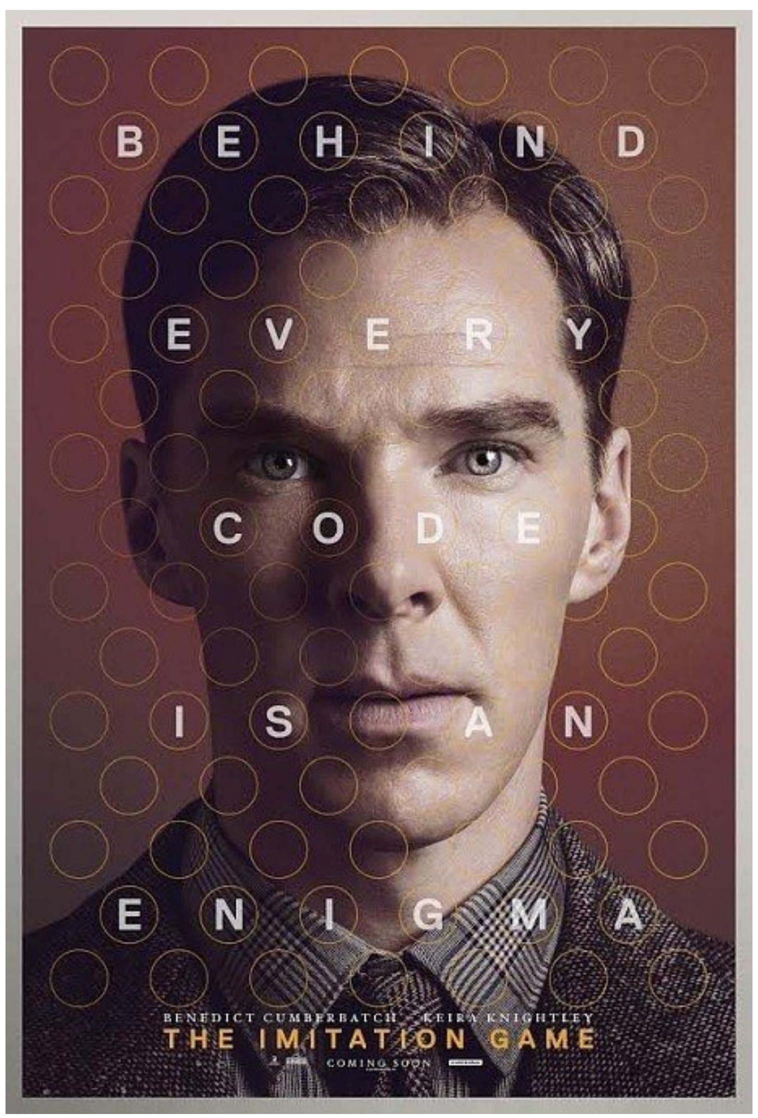
Automata & languages

A primer on the Theory of Computation



Laurent Vanbever

nsg.ethz.ch

ETH Zürich (D-ITET)

10 October 2024

Part 4 out of 5

Last week, we showed the

equivalence of DFA, NFA and REX

is equivalent to

DFA $\asymp$ NFA

$\asymp$

REX

We also started to look at
non-regular languages

Pumping lemma

If *A* is a regular language, then

there exist a number *p* s.t.

Any string in *A* whose length is at least *p*

can be divided into three pieces *xyz* s.t.

- $xy^iz \in A$, for each i≥0 and
- $|y| > 0$ and
- $|xy| \leq p$

To prove that a language A is not regular:

1. Assume that A is regular
2. Since A is regular, it must have a pumping length p
3. Find one string s in A whose length is at least p
4. For any split $s=xyz$,
   Show that you cannot satisfy all three conditions
5. Conclude that s cannot be pumped

To prove that a language *A* is not regular:

1 Assume that *A* is regular

2 Since *A* is regular, it must have a pumping length *p*

3 Find one string *s* in *A* whose length is at least *p*

4 For any split *s=xyz*,
Show that you cannot satisfy all three conditions

5 Conclude that ***s* cannot be pumped** ⟶ **A is not regular**

Wait…

What happens if A is a finite language?!

