Chapter 1

Introducing the Universal Machine

1.1 The Very Idea of the Universal Machine

A **universal machine** is a general-purpose symbol-manipulating machine, capable of any task that one can represent as a **program**.

A **program** is an organized sequence of logical steps.

Any actual computer approximates an ideal universal machine.

1.1.1 Universal versus Special-Purpose Machines

What are the differences and commonalities of the following "machines"?

- car, microwave oven, TV set, CD player, elevator, piano
- pocket calculator, game boy, computer

1.1.2 What is a "Computer"?

to compute: (Latin *computare*; as COM-, *putare* ´reckon´)

Computer Science = Science of how to **Calculate** ???

Computer:

- One who computes, a calculator, reckoner, specifically a person employed to make calculations in an observatory, in surveying, etc." (*Oxford English Dictionary, 1926).*
- An apparatus built to perform routine calculations ... (*Grolier Multimedia Encyclopedia, 1993).*
- A usually electronic device for **storing** and **processing data** (usually in binary form), according to **instructions** given to it in a **variable program**" (*The Concise Oxford Dictionary, 9th Edition).*

Computers are machines for the **processing** of **information:**

- storage
- retrieval
- analyzation
- generation
- transformation
- ...

Computer Science

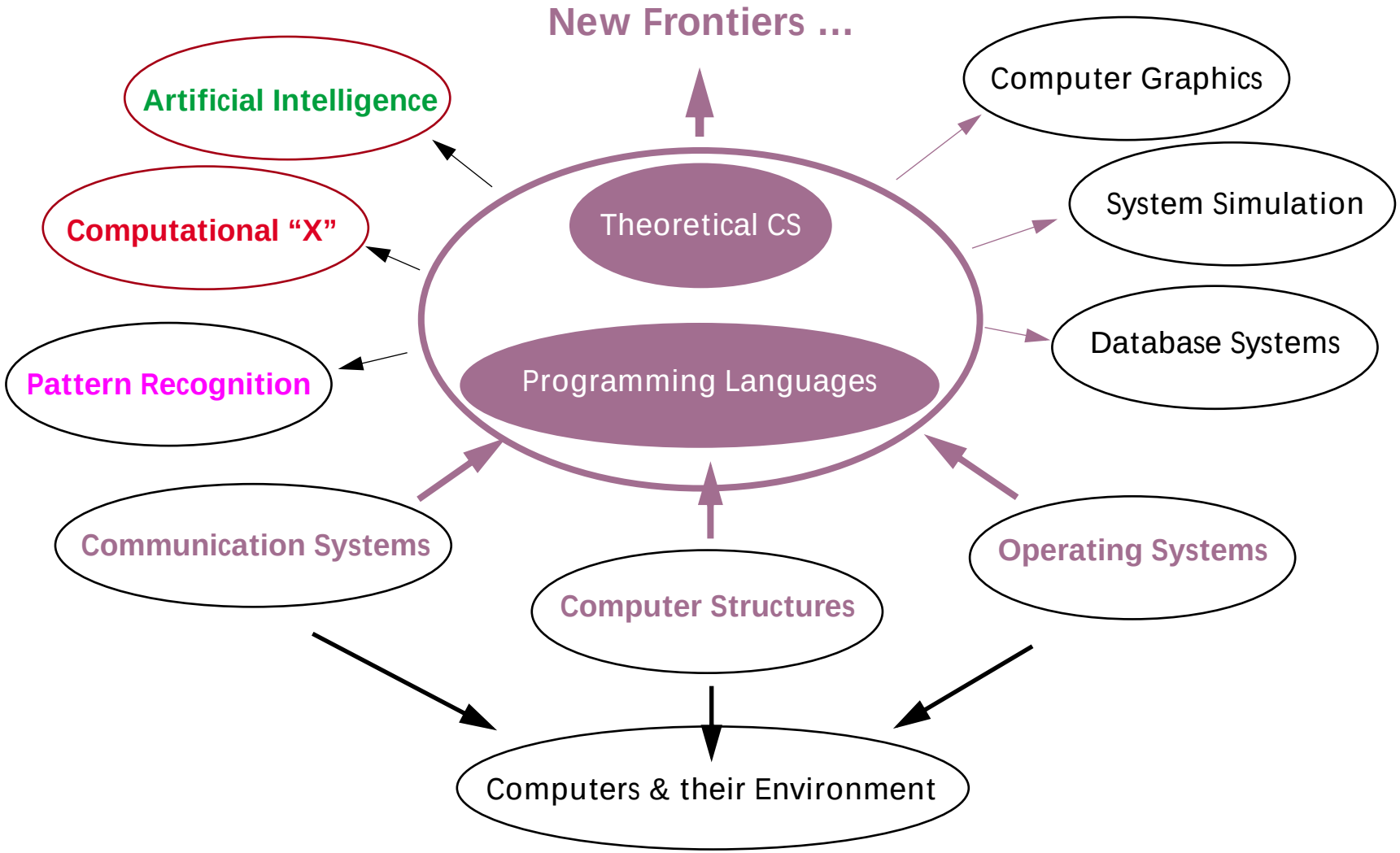
=

Science of **Processing Information** by Computers

=

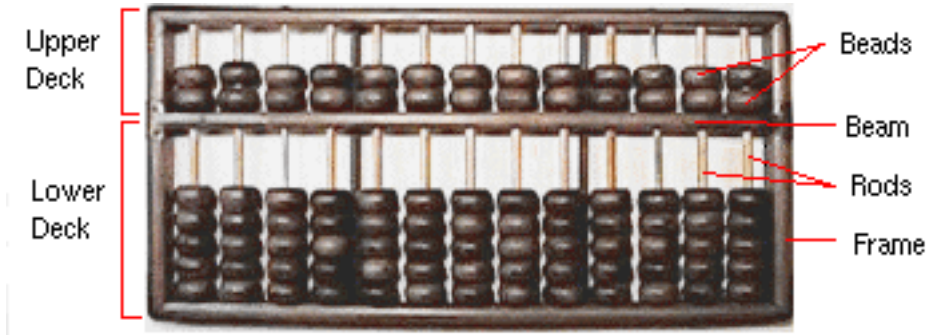
Informatics

1.1.3 Areas of Computer Science



1.1.4 Early Devices to Facilitate Numerical Operations

Abacus (about 200 B.C.)

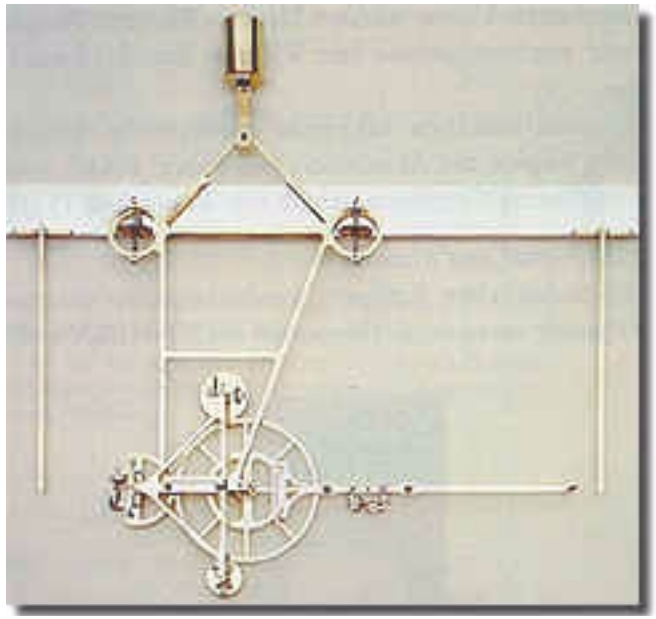


The abacus is a mechanical aid used for counting.

Addition, subtraction, division and multiplication can be performed on a standard abacus. The frame of the abacus has a series of vertical *rods* on which a number of wooden *beads* are allowed to slide freely.

A horizontal beam separates the frame into two sections, known as the *upper deck* and the *lower deck*.

