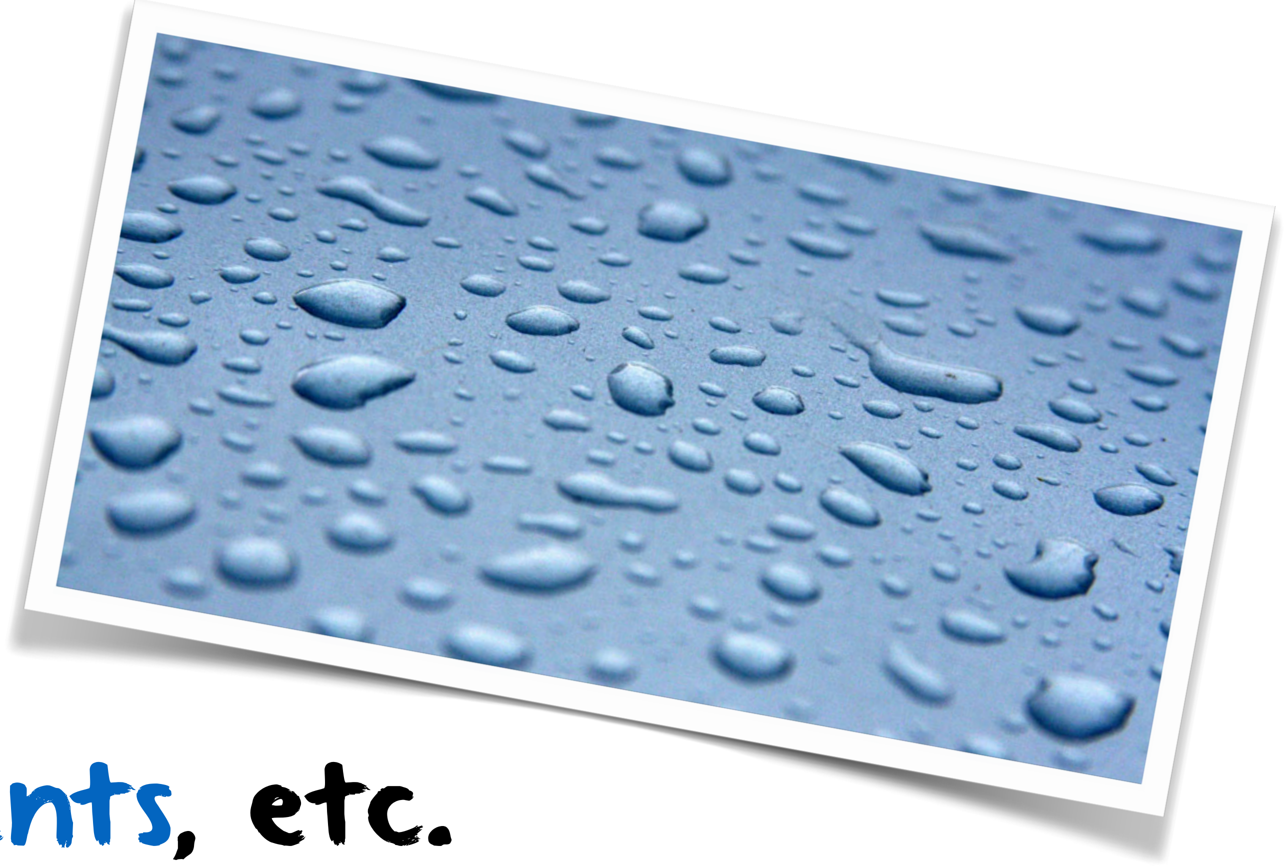
discrete math basics

learning objectives

algorithms
your software
system software
hardware

- learn the basics of discrete mathematics
- learn the basics of boolean algebra and logic
- learn the basics of set theory and first-order logic

what is discrete mathematics?



intuitively, discrete mathematics is the study of mathematical objects that are fundamentally discrete rather than continuous

such discrete objects can often be enumerated by integers, such as graphs, sets, state machines, logical statements, etc.

formally, discrete mathematics can be characterized as the branch of mathematics dealing with countable sets, e.g., finite sets or sets with the same cardinality as $\mathbb{N}$

at the university level, discrete mathematics appeared in the 1980s, often to support computer science curricula that were emerging

topics in discrete mathematics include logic, set theory, combinatorics, graph theory, topology, game theory, theoretical computer science, information theory, etc.

what is discrete mathematics?

continuous mathematics is the study of mathematical structures that are fundamentally continuous

limits, differentiation, integration, infinite, analytic functions

$\mathbb{R}$

discrete mathematics is the study of mathematical structures that are fundamentally discrete

graphs, sets, state machines, logical statements

$\mathbb{N}$

boolean algebra & logic

logic is the intellectual tool for reasoning about the truth and falsity of statements

logic & algorithms

most algorithms rely on some boolean variables, which can take values $\in$ {`true`,`false`}

in some low-level programming languages, integer numbers are used for the same purpose, e.g.. with:

```
p = false    ⇔  p = 0
q = true     ⇔  q = 1 (sometimes  q = true   ⇔   q ≠ 0 )
```

when combined with operators $\wedge$, $\vee$ and $\neg$, boolean variables constitute an algebra used in conditional branching

