

# Summary of “Untyped Arithmetic Expressions”

CHRISTOPH MENDE

---

## ACM Reference format:

Christoph Mende. 2017. Summary of “Untyped Arithmetic Expressions”. 1, 1, Article 1 (April 2017), 1 pages.  
DOI: 10.1145/nnnnnnn.nnnnnnn

---

## 1 INTRODUCTION

The chapter “Untyped Arithmetic Expressions” from the Book “Types and Programming languages”[1] is about proofs about programming languages. To do these proofs, a simple, untyped programming language is used. This language consists of the terms *true*, *false*, and 0, the functions *succ* to return the successor of a number, *pred* to return the predecessor of a number, and *iszero* to return whether a number is zero, and the term *iftthenelset* where *t* is any term.

The proofs are done by induction which fits well, because the language is inductively defined. That is, any number is built by calling the successor function. This correlates to how induction works, where you prove that a theorem holds for any term by proving it for a base case (usually 0) and prove that if it holds for a specific case (*n*), it also holds for its successor (*n* + 1), thus proving that it holds for any *n*. The correlation here is that the inductive definition of the language (numbers are defined as successors) is similar to the second part of the inductive proof (prove that a theorem holds for a successor).

## REFERENCES

- [1] Benjamin C. Pierce. 2002. *Types and Programming Languages* (1st ed.). The MIT Press.