Description of MiniML Float Extension

This short paper describes and documents my extension of floats to MiniML and how I implemented them.

First, I modified expr.ml, wherein I added a new variant, `Flt of float`, to the `expr` type and then to the `subst_aux`, `exp_to_concrete_string`, and `exp_to_abstract_string` functions.

Next, I modified evaluation.ml, wherein I added the variant `Flt` to the `eval_s_expr` and `eval_d_expr` functions. I did not add new binary operators to either of these later functions. Instead, I extended the existing functions (such as `Add`) with a case for the new variant `Flt`. My extension of MiniML is therefore weakly typed. I added extra failure checks to ensure users do not call binary operations on one `Flt` and another variant.

Last, I modified miniml_lex.mll and miniml_parse.mly by duplicating and renaming code concerning the variant `Num` as appropriate. I finally added a new rule token to miniml_parse.mly to process users' float input, as follows: `| digit+ '.' digit* as inum { let num = Float.of_string inum in FLOAT num }` .

The completed extension allows MiniML to REPL expressions such as `0.1 + 0.2 ;;`, `0.1 = 0.1 ;;` and `0.1 < 0.2 ;;`, among others.