where: $\neg \Leftrightarrow$ not
$\vee \Leftrightarrow$ or
$\wedge \Leftrightarrow$ and

boolean algebra

assume that p , q and r are boolean variables (or statements) and that $T = true$, $F = false$, we have:

p	$\neg p$
F	T
T	F

p	q	$p \wedge q$
F	F	F
F	T	F
T	F	F
T	T	T

p	q	$p \vee q$
F	F	F
F	T	T
T	F	T
T	T	T

$\neg \Leftrightarrow$ **not**
$\vee \Leftrightarrow$ **or**
$\wedge \Leftrightarrow$ **and**

python	java

`a = False`
`b = True`
 	
`c = a and b`
`c = a or b`
`c = not a`
	`boolean a = false;`
	`boolean b = true;`
	
	`boolean c = a && b;`
	`c = a || b;`
	`c = !a;`

De Morgan's law

Associative Rules:	$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$	$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
Distributive Rules:	$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$
Idempotent Rules:	$p \wedge p \Leftrightarrow p$	$p \vee p \Leftrightarrow p$
Double Negation:	$\neg\neg p \Leftrightarrow p$	
DeMorgan's Rules:	$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$	$\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$
Commutative Rules:	$p \wedge q \Leftrightarrow q \wedge p$	$p \vee q \Leftrightarrow q \vee p$
Absorption Rules:	$p \vee (p \wedge q) \Leftrightarrow p$	$p \wedge (p \vee q) \Leftrightarrow p$
Bound Rules:	$p \wedge F \Leftrightarrow F \quad p \wedge T \Leftrightarrow p$	$p \vee T \Leftrightarrow T \quad p \vee F \Leftrightarrow p$
Negation Rules:	$p \wedge (\neg p) \Leftrightarrow F$	$p \vee (\neg p) \Leftrightarrow T$

propositional logic

propositional calculus is a branch of logic dealing with propositions (true or false) and relations between propositions, including the construction of arguments based on them

Premise 1: if $\overbrace{\text{it is raining}}^{P}$ then $\overbrace{\text{it is cloudy}}^{Q}$.
Premise 2: it is raining.
Conclusion: it is cloudy.

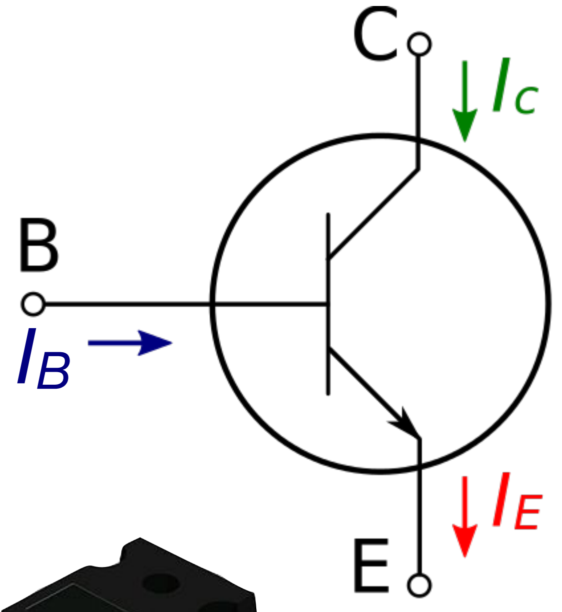
$$\left.\begin{array}{ll}\text{Premise 1:} & P \to Q \\ \text{Premise 2:} & P \\ \text{Conclusion:} & Q\end{array}\right\} \quad P \to Q, P \vdash Q$$

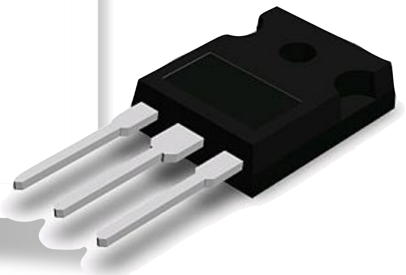
Name	Sequent	Description
Modus Ponens	$((p \to q) \land p) \vdash q$	If p then q; p; therefore q
Modus Tollens	$((p \to q) \land \neg q) \vdash \neg p$	If p then q; not q; therefore not p
Hypothetical Syllogism	$((p \to q) \land (q \to r)) \vdash (p \to r)$	If p then q; if q then r; therefore, if p then r
Tertium non datur (Law of Excluded Middle)	$\vdash (p \lor \neg p)$	p or not p is true
Law of Non-Contradiction	$\vdash \neg(p \land \neg p)$	p and not p is false, is a true statement

