Compilers and Formal Languages (10)

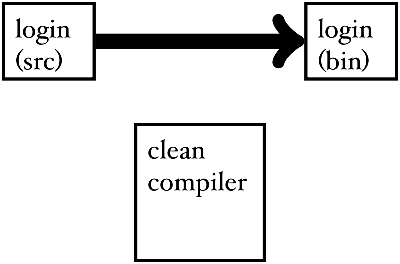
Email:	christian.urban at kcl.ac.uk
Office:	N7.07 (North Wing, Bush House)
Slides:	KEATS (also home work is there)

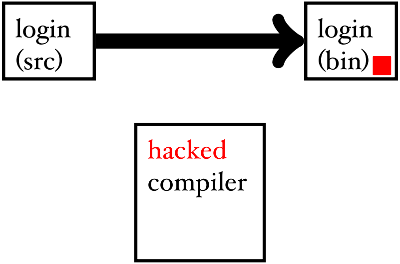
Using a compiler,
how can you mount the
perfect attack against a system?

What is a perfect attack?

1. you can potentially completely take over a target system
2. your attack is (nearly) undetectable
3. the victim has (almost) no chance to recover

clean
compiler


login
(src)
login
(bin)
clean
compiler


login
(src)
login
(bin)
hacked
compiler

my compiler (src)

V0.01

Scala

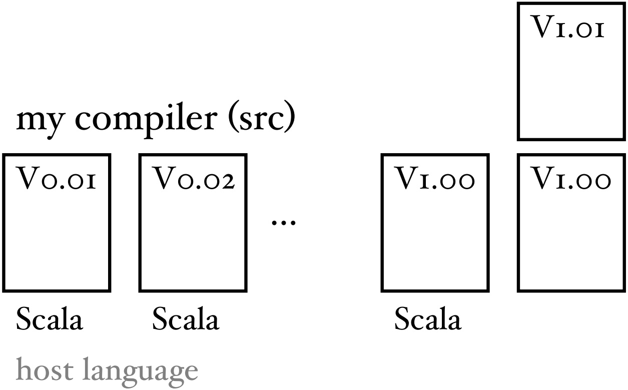
host language

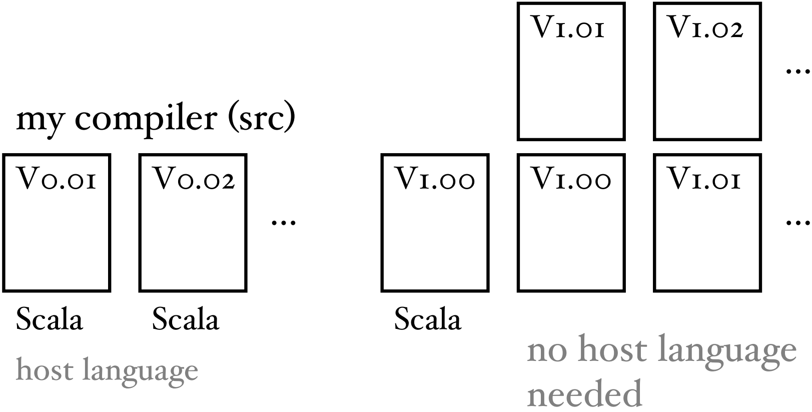
my compiler (src)

V0.01 V0.02 ... V1.00

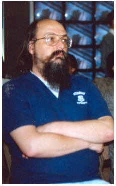
Scala Scala Scala

host language


my compiler (src)
V0.01
V0.02
...
V1.00
V1.00
V1.01
Scala
Scala
Scala
host language


my compiler (src)
V0.01
V0.02
...
Scala
Scala
host language
V1.01
V1.02
...
V1.00
V1.00
V1.01
...
Scala
no host language needed

Hacking Compilers

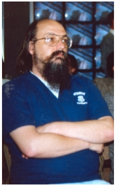


Ken Thompson
Turing Award, 1983

Ken Thompson showed how to hide a Trojan Horse in a compiler without leaving any traces in the source code.

No amount of source level verification will protect you from such Thompson-hacks.

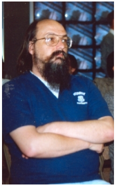
Hacking Compilers



Ken Thompson
Turing Award, 198

1) *Assume you ship the compiler as binary and also with sources.*
2) *Make the compiler aware when it compiles itself.*
3) *Add the Trojan horse.*
4) *Compile.*
5) *Delete Trojan horse from the sources of the compiler.*
6) *Go on holiday for the rest of your life. ;o)*

Hacking Compilers



Ken Thompson
Turing Award, 1983

Ken Thompson showed how to hide a Trojan Horse in a compiler without leaving any traces in the source code.

