FCS CHAPTER I NOTES

Chinmay.Bhoir

August 16, 2018

Contents

Notes on chapter 1 of "Foundations of Computer Science" by Alfred Aho and Jeffrey Ullman

1 Data Models

1.1 Two aspects of data model

- Value that an object can consume. This is usually the static part, also called as 'Type system'
- Operations on data. This is usually the dynamic part.

1.2 'Data model' and 'data structure' are different

- Data model is an abstraction, while data structure is an implementation

- For example, 'list' is an abstraction (data model), while 'linked list' is an implementation (data structure).

1.3 Data models in programming

- The basic principle under which most programming languages deal with data is that each program has access to "boxes", which we can think of as regions of storage. Each box has a type, such as int or char.
- We may store in a box any value of the correct type for that box.
- We often refer to the values that can be stored in boxes as data objects
- In C, a list of integers can be represented by a data structure called a linked list in which list elements are stored in cells.
- Lists and their cells can be defined by a type declaration such as

```
typedef struct CELL *LIST;
struct CELL {
    int element;
    LIST next;;
}
```

- This declaration defines a self-referential structure CELL with two fields. The first is element, which holds the value of an element of the list and is of type int.
- The second field of each CELL is next, which holds a pointer to a cell.
- Note that the type LIST is really a pointer to a CELL. Thus, structures of type CELL can be linked together by their next fields to form what we usually think of as a linked list

1.4 Exercise

- Describe data model of your favourite text editor
- Describe data model of spreadsheet program

2 The C Data Model

2.1 The C Type System

- Static part of C data model, describes the types of values that data can have.
- Basic types in C:
  - Characters (char, signed char, unsigned char)
  - Integers (int, short int, long int, unsigned)
  - Floating-point numbers (float, double, long double)
  - Enumerations (enum)
- Type formation rules in C:
  - *Array Types* : Array whose elements are type T is denoted by

$$TA[n]$$

- *Structure Types* : Structure is a grouping of variables called *members* or *fields*.

```
struct S {
    T1 M1 ;
    T2 M2 ;
    ...
    Tn Mn ;
}
```

- *Union Types* : A union type allows a variable to have different types at different times during the execution of a program.

The declaration

```
union {
    T1 M1 ;
    T2 M2 ;
    ...
    Tn Mn ;
} x;
```

defines a variable x that can hold a value of any of the types T1 , T2 , . . . , Tn .

- *Pointer Types* : A variable of type pointer contains the address of a region of storage. Following is the declaration of variable p to be a pointer to a variable of type T.

$$T * p;$$

2.2 Exercise

- Give an example of a C data object that has more than one name.
- If you are familiar with another programming language besides C, describe its type system and operations.

3 Algorithms and the Design of Programs

Data models, their properties, and their appropriate usage is one pillar of computer science. Other, equally important, pillar is algorithms and and their associated data structures.

3.1 Software development process

- Problem definition and specification
- Design
- Implementation
- Integration and system testing
- Installation and field testing
- Maintenance

3.2 Programming style

- Modularize the program into coherent pieces
- Lay out the program so that its structure is clear

- Write intelligent comments to explain the program. Describe the data models, data structures used to represent them, and operations performed.
- Use meaningful names for procedures and variables
- Avoid explicit constants whenever possible.
- Avoid using global variables