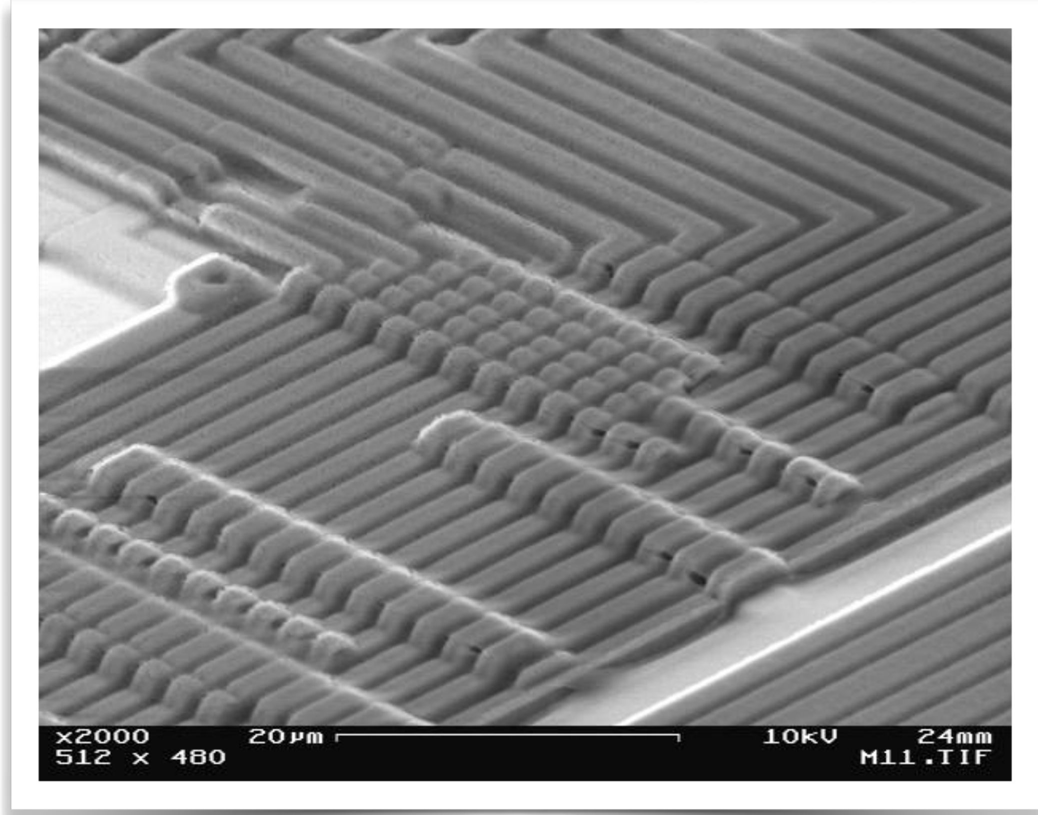
transistors & boolean algebra

the example of the "and" and "or" gates

a transistor is a device that can amplify or switch an electrical current, using three layers of a semiconductor material





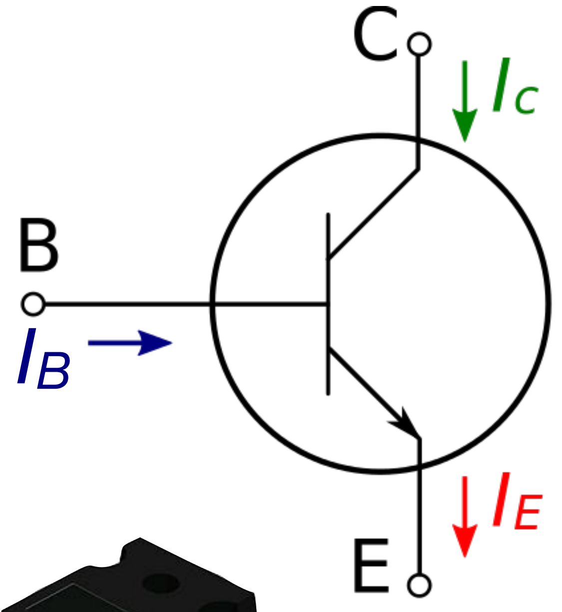


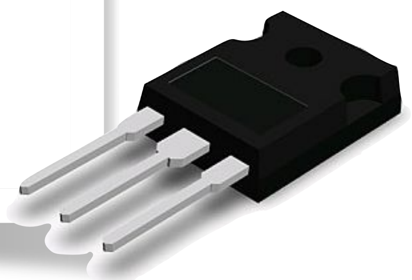
10 µm	1971
6 µm	1974
3 µm	1977
1.5 µm	1981
1 µm	1984
800 nm	1987
600 nm	1990
350 nm	1993
250 nm	1996
180 nm	1999
130 nm	2001
90 nm	2003
65 nm	2005
45 nm	2007
32 nm	2009
22 nm	2012
14 nm	2014
10 nm	2016
7 nm	2018
5 nm	2019
3 nm	2021

