主専攻実験[S-8] 関数プログラミング

第 2 回

情報科学類 202113564 三村潤之介

課題 2-2.

・上記の exp 型の定義を入力した上で、IntLit 1 や Plus(IntLit 1, IntLit 3) などを OCaml 処理系に入力し、どういう型であるかを求めよ。

以下のような出力となった。

```
# Plus(IntLit 1, IntLit 3);;
- : exp = Plus (IntLit 1, IntLit 3)
```

これは exp 型が返されていることを示している。

・上記の exp 型の定義に、引き算と割り算を表す式を 増やしてみなさい。

・E という exp 型の式をもらうと、Plus(E, IntLit 2) という式を返す関数を記述せよ。

https://github.com/mimunojun/functional_prog/tree/master/02

上記の GitHub レポジトリ内の data_structure.ml に記述する。

引き算、割り算はすでに与えられている足し算などと同様に実装した。

Plus 式を返す関数の実装については、exp_plus_two という名前で let 式を定義した。受け取る引数 e は exp で指定しているが、右辺の Plus 式の中は exp 型であるので、指定がなくとも型を推定し、同様の動作を行うだろうと考える。

課題 2-3.

・上記のプログラム eval1 に様々なテストデータ(exp 型の式)を与えて試しなさい。

1+2*3 を与えて試した結果を示す。

```
# eval1(Plus(IntLit 1, Times(IntLit 2, IntLit 3)));;
- : int = 7
```

正しく計算されていることが分かる。

・match 式の評価は上から順番に行われる。このことを確認するために、 以下のような(間違った)定義を行い、評価がどうおかしくなるか試しなさい。

読み込み時に以下の様な警告が出た。

```
16 |   | Times(e1,e2) -> (eval1 e1) * (eval1 e2)
       ^^^^^^^^^^^^
Warning 11: this match case is unused.
```

これは Times の match が不使用であることを示す。

実際に、Times を含むデータを評価させると、unknown expression として扱われる。

```
# eval1(Times(Plus(IntLit 2, IntLit 3), IntLit 2));;
Exception: Failure "unknown expression".
```

match 文では上から評価されていくため、このような記述では IntLit(),Plus()以外がすべて”_”で受け取られてしまい、Times もここに含まれてしまう。

・対象となる言語に、「引き算」と「割り算」を追加して、 eval1 がそれらも処理できるようにしなさい。 この場合、0 で割るとエラーとすべきであるので、どう対処するかも考えなさい。

data_structure.ml に記述する。

引き算、割り算(Sub(),Div()とする)は足し算などと同様に実装できる。0 で割る時のエラーについては、Div()の定義の先頭に、Div(e1, IntLit 0)のパターンでエラーになるようにすることで、いわゆる”divide by zero exception”を実装できると考えた。