No amount of source level verification will protect you from such Thompson-hacks.

Compilers & Boeings 777

First flight in 1994. They want to achieve triple redundancy in hardware faults.

They compile 1 Ada program to

- Intel 80486
- Motorola 68040 (old Macintosh's)
- AMD 29050 (RISC chips used often in laser printers)

using 3 independent compilers.

Compilers & Boeings 777

First flight in 1994. They want to achieve triple redundancy in hardware faults.

They compile 1 Ada program to

- Intel 80486
- Motorola 68040 (old Macintosh's)
- AMD 29050 (RISC chips used often in laser printers)

using 3 independent compilers.

Airbus uses C and static analysers. Recently started using CompCert.

How many strings are in $L(a^*)$?

How many strings are in $L(a^*)$?

$[]$	a	aa	aaa	$aaaa$	...
0	1	2	3	4	...

There are more problems, than there are programs.

There are more problems, than there are programs.

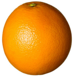
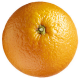
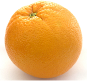
There must be a problem for which there is no program.

Subsets

If $A \subseteq B$ then A has fewer or equal elements than B

$A \subseteq B$ and $B \subseteq A$

then $A = B$

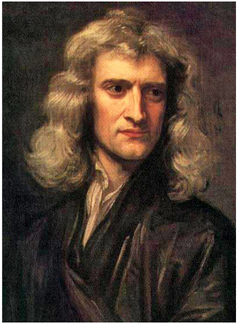
{ , , , ,  }

5 elements

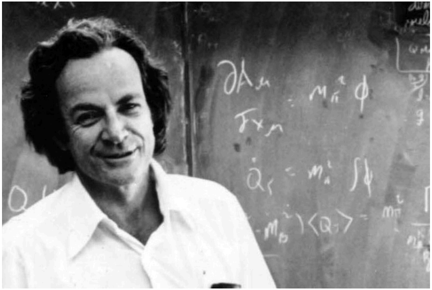
{ , ,  }

3 elements

Newton vs Feynman



classical physics



quantum physics

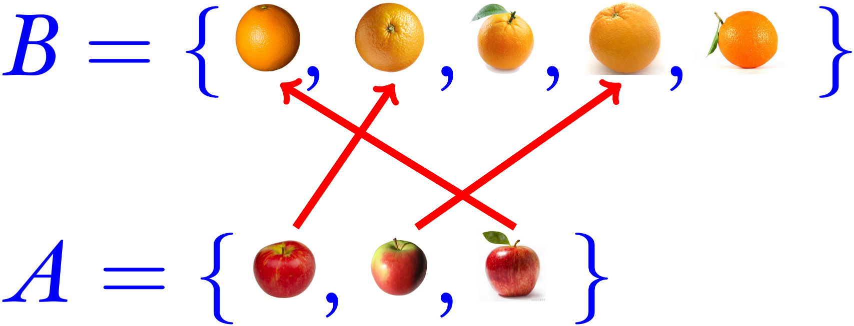
The Goal of the Talk

- show you that something very unintuitive happens with very large sets
- convince you that there are more **problems** than **programs**

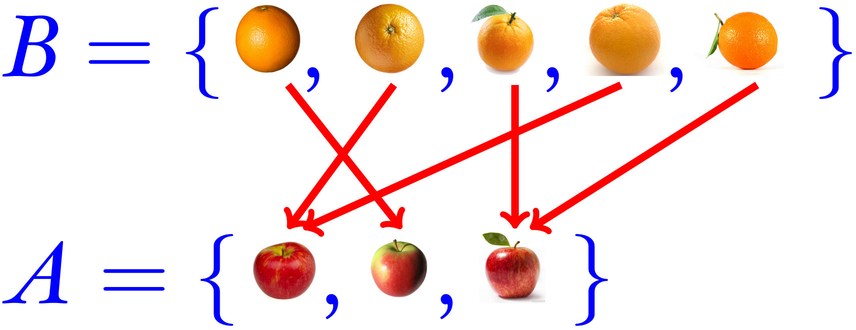
$B = \{$ , , , ,  $\}$

$A = \{$ , ,  $\}$

$$|A| = 5, |B| = 3$$



then $|A| \leq |B|$



for $=$ has to be a **one-to-one** mapping

Cardinality

$|A| \stackrel{\text{def}}{=}$ "how many elements"

