The `parallel` package

Masaki Waga

This is a SAT_YSF_I package providing typesetting of two column texts in parallel. This package is inspired by the L^AT_EX package with the same name.

1. Installation

You can install SAT_YSF_I parallel with opam and Satyrographos:

```
opam install satysfi-parallel
```

You can also tangle `parallel.satyh` from `parallel.org` using GNU Emacs by typing `C-c C-v t` (or equivalently, `M-x org-babel-tangle`) after opening parallel.org. Otherwise, you can tangle from command line as follows:

```
emacs --batch --eval "(require 'org)" --eval '(org-babel-tangle-file
"$<")'
```

2. Commands

The parallel package provides only `+parallel` command. The signature of `+parallel` is as follows.

```
direct +parallel : [float?; float?; block-text; block-text] block-cmd
```

The first and the second arguments of `+parallel` are the float values representing the ratios of the left and the right columns, respectively. These values are optional. The third and the forth arguments of `+parallel` are the contents of the left and the right columns, respectively.

3. Examples

The following shows a small working example of `parallel` package.

```
@require: stdja
@import: ./parallel

StdJa.document (|
  title = {Example usage of parallel};
  author = {Masaki Waga};
  show-title = true;
  show-toc = false;
|) '<
  +parallel <
    +p {
      Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua.
      Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat.
      Duis aute irure dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur.
      Excepteur sint occaecat cupidatat non proident, sunt in culpa qui
officia deserunt mollit anim id est laborum.
    }
  > <
    +p {
      ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。
ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの
卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はお
いしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしい
ぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞ
うの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵
はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおい
しいぞう。ぞうの卵はおいしいぞう。
    }
```

```
 >
>
```

You can also specify the ratios of the columns as follows. We note that their sum should be smaller than 1.

```
@require: stdja
@import: ./parallel

StdJa.document (|
  title = {Example usage of parallel};
  author = {Masaki Waga};
  show-title = true;
  show-toc = false;
|) '<
   +parallel ?:(0.7) ?:(0.28) <
     +p {
       Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua.
       Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat.
       Duis aute irure dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur.
       Excepteur sint occaecat cupidatat non proident, sunt in culpa qui
officia deserunt mollit anim id est laborum.
     }
  > <
     +p {
       ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。
ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵
はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいし
いぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。
ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵
はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいし
いぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。ぞうの卵はおいしいぞう。
```

```
ぞうの卵はおいしいぞう。
    }
  >
>
```

4. Dependency

`parallel` is independent of any external packages except for `pervasives` that is distributed with SATYSFI.

5. Contributing

The original source code of this package is written in `parallel.org` and the distribution is generated by tangling it with GNU Emacs. If you want to make a pull request on GitHub, please modify this file.

It is also welcome to make an issue with a suggestion of improvement, bug fix, and so on.

6. Implementation

The following shows the implementation of the parallel package.

```
@require: pervasives

module Parallel : sig

  direct +parallel : [float?; float?; block-text; block-text] block-
cmd
end = struct
    let-block ctx +parallel ?:left-ratio ?:right-ratio left-block right-
block =
      let left-ratio =
  match left-ratio with
  | None -> 0.48
  | Some(left-ratio) -> left-ratio in
      let right-ratio =
```

```
  match right-ratio with
  | None -> 0.48
  | Some(right-ratio) -> right-ratio in
      let left-column-width = (get-text-width ctx) *' left-ratio in
      let right-column-width = (get-text-width ctx) *' right-ratio in
      let margin = (get-text-width ctx) -' left-column-width -' right-
column-width in
      let br-left-pbox =
  embed-block-top ctx left-column-width (fun ctx -> read-block ctx left-
block) in
      let br-right-pbox =
  embed-block-top ctx right-column-width (fun ctx -> read-block ctx right-
block) in
      form-paragraph ctx (br-left-pbox ++ (inline-skip margin) ++ br-
right-pbox)
end
```