Mathematical Instruments for Analog Computation

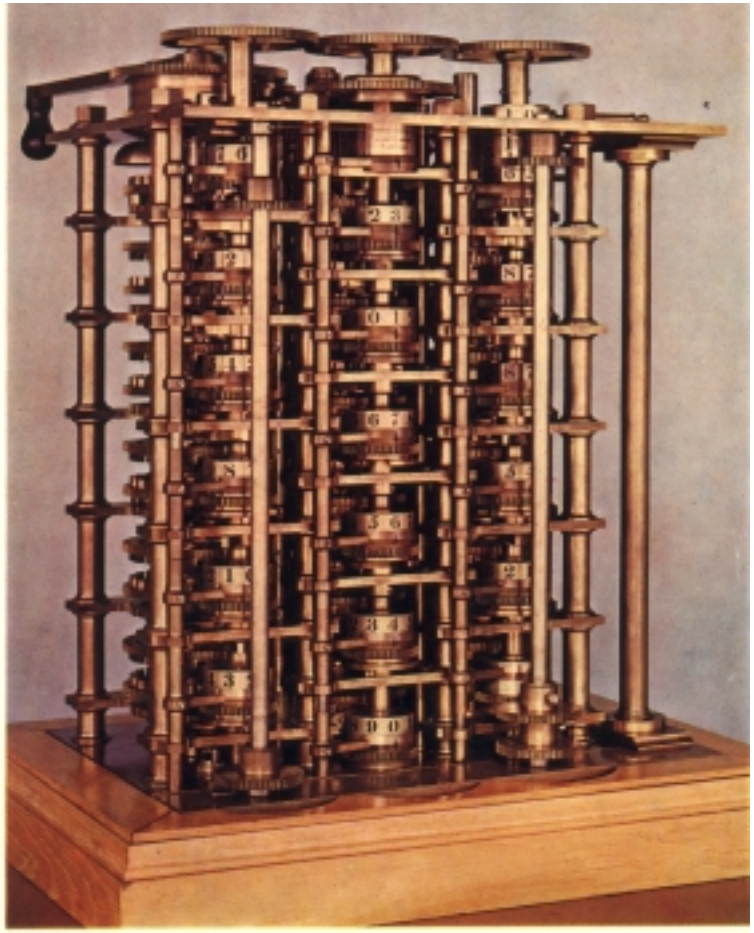


Left: The image shows a **calculator** created by Braun & Vayringe in 1735.

Right: **Planimeters** (1906) are very precise instruments for quick and accurate determination of areas on blueprints, maps, photographs, drawings, etc. By following the outline of the area with the tracer the revolution of the measuring wheel in either direction is sensed.

1.1.5 Babbage's Engines

Charles Babbage´s Difference Engine (1833)