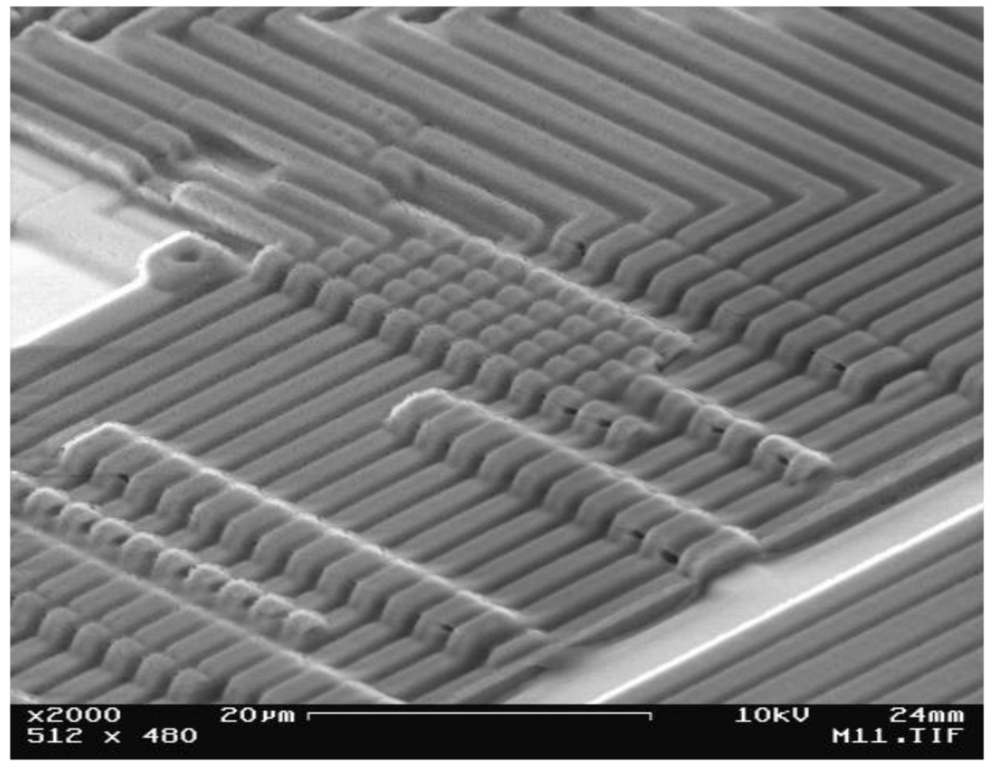
transistors & boolean algebra

the example of the "and" and "or" gates

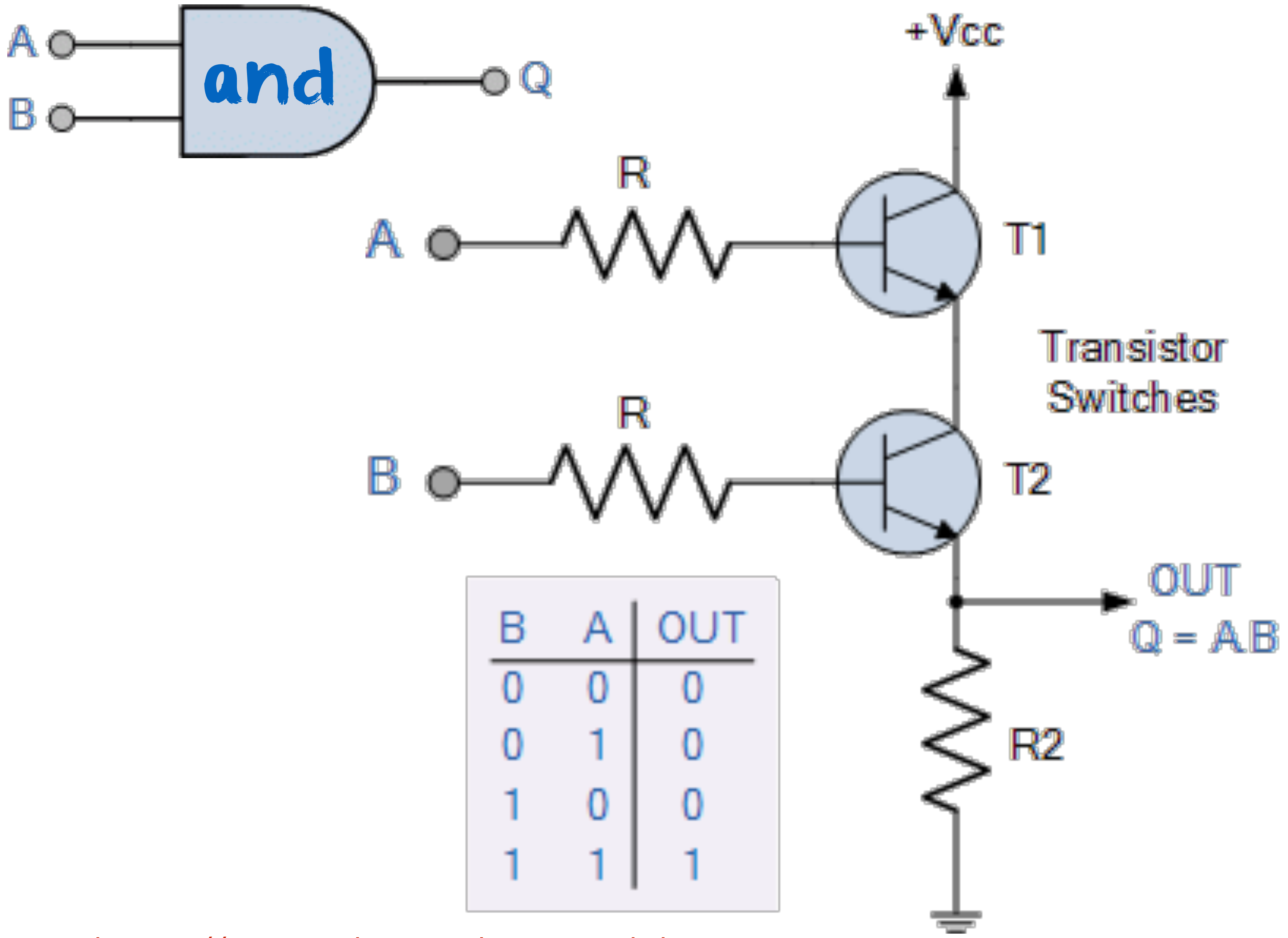
a transistor is a device that can amplify or switch an electrical current, using three layers of a semiconductor material

