Introduction to Modelling

- **Reality → computer model**
- process of **abstraction**
  - Properties (attributes) of objects/entities
- definition of **operations** on objects/entities

- **Mental "tools" and concepts**
  - **abstract data types** (ADTs)
  - **operations** on ADTs
  - **algorithms** - a sequence of operations (on ADTs)

ADTs & Modelling

ADT	Examples
set	General collections; RDBs; Boolean {T,F}; People waiting for a bus in Karlstad!
sequence	Dictionaries; instructions; algorithms; time; files; tables; & with restrictions, stack & queue
tree	Hierarchies (family trees; organisations; file systems; taxonomies); books; function calls;
undirected graph	Network systems (transport; computer; telephone)
directed graph	Flow systems; disease vectors; university courses; project task organisation (PERT)

Terminology

- Abstract Data Types (ADTs)
  - implementation independent - set of values + operations
- Abstract Data Structures (ADSs)
  - implementation independent

- Data Types
  - implementation (programming language) dependent
  - set of values + operations **(mathematical definition - domain + ops)**
- Data Structures
  - implementation (programming language) dependent
    - usually **arrays** and **pointers + structures**

Entities, Collections & Relations

- **Entity / Relationship & Attributes (E/R Model)**
  - abstraction from reality
  - **set of properties** which represent a real life object
  - **student ➔ (name, address, job, personal #, gender)**
- **Collection**
  - a **set** of entities having a common property
  - **all third year students**
- **Relation / relationship**
  - a **property** connecting two entities
  - **mother/daughter, distance between two cities**

Implementation - programming languages

- **Entity** - a data type (language or user defined)
- **Record** - a collection of **different** attributes
- **Array** - a collection of **like** entities
- **Relation** - a **reference** to another entity
  - **array index / pointer / name**
- **Other ideas** - **order & sorting**
  - to help **searching**

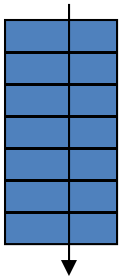
ADTs in Computer Science

Computer programs

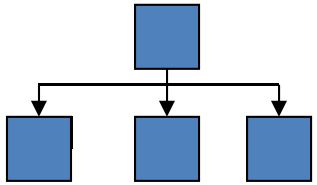
- **Sequence**

  c = f1(a, b);

  e = f2(c, d);

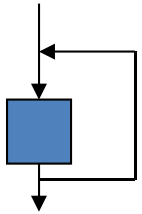


- **Decision**



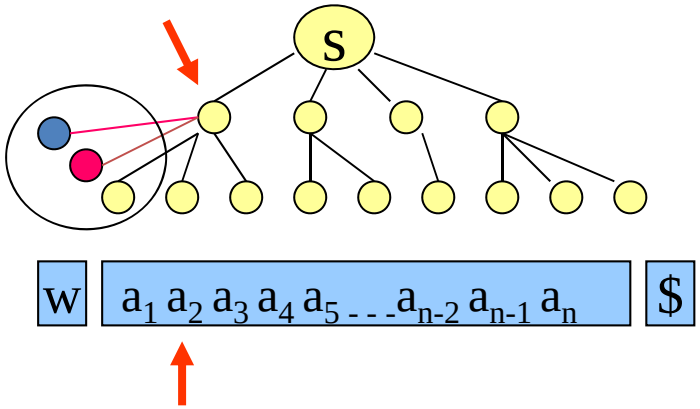
- **Repetition**

  (loop)



Compiler Construction

- a program is a **sequence**
- **G = (S, P, NT, T)**
- parse **tree**



Modelling & general computer science

- In reality most modelling is done using **databases**
- E/R model
- Entity / relationship
  - + attributes
- Each entity is **unique**
- What does this imply?
- **Relational databases – theory**
  - Entity ➔ **tuple**
  - Collection of entities ➔ **relation**
  - Relationship ➔ **relation**

attr1	attr2	attr3	…	attrn
key 1				
key 2				
…	…	…	…	…
key n				

a **set** of attributes
a **set** of entities
a **set** of relationships

Modelling & general computer science

- In reality most modelling is done using **databases**
- Sets can only be implemented as **sequences**
- Relationships
  - 1-1, 1-n, n-m

attr1	attr2	attr3	…	attrn
key 1				
key 2				
…	…	…	…	…
key n				

- **Relational databases – in practice**
  - Entity ➔ **record** — a **seq** of attributes
  - Collection of entities ➔ **table** — a **seq** of entities
  - Relationship ➔ **table** — a **seq** of relationships