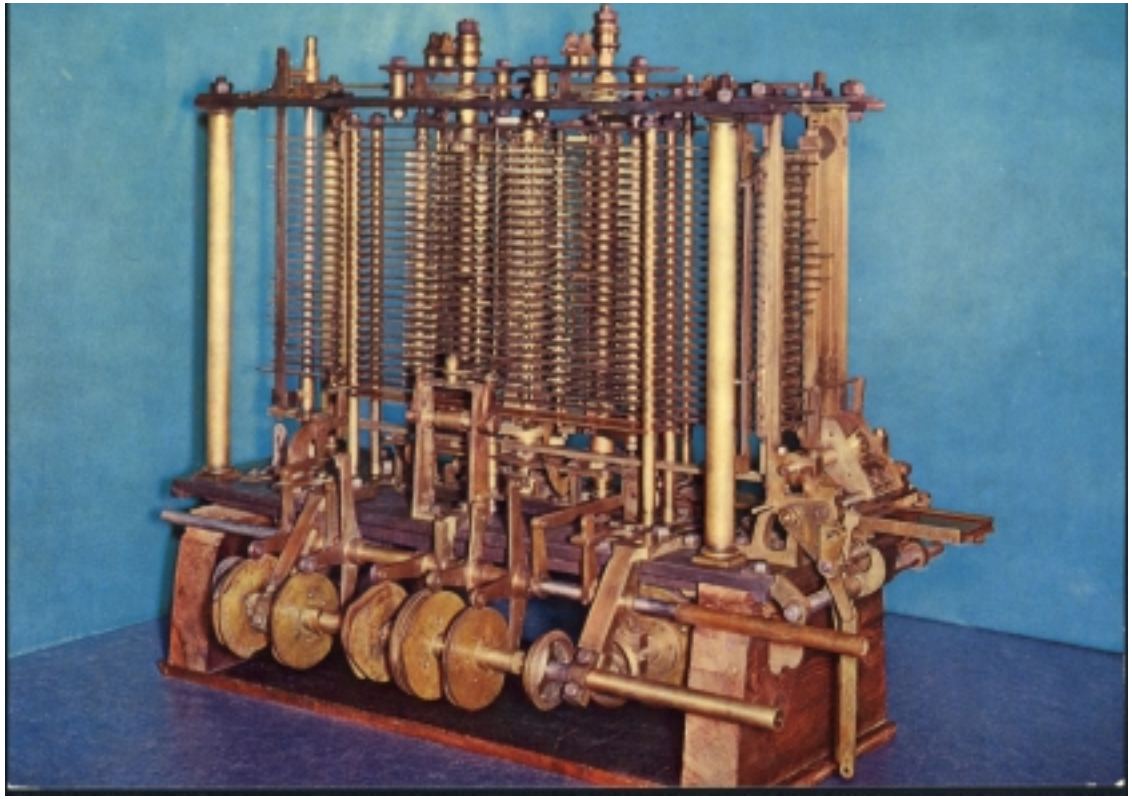


Charles Babbage (1791-1871) designed this engine in an attempt to mechanize the **computation of mathematical tables**.

He used the principle of the **method of differences**, i.e., adding increments (= differences) to intermediate results.

⇒ special purpose calculator

Charles Babbage´s Analytical Engine (1871)



Babbage abandoned the Difference Engine, to pursue a better idea: a **general purpose computer** for automatically solving mathematical equations.

The Analytical Engine would be able to

- carry out **any mathematical operation**
- by use of **stored programs**.

One could modify its behaviour by altering two boxes of punched cards:

- one box of variables, or **data**, and
- one box of **instruction codes**.

Add *Subtract* *Multiply* *Divide*

Thus a box of punched cards might be a program (a sequence of instructions), or it might be data (for example, a table that the engine had already computed and punched).

Babbage also noticed the importance of maintaining machine-readable libraries of programs and data (⇒ reusable software).

1.1.6 Babbage's Idea and Modern Computers

Babbage's engine separates a "store" from a "mill".

These correspond to the two most important components of a modern computer:

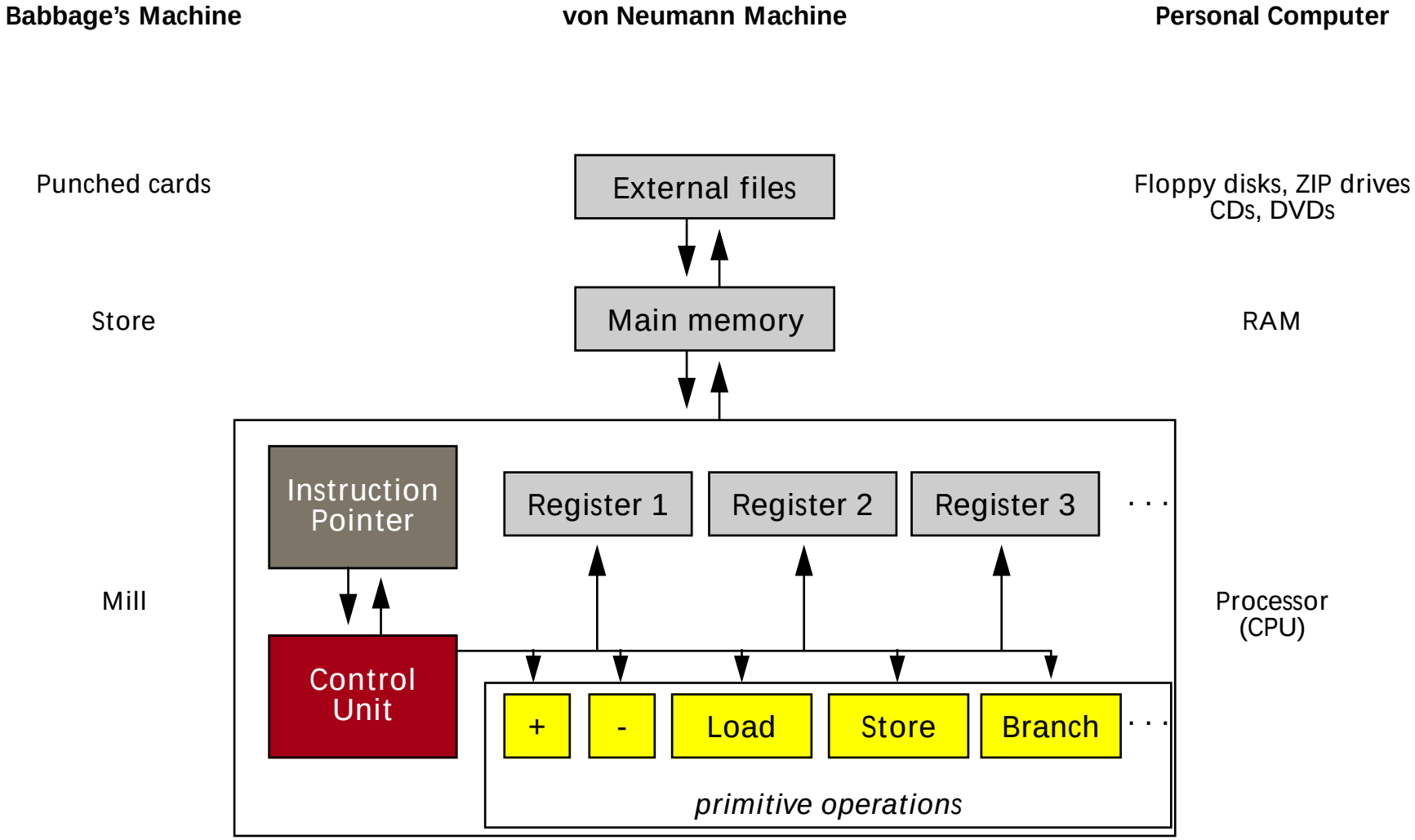
- the **main memory** and
- the **central processing unit** (CPU).

