Exercise 4.5

#

Extend the (untyped) functional language with infix operator “&&” meaning sequential logical “and” and infix operator “||” meaning sequential logical “or”, as in C, C++, Java, C#, or F#. Note that e1 && e2 can be encoded as if e1 then e2 else falseandthate1 || e2canbeencodedasif e1 then true else e2.Henceyouneedonlychangethelexerandparserspecifications, and make the new rules in the parser specification generate the appropriate abstract syntax. You need not change Absyn.fs or Fun.fs.

Solution:

The following infix operators was added to the lexer in: `FunLex.fsl`

```
    | "&&"            { AND }
    | "||"            { OR }
```

The following was added to the parser in: `FunPar.psy`

```
    %token AND OR
    | Expr AND   Expr                     { If($1, $3, CstB false) }   (*  added for 4.5
    | Expr OR    Expr                     { If($1, CstB true, $3 ) }   (*  added for 4.5
```