

6.005 — Software Construction on MIT OpenCourseWare (<https://ocw.mit.edu/ans7870/6/6.005/s16/>) |
OCW 6.005 Homepage (<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-005-software-construction-spring-2016/>)
Spring 2016

Reading 16: Recursive Data Types

Introduction

Part 1: Recursive Data Types

Part 2: Writing a Program with ADTs

Summary

Reading 16: Recursive Data Types

Software in 6.005

Safe from bugs	Easy to understand	Ready for change
Correct today and correct in the unknown future.	Communicating clearly with future programmers, including future you.	Designed to accommodate change without rewriting.

Objectives

- Understand recursive datatypes
- Read and write datatype definitions
- Understand and implement functions over recursive datatypes
- Understand immutable lists and know the standard operations on immutable lists
- Know and follow a recipe for writing programs with ADTs

Introduction

In this reading we'll look at recursively-defined types, how to specify operations on such types, and how to implement them. Our main example will be *immutable lists*.

Then we'll use another recursive datatype example, *matrix multiplications*, to walk through our process for programming with ADTs.

Part 1: Recursive Data Types (recursive/)

Part 2: Writing a Program with ADTs (matexpr/)

Summary

Let's review how recursive datatypes fit in with the main goals of this course:

- **Safe from bugs**. Recursive datatypes allow us to tackle problems with a recursive or unbounded structure. Implementing appropriate data structures that encapsulate important operations and maintain their own invariants is crucial for correctness.

- **Easy to understand** . Functions over recursive datatypes, specified in the abstract type and implemented in each concrete variant, organize the different behavior of the type.
- **Ready for change** . A recursive ADT, like any ADT, separates abstract values from concrete representations, making it possible to change low-level code and high-level structure of the implementation without changing clients.