

# Stylo: the diagram toolkit for SAT<sub>Y</sub>SF<sub>I</sub>

Yu Shimura

Stylo is a satisfactory graphics library which focuses on drawing geometrical diagrams in its entirety. Rendered diagrams are meant to be comparable to X<sub>Y</sub>-pic package of L<sup>A</sup>T<sub>E</sub>X, but the code looks readable and descriptive.

## 1. Concepts

TBW (some amazing art)

TBW

### 1.1. Abstract Objects

Datatypes described under this section are the essential ones, which represent abstract objects not tied to its actual appearance on the rendered paper.

#### 1.1.1. Point

TBW (illustration of the Big Dipper?)

A point—is the most primitive object in the world of graphics.

#### 1.1.2. Path

TBW

A path—that's all about diagrams. Stylo has its own implementation of path instead of SAT<sub>Y</sub>SF<sub>I</sub>'s built-in one, in order to implement a series of manipulation around path in depth. Constructed paths are transformed straightforward into built-in paths at the time of conversion into graphics.

#### 1.1.3. Position

TBW

A position—is a waypoint on the path. Some path operations require some positions on

the path to be specified by scalar values. In the term of specifying these positions, paths are always treated as if the starting point and ending point are connected to each other, regardless of being closed or not. That is:

- When specified values are negative, interpreted backwardly from the ending point of the path.
- When the absolutes of specified values exceed the total length of the path, interpreted cyclically through the path.

Positions can be specified in scale or length. That is:

- When in scale, interpreted assuming 0.0 and 1.0 is the starting point and the ending point of the path respectively.
- When in length, interpreted assuming 0cm is the starting point of the path.

## 1.2. Decorative Objects

In this section, style-related datatypes are shown in series.

- TBW

### 1.2.1. Line

TBW

TBW

### 1.2.2. Arrowhead

TBW

TBW

### 1.2.3. Label

TBW

TBW

## 1.3. Others

### 1.3.1. Terminology

Some special terms appear throughout this documentation.

## 2. Features

There's a set of utility functions for constructing, splicing, measuring and decorating paths. Example diagrams are following; see how it works.



TBW

### 2.1. Constructing paths

Paths are constructed in several ways; using predefined functions for idiomatic shapes, drawing from scratch, and composing them.

#### 2.1.1. Basic constructs

These constructs are just straightforward to SATySFy's built-in functions with a few exceptions.

**start** : point → pre-path

Starts a pre-path from a given point.

**start-with-tangent** : point → point → pre-path

Starts a pre-path from a given point, with a phantom tangent which may be referred by the subsequent construct such as **arc-to-for**. The first argument is a point representing the tangent from the second argument which is the starting point.

**line-to** : pre-path  $\rightarrow$  point  $\rightarrow$  path

Draws a line to a given point.

**bezier-to** : pre-path  $\rightarrow$  point  $\times$  point  $\times$  point  $\rightarrow$  path

Draws a Bézier curve to the last given point with the first two given control points.

**terminate** : pre-path  $\rightarrow$  path

Terminates a pre-path leaving it open.

**terminate-with-line** : pre-path  $\rightarrow$  path

Terminates a pre-path closing with a line.

**terminate-with-bezier** : point  $\times$  point  $\rightarrow$  pre-path  $\rightarrow$  path

Terminates a pre-path closing with a Bézier curve. Two points stand for control points.

## 2.1.2. Arcs

Stylo has a powerful set of constructs for drawing arcs in different ways.

**arc-around** : point  $\rightarrow$  float  $\rightarrow$  pre-path  $\rightarrow$  pre-path

Draws an arc around a given point, in radians.

**arc-aside** : length  $\rightarrow$  float  $\rightarrow$  pre-path  $\rightarrow$  pre-path

Draws an arc around a given point which is a given length away from the current point orthogonally to the current tangent, in radians. Positive length means the left direction.

**arc-to-for** : point  $\rightarrow$  point  $\rightarrow$  pre-path  $\rightarrow$  pre-path

Draws an arc to the first point, referring the current tangent and the second point as a forwarding tangent. As a result, the arc drawn is smoothly tangented to the current pre-path and also to the subsequent construct as long as it uses the identical point to the forwarding tangent as its starting tangent.

**arc-to-for-opposite** : point  $\rightarrow$  point  $\rightarrow$  pre-path  $\rightarrow$  pre-path

Same as **arc-to-for** with the exception of its opposed arc so tangented like a pair of needles. Not implemented yet.

## 2.2. Splicing paths

Manipulating constructed paths is the main way to produce complex paths.

### 2.2.1. Split

Paths can be splited at an arbitrary position.

**split-at** : position  $\rightarrow$  path  $\rightarrow$  path  $\times$  path

Splits a path at a given position. Just match-cased into **split-at-in-scale** or **split-at-in-length**.

**split-at-in-scale** : float  $\rightarrow$  path  $\rightarrow$  path  $\times$  path

Splits a path at a position specified in scale.

**split-at-in-length** : length  $\rightarrow$  path  $\rightarrow$  path  $\times$  path

Splits a path at a position specified in length.

### 2.2.2. Trim

Paths can be trimmed at arbitrary endpositions. 切り落としの始端位置から終端位置への向きがパスを逆行する場合、切り出されるパスの向きもそれに従う。Not implemented yet.

**trim-at** : position  $\times$  position  $\rightarrow$  path  $\rightarrow$  path

Trims a path at given endpositions. Just match-cased into **trim-at-in-scale** or **trim-at-in-length**.

**trim-at-in-scale** : float  $\times$  float  $\rightarrow$  path  $\rightarrow$  path

パス全体の長さに対して、パスの始端点を 0.0、終端点を 1.0 とする無次元数で切り落とし位置を指定し、パスの部分を切り出す。

**trim-at-in-length** : length  $\times$  length  $\rightarrow$  path  $\rightarrow$  path

パスの始端点を 0cm とする長さで切り落とし位置を指定し、パスの部分を切り出す。

### 2.2.3. Trace

Not implemented yet.

## 2.3. Measurement

Stylo provides a set of features that supports measuring distances, lengths, areas and angles amongst geometrical objects.

### 2.3.1. Distance

TBW

### 2.3.2. Length

Stylo's internal representations of lengths are all in centimeters. TBW

### 2.3.3. Angle

Angles are interpreted all in radians. TBW

## 2.4. Composite paths

Not implemented yet.

### 2.4.1. Boolean operations

Not implemented yet.