CHAPTER IV

CONCLUSION

We have created a new λ-calculus in order to describe a larger class of functions by adding patterns and by modifying some definitions in the original λ-calculus. These include terms, free and bound variables, substitution, changes of bound variables, contractions, and reductions. This new λ-calculus has been shown to possess the basic properties of the original λ-calculus. In particular, the new λ-calculus still satisfies the Church-Rosser theorem.

It is interesting to note that the definition of patterns we have used does not allow any variables to occur more than once in the same pattern. The patterns used in functional programming languages also have this restriction. The reason usually given in the case of functional programming is to avoid an implicit test for equality during pattern matching, since this simplifies the pattern matching algorithm. However, we have seen from our work that the restriction is needed for the proof of the Church-Rosser theorem.

However, in this new λ-calculus, some contractions, namely βγ-contraction and βδ-contraction, may not be decidable since we are not able to consider every possible substitution in finitely many steps. Nevertheless it is necessary to consider every substitution in the definitions of these contractions to make the new λ-calculus satisfy the Church-Rosser theorem.

A major open question concerns the semantics of the new λ-calculus. In particular, designing a semantic that is compatible with βδ-equality appears challenging, since the pattern matching used in β- and δ-contraction depends heavily on the syntax of the terms involved.