Modelling & general computer science

- **Example: students, teachers and courses**
- student (name, **spno**, email, address, gender, …)
- teacher (name, **tpno**, email, address, gender, …)
- course (name, **code**, cno, syllabus, …)
  - **spno, tpno** & **code** are **unique** identifiers – **primary key**
- Relationships: **1-1**, **1-n** (one to many), **n-m** (many to many)
  - 1-1 one teacher teaches a course
  - 1-n one course is taught by several teachers
  - n-m each course has several students
each student takes several courses

Modelling & general computer science

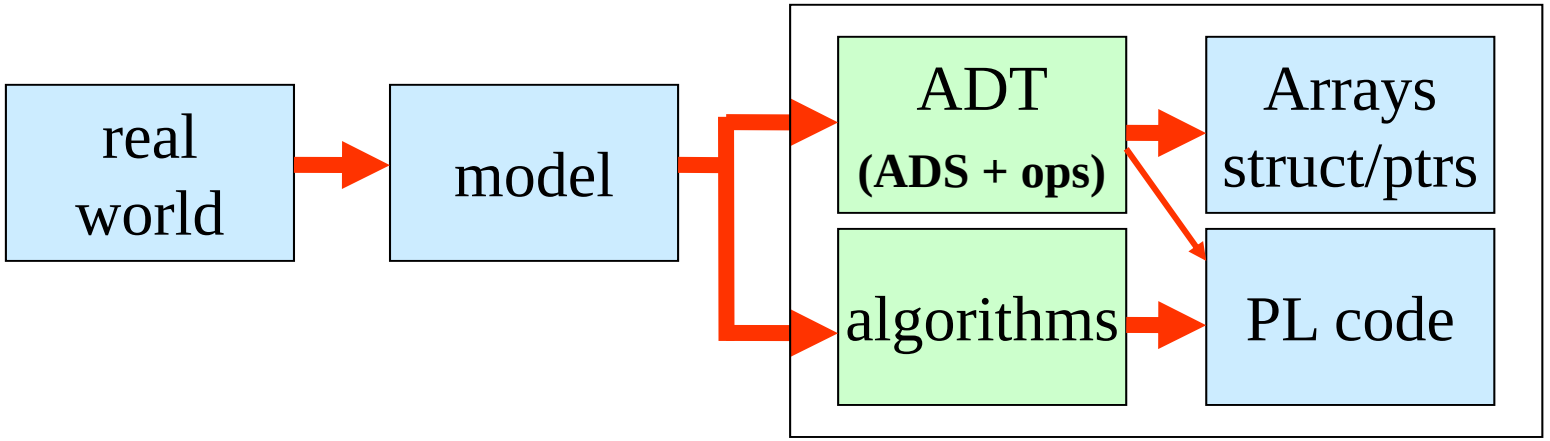
- **Example: students, teachers and courses**
- **1-1** course (name, **code**, cno, **tpno**, syllabus, …)
  - **tpno** id the teacher's personal number - **foreign key**
- **1-n** **n-m**

course	teacher
code 1	tpno 1
code 1	tpno 2
…	…
code 1	tpno n

student	course
spno 1	code 1
spno 1	code 2
…	…
spno n	code 1

The Red Thread

- **Modelling ➔ abstraction ➔ ADTs ➔ ADSs + ops**
- **Model World: entities, collections, relations + operations**
- **ADS is the Set, Sequence, Tree, Graph**
- **programming language DSs are arrays /structures + pointers**
- **algorithms are sequences of ops on ADTs**



Levels of Abstraction

Collection
Implementation