$A \subseteq B \Rightarrow |A| \leq |B|$

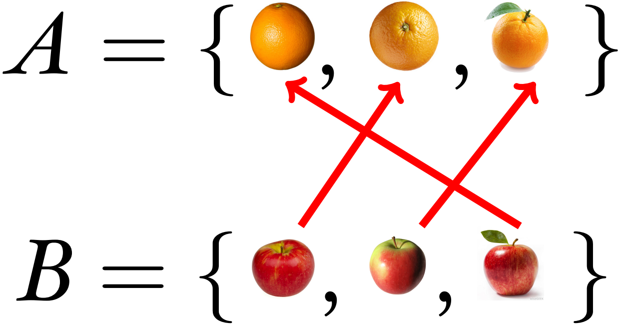
Cardinality

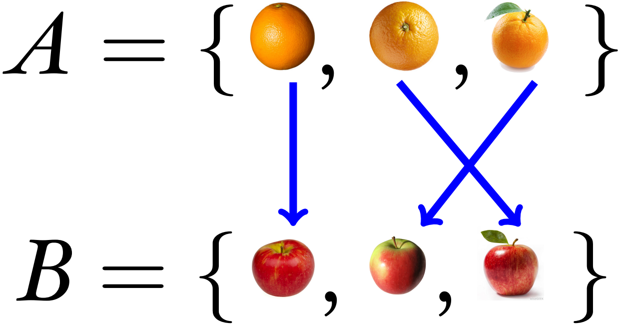
$|A| \stackrel{\text{def}}{=}$ "how many elements"

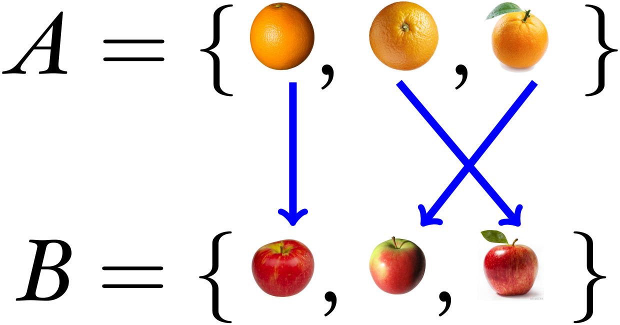
$A \subseteq B \Rightarrow |A| \leq |B|$

if there is an injective function $f : A \rightarrow B$ then $|A| \leq |B|$

$$\forall xy.\, f(x) = f(y) \Rightarrow x = y$$


A =
B =


A =
B =



then $|A| = |B|$

Natural Numbers

$$\mathbb{N} \stackrel{\text{def}}{=} \{0, 1, 2, 3, \ldots\ldots\}$$

Natural Numbers

$$\mathbb{N} \stackrel{\text{def}}{=} \{0, 1, 2, 3, \ldots\ldots\}$$

A is **countable** iff $|A| \leq |\mathbb{N}|$

First Question

$$|\mathbb{N} - \{0\}| \quad ? \quad |\mathbb{N}|$$

$\geq$ or $\leq$ or $=$?

First Question

$$|\mathbb{N} - \{0\}| \quad ? \quad |\mathbb{N}|$$

$\geq$ or $\leq$ or $=$?

$x \mapsto x + 1$,
$|\mathbb{N} - \{0\}| = |\mathbb{N}|$

$$|\mathbb{N} - \{0, 1\}| \quad ? \quad |\mathbb{N}|$$

$$|\mathbb{N} - \{0, 1\}| \quad ? \quad |\mathbb{N}|$$

$$|\mathbb{N} - \mathbb{O}| \quad ? \quad |\mathbb{N}|$$

$\mathbb{O} \stackrel{\text{def}}{=}$ odd numbers $\quad \{1, 3, 5.....\}$

$$|\mathbb{N} - \{0, 1\}| \quad ? \quad |\mathbb{N}|$$

$$|\mathbb{N} - \mathbb{O}| \quad ? \quad |\mathbb{N}|$$

$\mathbb{O} \stackrel{\text{def}}{=}$ odd numbers $\quad \{1, 3, 5 \ldots\ldots\}$

$\mathbb{E} \stackrel{\text{def}}{=}$ even numbers $\quad \{0, 2, 4 \ldots\ldots\}$

$$|\mathbb{N} \cup -\mathbb{N}| \quad ? \quad |\mathbb{N}|$$

$\mathbb{N} \stackrel{\text{def}}{=}$ positive numbers $\{0, 1, 2, 3, \ldots\ldots\}$

$-\mathbb{N} \stackrel{\text{def}}{=}$ negative numbers $\{0, -1, -2, -3, \ldots\ldots\}$