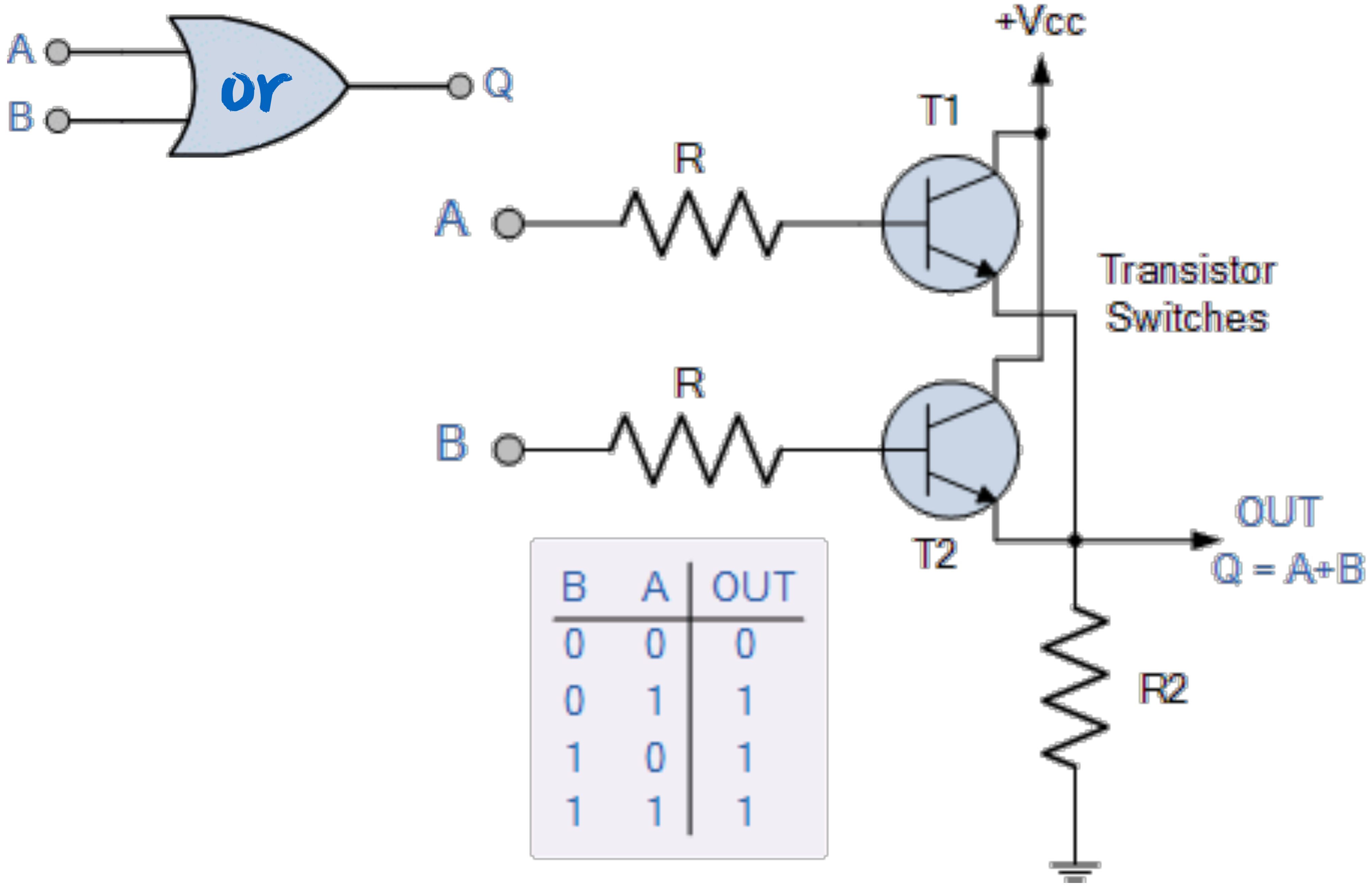






10 µm	1971
6 µm	1974
3 µm	1977
1.5 µm	1981
1 µm	1984
800 nm	1987
600 nm	1990
350 nm	1993
250 nm	1996
180 nm	1999
130 nm	2001
90 nm	2003
65 nm	2005
45 nm	2007
32 nm	2009
22 nm	2012
14 nm	2014
10 nm	2016
7 nm	2018
5 nm	2019
3 nm	2021





source: *https://www.electronics-tutorials.ws*

what is set theory ?