A modern **stored-program machine** can execute any sequence of instructions stored in its memory.

Both instructions and data are read into main memory.

⇒ By reading externally originated programs (external files) stored within its memory, the **actual machine** becomes whatever **virtual machine** the programmer wishes.

Structure of a stored-program machine



How does a stored-program machine work?

Like Babbage's machine, a modern machine can perform a small number of **primitive operations**.

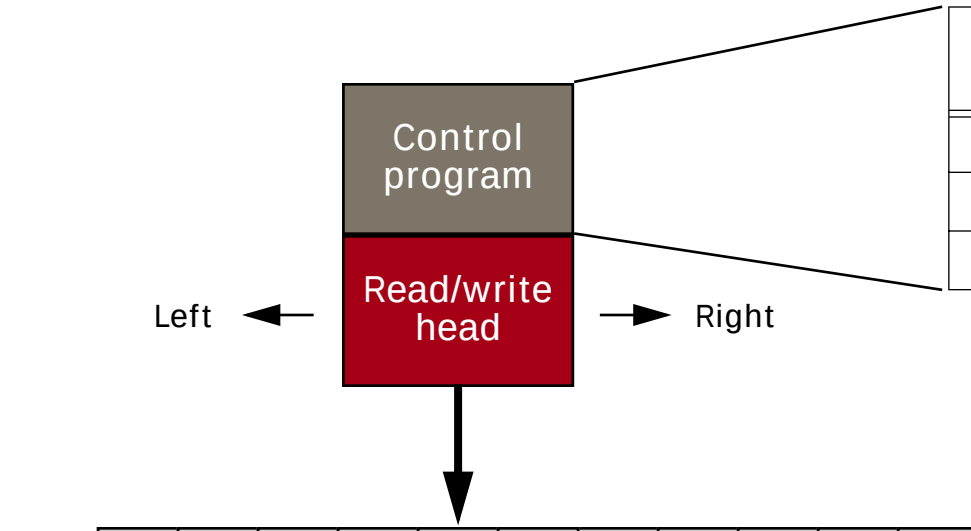
The **instruction pointer** tells where to find the next instruction in main memory.

The **control unit**

- fetches an instruction from memory,
- decodes it and
- executes the corresponding primitive operation.

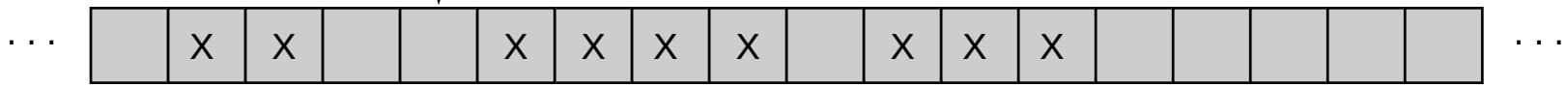
A **branch** instruction changes the value of the instruction pointer, so that it would jump to another part of the program.