A is countable if there exists an injective $f : A \to \mathbb{N}$

A is uncountable if there does not exist an injective $f : A \to \mathbb{N}$

countable: $|A| \leq |\mathbb{N}|$
uncountable: $|A| > |\mathbb{N}|$

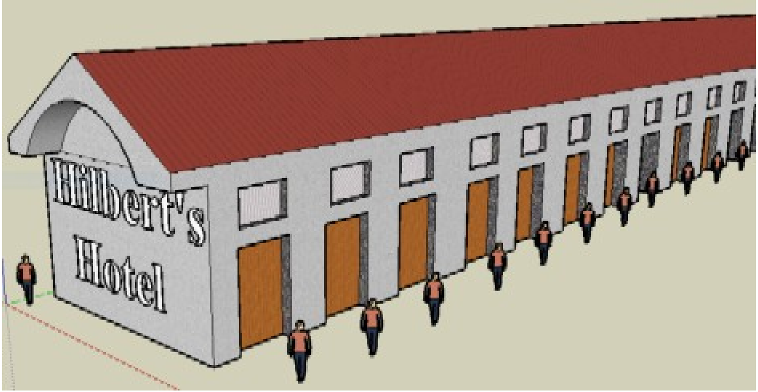
A is countable if there exists an injective $f : A \rightarrow \mathbb{N}$

A is uncountable if there does not exist an injective $f : A \rightarrow \mathbb{N}$

countable: $|A| \leq |\mathbb{N}|$
uncountable: $|A| > |\mathbb{N}|$

Does there exist such an A ?

Hilbert's Hotel



- ...has as many rooms as there are natural numbers

Real Numbers between 0 and 1

1	3	3	3	3	3	3	...
2	1	2	3	4	5	6	7
3	0	1	0	1	0	...	
4	7	8	5	3	9	...	

...

...

Real Numbers between 0 and 1

1	4	3	3	3	3	3	…
2	1	2	3	4	5	6	7
3	0	1	0	1	0	…	
4	7	8	5	3	9	…	

…

…

Real Numbers between 0 and 1

1	4	3	3	3	3	3	…	…
2	1	3	3	4	5	6	7	
3	0	1	0	1	0	…		
4	7	8	5	3	9	…		

…

Real Numbers between 0 and 1

1	4	3	3	3	3	3	...	...
2	1	3	3	4	5	6	7	
3	0	1	1	1	0	...		
4	7	8	5	3	9	...		

...

Real Numbers between 0 and 1

1	4	3	3	3	3	3	…	…
2	1	3	3	4	5	6	7	
3	0	1	1	1	0	…		
4	7	8	5	4	9	…		

…

Real Numbers between 0 and 1

1	4	3	3	3	3	3	...	...
2	1	3	3	4	5	6	7	
3	0	1	1	1	0	...		
4	7	8	5	4	9	...		

...

$$|\mathbb{N}| < |R|$$

The Set of Problems

$\aleph_0$

	0	1	2	3	4	5	...
1	0	1	0	1	0	1	...
2	0	0	0	1	1	0	0
3	0	0	0	0	0	...	
4	1	1	0	1	1	...	

...

...

The Set of Problems

$\aleph_0$

	0	1	2	3	4	5	$\cdots$	
1	0	1	0	1	0	1	$\cdots$	$\cdots$
2	0	0	0	1	1	0	0	
3	0	0	0	0	0	$\cdots$		
4	1	1	0	1	1	$\cdots$		

$\cdots$

$$|Progs| = |\mathbb{N}| < |Probs|$$

Halting Problem

Assume a program H that decides for all programs A and all input data D whether

- $H(A,D) \stackrel{\text{def}}{=} 1$ iff $A(D)$ terminates
- $H(A,D) \stackrel{\text{def}}{=} 0$ otherwise

Halting Problem (2)

Given such a program H define the following program C: for all programs A

- $C(A) \stackrel{\text{def}}{=} 0$ iff $H(A, A) = 0$
- $C(A) \stackrel{\text{def}}{=}$ loops otherwise

Contradiction

$H(C,C)$ is either 0 or 1.

- $H(C,C) = 1 \stackrel{\text{def}\,H}{\Rightarrow} C(C)\downarrow \stackrel{\text{def}\,C}{\Rightarrow} H(C,C) = 0$
- $H(C,C) = 0 \stackrel{\text{def}\,H}{\Rightarrow} C(C)$ loops $\stackrel{\text{def}\,C}{\Rightarrow}$
  $H(C,C) = 1$

Contradiction in both cases. So H cannot exist.

Take Home Points

- there are sets that are more infinite than others
- even with the most powerful computer we can imagine, there are problems that cannot be solved by any program
- in CS we actually hit quite often such problems (halting problem)