦:y˺` (**no brackets**, modifier letter low left/right arrowhead U+02F1/U+02F2),
- `˹|x¦:y|˺`, `˹|x¦:y|˺` (box drawings light vertical U+2502, for math in markdown tables),
- `˹‖x¦:ᵈy‖˺` (double vertical line U+2016).

Arguments search algorithm is the same as for matrices.

SugarTeX Completions for Atom:

- `|` ← `\|`,
- `‖` ← `\||`,
- `˹` ← `\_<`,
- `˺` ← `\_>`,
- `˹˺` ← `\_<>`,
- `¦` ← `\\`,

- `¦` ⟵ `\|/2`,
- `⁝` ⟵ `\^:`.

Greedy center binary operators

Arguments search algorithm is the same as for matrices (except it now does not have brackets).

1. `‹smth1 ¦⁝⁝ᵗ smth2›` ⟶ `\begin{smallmatrix}smth1¦smth2\end{smallmatrix}`, (Braille Pattern Dots-1245 U+281B).

```
`..(‹a .b ¦⁝⁝ᵗ c .d›)...`
```

2. `‹smth1 ¦⁝⁝ smth2›` ⟶ `\begin{array}smth1¦smth2\end{array}`, (Braille Pattern Dots-1245 U+281B).

```
``
.[‹                          ‹cccc|c›
    x₁₁ .x₁₂ .x₁₃ .… .x₁ₙ  ¦⁝⁝
    x₂₁ .x₂₂ .x₂₃ .… .x₂ₙ  ¦
     ⋮  . ⋮  . ⋮  .⋱ . ⋮   ¦
    xₚ₁ .xₚ₂ .xₚ₃ .… .xₚₙ  ›].
``
```

3. `‹smth1 ¦# smth2›` ⟶ `\begin{aligned}smth1¦smth2\end{aligned}`,

```
``
  .|x|. = .{‹ x. ‹if› x≥0  ¦#
             -x. ‹if› x<0  › ›.
``
```

4. ˹smth1 ¦␣ smth2˺ / ˹smth1 ¦␣ smth2˺ ⟶ `\substack{smth1¦smth2}`,
   (modifier letter shelf U+02FD / bottom square bracket U+23B5)

```
`` ∑ⁿ_{0≤i≤N ¦␣ 0≤j≤M} (ij)³ ``
```

5. ˹smth1 ¦␣ˡ smth2˺ / ˹smth1 ¦␣ˡ smth2˺ ⟶ `\begin{subarray}{l}smth1¦smth2\end{subarray}`,
   (modifier letter shelf U+02FD / bottom square bracket U+23B5)

```
`` ∑ⁿ_{0≤i≤N ¦␣ˡ 0≤j≤M} (ij)³ ``
```

Instead of ˡ (left) it can also be ᶜ (center) or ʳ (right).

SugarTeX Completions for Atom:

- ¨ ⟵ `\^::`,
- ␣ ⟵ `\__`,
- ⌴ ⟵ `\_]\rot`,
- ⎵ ⟵ `\_]\rot2`,
- ¦ ⟵ `\\`,
- ⫽ ⟵ `\|/2`.

Standard center binary operators

Fractions

- `x/y` ⟶ `\frac{x}{y}` (division slash U+2215),
- `1+x/y` ⟶ `\frac{1+x}{y}`,
- `1 + {x + z}/y` ⟶ `1 + \frac{{x + z}}{y}`,
- `x/ᵈy` ⟶ `\dfrac{x}{y}`,
- `x/ᵗy` ⟶ `\tfrac{x}{y}`,
- `x/ᶜy` ⟶ `\cfrac{x}{y}`,

- `x/ˢy` and `x/ˣˢy` are the same as `x/ᵗy` but smaller and use `\genfrac` macros. Bar thickness can be set this way: `{0.5px}x/ˢy`.

Roots, overset, underset

- `√64` ⟶ `\sqrt[]{64}` (square root U+221A),
- `⁶√64` ⟶ `\sqrt[6]{64}`,
- `1 + ⁶√64` ⟶ `1 + \sqrt[6]{64}`,
- `˽lim˽_x→0` / `˽lim˽_x→0` ⟶ `\underset{x→0}{˽lim˽}` (modifier letter shelf U+02FD / bottom square bracket U+23B5),
- `{x + … + x}⏞{k ‹times›}` ⟶ `\overset{{k ‹times›}}{{x + … + x}⏞}` (top square bracket U+23B4).

Binomial coefficients

- `(i¦ᶜn)` ⟶ `\binom{i}{n}`,
- `(i¦ᶜᵈn)` ⟶ `\dbinom{i}{n}` (display),
- `(i¦ᶜᵗn)` ⟶ `\tbinom{i}{n}` (text).