set theory is a branch of mathematical logic that studies sets, which can informally be seen as collections of objects

although any type of object can be collected into a set, set theory is applied most often to objects that are relevant to mathematics

the study of set theory was initiated by Georg Cantor in the 1870s

the language of set theory can be used to define nearly all mathematical objects

$$\mathbb{N}$$

$$\begin{aligned} 0 &= \{\} &&= \emptyset, \\ 1 &= \{0\} &&= \{\emptyset\}, \\ 2 &= \{0,1\} &&= \{\emptyset,\{\emptyset\}\}, \\ 3 &= \{0,1,2\} &&= \{\emptyset,\{\emptyset\},\{\emptyset,\{\emptyset\}\}\}. \end{aligned}$$

for this reason, it is commonly used as a foundational system for mathematics, particularly via the Zermelo-Fraenkel set theory with the axiom of choice

set theory notation

set definition and set membership

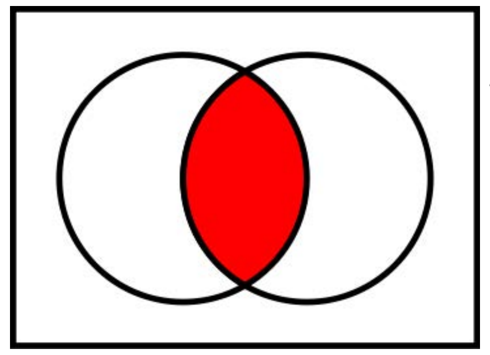
$\{7, 3, 15, 31\}$ $\quad \{a, c, b\} = \{a, b, c\}$ $\quad \{1, 2, 3, \ldots, 100\}$ $\quad \{1, 2, 3, \ldots\}$

$15 \in \{7, 3, 15, 31\}$ $\quad 6 \notin \{7, 3, 15, 31\}$ $\quad c \in \{a, c, b\}$ $\quad 64 \in \{1, 2, 3, \ldots, 100\}$

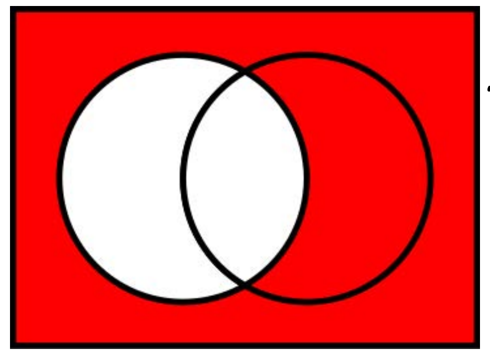
$\{2n \mid n \in \mathbb{N}\}$ $\quad \{x \in \mathbb{R} \mid x > 0\}$ $\quad \{x \in \mathbb{R} \mid |x| = 1\}$ $\quad \{(x, y) \in \mathbb{R} \times \mathbb{R} \mid 0 < y < f(x)\}$

$64 \in \{2n \mid n \in \mathbb{N}\}$ $\quad -1 \in \{x \in \mathbb{R} \mid |x| = 1\}$

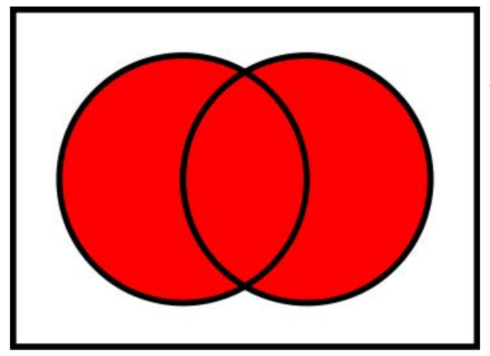
set operations



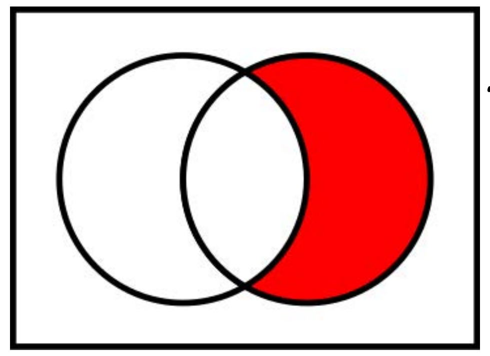
$A \cap B$



$A^c = U \setminus A$



$A \cup B$



$A^c \cap B = B \setminus A$

some laws

$\varnothing^C = U$ $\quad A \cap \varnothing = \varnothing$ $\quad A \cup A = A$ $\quad A \cup (A \cap B) = A$

$U^C = \varnothing$ $\quad A \cup U = U$ $\quad A \cap A = A$ $\quad A \cap (A \cup B) = A$

subsets

$A \subseteq B \Leftrightarrow A \cap B = A$ $\qquad A \subseteq B \Leftrightarrow A \cup B = B$

what is first-order logic ?

