

FIT 2102 A2 Report  
Tho Jun Quan 33309027

I've designed 6 new data structures: Text, OrderedListItem, TableHead, TableRow, TableCell, and BodyEle.

BodyEle can consist of FreeText, Images, Footnote References, Blockquotes, quotes, Heading, Codeblocks, OrderedLists or Tables.

Most of the Parsers were constructed using do blocks, as I find them to be more readable.

A parser for FreeText would parse multiple Texts until a new line, by using the Alternative, FreeText parser prioritises modifiers, when it fails it parses the text as a PlainText.

PlainText uses someTill, parsing in characters that are not new lines, until end of line, or looking ahead to find a valid parse for modifiers or a new line. To chain the three functions mentioned at the end, void was used to remove the values parsed in the two parsers wrapped in lookahead.

I've created a helper function, textmodParser to parse some text mods with the same structure, (Italic, Bold, Strikethrough, Inline). This is done so that code is repeated.

I've also created a isPosInt parser, which parses positive integers, and used it in Footnote and FootnoteRef.

headingParser is a Parser for normal Heading it fmaps length function over the helper function count and then uses guard to validate the length, while a helper function is used for alternative headings, altHeadingParser, takes in a character and an int. Where the parser validates input based on the character and returns the text parsed and level wrapped in Heading. It is all chained together with Alternative.

codeParser parses using two manyTills, one to check for the end of first line and another to check for a new line with only three ``s. I've used a helper function eofOrnewline here to check if it is the end of input or a newline.

To parse ordered lists, I have a parser function that takes in an integer for the indents required and a parser to parse the numbering required. This parser parses an item, and once it's done tries to parse a sublist if it exists, using a helper function, which calls the main orderedlist parser, if fails returns an EmptyList value.

OrderedList items contains some text, followed by either a sublist or an empty text. This is done as the html output of orderedlists with sublists can be easily pattern matched.

Table works by parsing the table head and the separator, then parsing the table body if exists while validating if it's length is equal to the head and separator.

The tableRowParser, a parser for tableBody, takes in an integer in order to compare it with the number of tableCells parsed in order to validate it's length, i've.

I've also created a plainText for table, called tablePlainParser, which avoids parsing '|'. An issue I've face when parsing in table cells is to remove the trailing spaces in a table cell, this is due to the not using a separator, the whole row is parsed into the table cell function, including multiple cells. When my endInlineSpace helper function is called it only trims whitespaces for the end of the row.

The final ADT will consist of Body, which is a list of BodyEle. In markdownParser I trim leading and trailing whitespaces and parse multiple BodyEles using Alternatives. To trim trailing whitespaces I've created an endSpaces, that uses dropWhileEnd.

When converting my ADT to HTML, I used addIndent which takes in an integer, and adds ' ' chars based on the integer taken in.