writeup.md

MiniML Extension

Float Data Type

As the only extension of MiniML, the float data type was implemented to expand the range of operations available. In addition, float operators were also added to make all data types in MiniML unique in their concrete syntax.

Modifying the MiniML Parser

In order to add floats, `miniml_parse.mly` and `miniml_lex.mly` were modified. The code written is analogous to the default content so it was simple to follow patterns and successfully give birth to floats in `miniml_parse.mly` . For `miniml_lex.mly` , four new key-value pairs were added to the `sym_table` hashtable representing the float operators. Down the code, the regular expression pattern `digit+ '.' digit*` was added to represent floats as a digit followed by a `.` and, optionally, more digits.

Modifying Concrete Syntax in .ml Files

In `expr.ml` , `Fnegate` was added to `unop` type as a means to represent the negation operator. In similar fashion, `Fplus` , `Fminus` , and `Ftimes` were added to the `binop` type to represent the inferred float operators. Of course, `Float of float` must have been added to the `expr` type. Finally, two auxiliary functions were implemented in `evaluation.ml` using pattern-matching to negate floats and conduct binary operations between floats.

Testing

Testing was conducted to confirm that floats and their operators were rightly implemented. In particular, mixing operators between floats and integers was done to ensure correctness. Examples of testing are below.

```
<== 2. +. 3. ;;
==> Float(5.)
<== 2 +. 3 ;;
xx> evaluation error: Type error, values of type Num or Float were expected
<== 3. - 1. ;;
xx> evaluation error: Type error, values of type Num or Float were expected
<== 4.1415 -. 1. ;;
==> Float(3.1415)
```