first-order logic is a formal system used in mathematics that relies on **quantified variables** over non-logical objects

quantified variables

there exists x $\longleftarrow\!\!\!\mid$ $\exists x$ $\qquad\qquad$ $\forall y$ $\longmapsto$ for all y

quantifiers

first-order logic can be seen as an extension of **propositional logic**

$$\exists x \, P \qquad\qquad \forall y \, Q$$

propositions

Russell's paradox

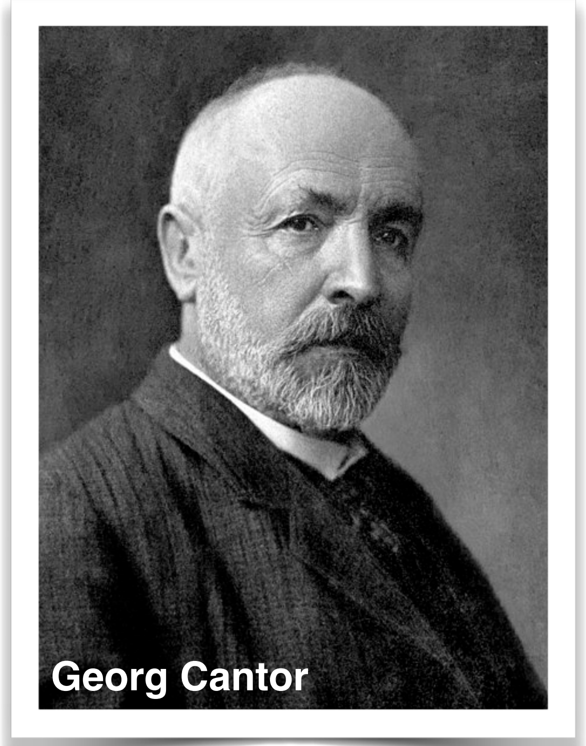
this paradox shows that the naive set theory created by Cantor is contradictory

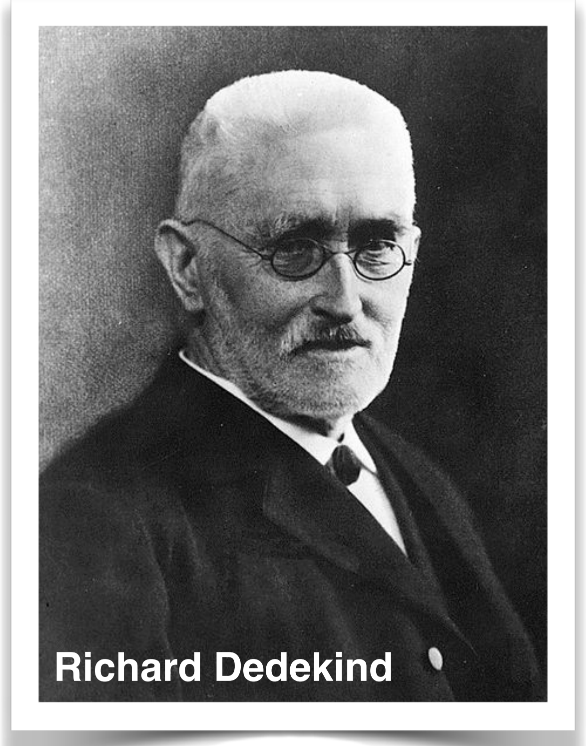
according to this theory, any definable collection is a set

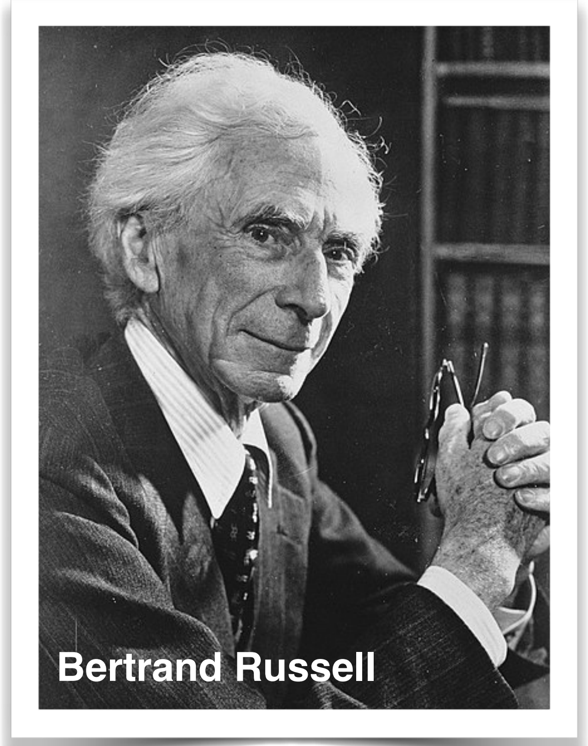
$$\text{Let } R = \{x \mid x \notin x\}, \text{ then } R \in R \iff R \notin R$$

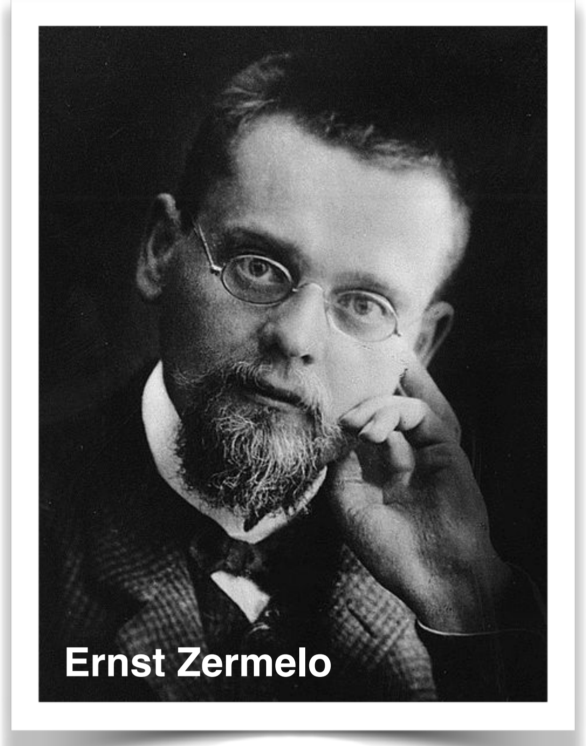
to avoid this paradox, Russell proposed to alter the logical language itself, whereas Zermelo simply altered the axioms of set theory