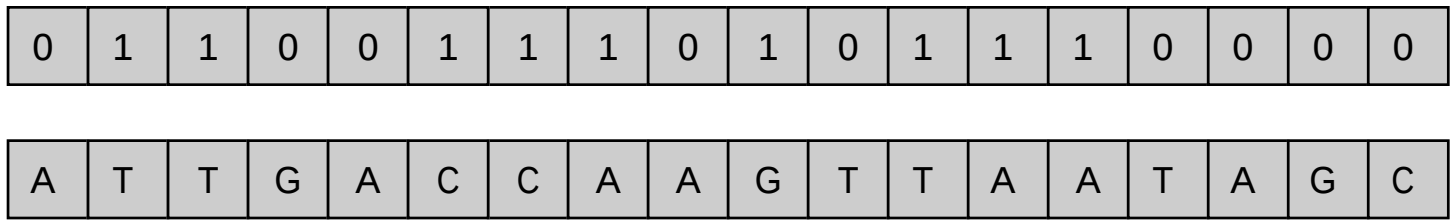
1.2 Turing's Symbol Manipulator



State	Read	Write	Move	Next state
1	B	B	Right	2
2	X	X	Right	2
2	B	B	STOP	

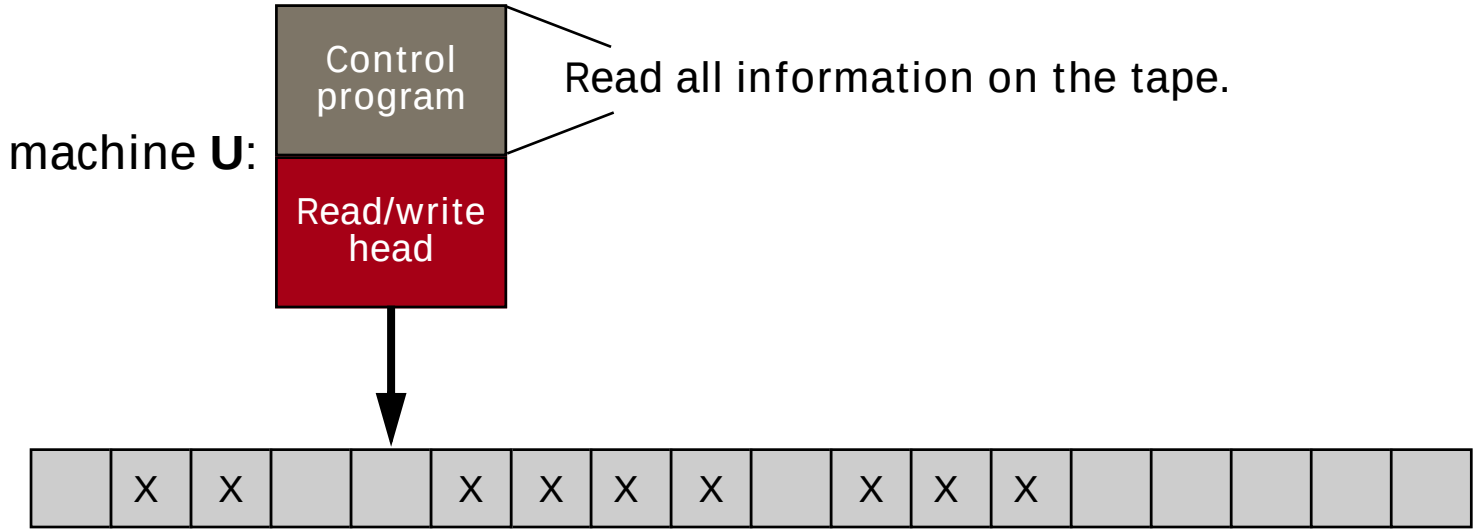
Moving right to the next blank





Alan Turing's universal machine, **U**, can **imitate any other Turing machine**, **M**.

Just encode **M**'s rules onto **U**'s tape. Then **U** reads **M**'s rules as a **program**.



Control program for machine **M** on the tape

Church-Turing Thesis: anything computable can be computed by
some Turing machine,
some register machine, some Pascal, C, C++, Java, etc. program.

1.3 References

- G. Blank and R. Barnes, *The Universal Machine*, Boston, MA: WCB/ McGraw-Hill, 1998. Chapters 1.1 and 1.2.
- M. R. Williams, *A History of Computing Technology*, Los Alamitos, CA: IEEE Compute Society Press, 1997. Chapter 4.