In this case SugarTeX finds non-escaped binary operator separator `¦ᶜ` first then searches for `(` and `)`. Other stop symbols do not work.

SugarTeX finds non-escaped binary operator separator (like `/`) first then:

- searches for a place after non-escaped `{`, `˹`, space, newline or start of the string that is not inside `{}` or `˹˺`,
- searches for a place before non-escaped `}`, `˺`, space, newline or end of the string that is not inside `{}` or `˹˺`,
- this way it finds two arguments (SugarTeX counts opening and closing `{}˹˺`, `˹˺` would later be replaced with `{}` so both are counted together).

SugarTeX Completions for Atom:

- `␣` ⟵ `\__`,
- `␣` ⟵ `\_]\rot`,
- `␣` ⟵ `\_]\rot2`,

- ¯ ← \^^,
- ¯ ← \^]\rot,
- / ← \/,
- √ ← \^1/2,
- ¦ ← \\,
- ¦ ← \|/2.

Regular expressions loop replacements

Nothing. But can be tweaked.

Regular expressions post-replacements

Nothing. But can be tweaked.

Simple post-replacements

- ¦ → \\ (broken bar U+00A6, this should be after other ¦ replacements),
- ˚ → & (modifier letter low ring U+02F3, this should be after brackets and other ˚ replacements),
- ˂ → { and ˃ → } (modifier letter low left/right arrowhead U+02F1/U+02F2),
- ˍ → _ (modifier letter low macron U+02CD),
- ↕^{d} → \displaystyle (up down arrow U+2195),
- ↕^{t} → \textstyle,
- ↕^{s} → \scriptstyle,
- ↕^{xs} → \scriptscriptstyle,

- Superscripts and Subscripts replacements give:
- `↕ᵈ` ⟶ `\displaystyle`,
- `↕ᵗ` ⟶ `\textstyle`,
- `↕ˢ` ⟶ `\scriptstyle`,
- `↕ˣˢ` ⟶ `\scriptscriptstyle`.

SugarTeX Completions for Atom:

- `¦` ⟵ `\\`,
- `¦` ⟵ `\|/2`,
- `˳` ⟵ `\&`,
- `˳` ⟵ `\_o\small`,
- `˂` ⟵ `\_<`,
- `˃` ⟵ `\_>`,
- `˂˃` ⟵ `\_<>`,
- `˗` ⟵ `\_`,
- `↕` ⟵ `\<->\rot`.

Escapable characters

All one-character replacements from:

- Prefix unary operators,
- Postfix unary operators,
- Center binary operators,
- Nullary operators,
- Simple pre-replacements,
- Simple post-replacements,

and `∈`, `›`, `ˀ`, `↕`, `ˌ`

(element of with long horizontal stroke U+22F2, single right-pointing angle quotation mark U+203A, modifier letter end high tone U+02FA, up down arrow U+2195, modifier letter low vertical line U+02CC)

are escapable with `\`.

Examples

```
``
⸢∇ × [⃗B] - 1/c ∂[⃗E]/∂t ⸣= 4π/c [⃗j] ¦#
               ∇ · [⃗E]\ ⸣= 4πρ       ¦
 ∇ × [⃗E] + 1/c ∂[⃗B]/∂t ⸣= [⃗0]      ¦
               ∇ · [⃗B]\ ⸣= 0         ⸣
,,,{#eq:max}

where ⸢[⃗B], [⃗E], [⃗j]: ℝ⁴ → ℝ³⸣ – vector functions of the form
⸢(t,x,y,z) ↦ [⃗f](t,x,y,z), [⃗f] = (f_ˈxˈ, f_ˈyˈ, f_ˈzˈ)⸣.
```

renders to:

$$\begin{aligned} \nabla \times \mathbf{B} - \frac{1}{c}\frac{\partial \mathbf{E}}{\partial t} &= \frac{4\pi}{c}\mathbf{j} \\ \nabla \cdot \mathbf{E} &= 4\pi\rho \\ \nabla \times \mathbf{E} + \frac{1}{c}\frac{\partial \mathbf{B}}{\partial t} &= 0 \\ \nabla \cdot \mathbf{B} &= 0 \end{aligned}, \qquad (1)$$

where $\mathbf{B}, \mathbf{E}, \mathbf{j} : \mathbb{R}^4 \to \mathbb{R}^3$ – vector functions of the form $(t, x, y, z) \mapsto \mathbf{f}(t, x, y, z)$, $\mathbf{f} = (f_\mathrm{x}, f_\mathrm{y}, f_\mathrm{z})$

(you can see it in the PDF version of this documentation).

TODO

- Add examples to every section,
- Add more examples here at the end.