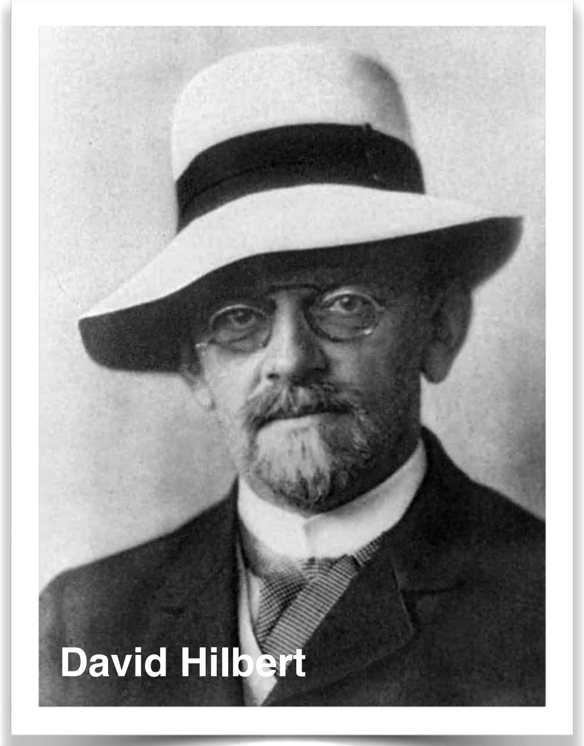
the now-standard Zermelo-Fraenkel set theory turned out to be first-order logic

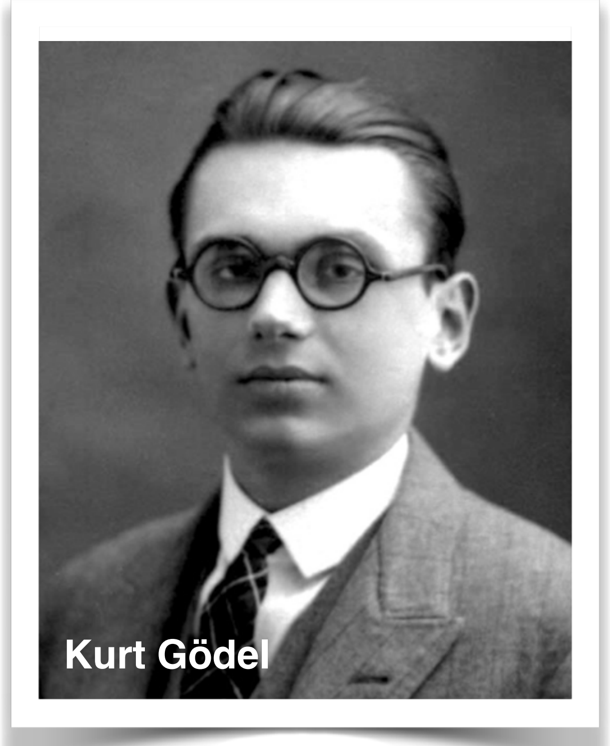

Georg Cantor

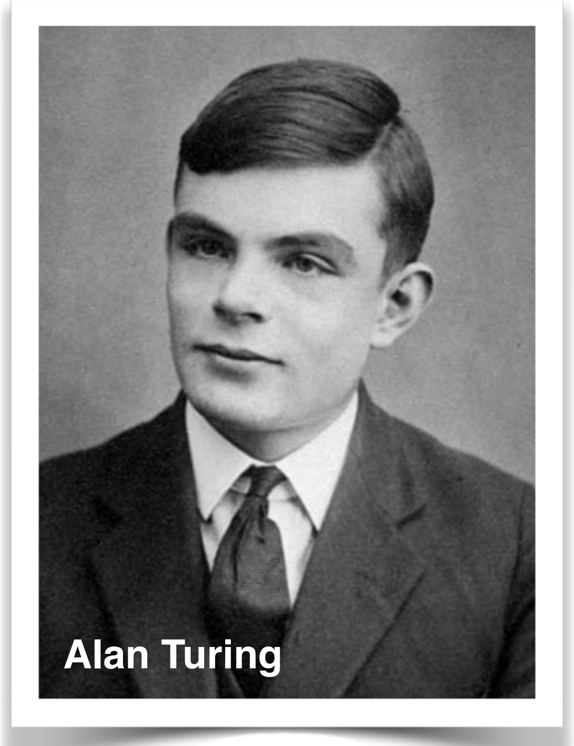

Richard Dedekind


Bertrand Russell


Ernst Zermelo


David Hilbert


Kurt Gödel


Alan Turing