Pumping lemma

If ***A*** is a regular language, then

there exist a number p s.t.

Any string in A whose length is at least p

can be divided into three pieces xyz s.t.

- $xy^iz \in A$, for each $i \geq 0$ and
- $|y| > 0$ and
- $|xy| \leq p$

Pumping lemma

If ***A* is a regular language**, then

there exist a number *p* s.t.

As we saw two weeks ago,

all finite languages are regular…

So what's ***p***?

***p* := len(longest_string) + 1**

makes the lemma hold vacuously

Out of the 3 examples we saw last week

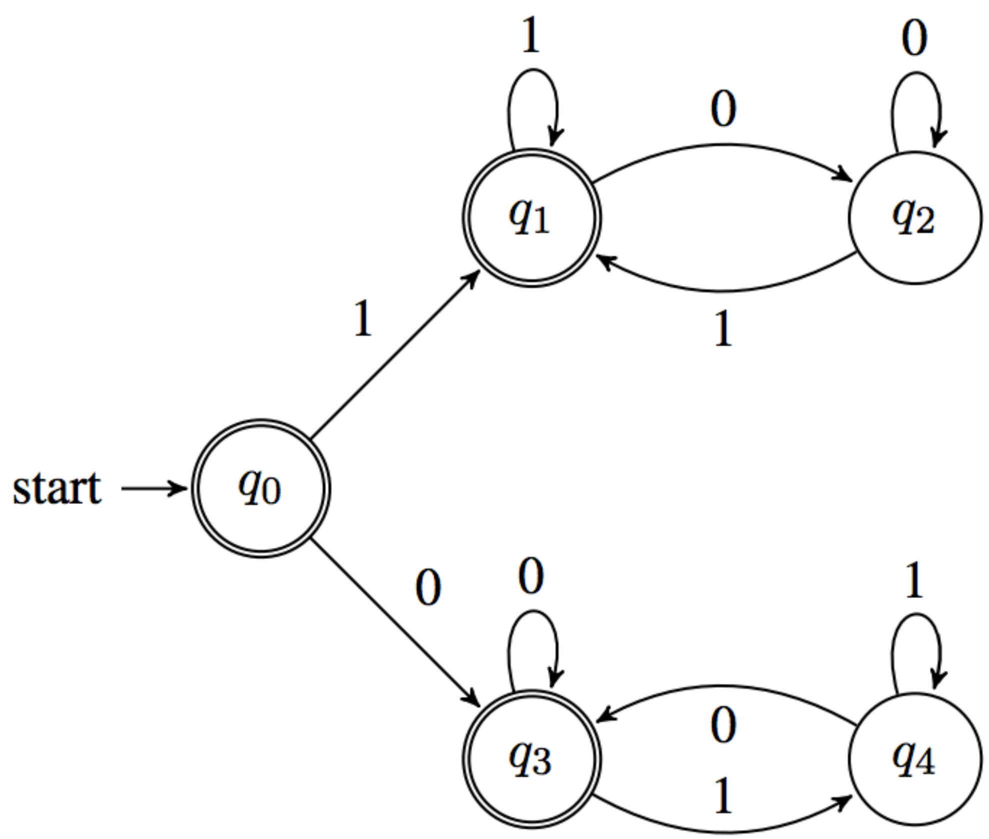
the last one is actually regular

L_1	$\{0^n 1^n \mid n \geq 0\}$
L_2	{w | w has an equal number of 0s and 1s}
L_3	{w | w has an equal number of occurrences of 01 and 10}

how do you show that? You provide a DFA/NFA/REX (you pick)

L_3 {w | w has an equal number of occurrences of 01 and 10}

101 is in L3, not 1010



Key observation

Any binary string beginning and ending with the same digit has an equal number of occurrences of the substrings 01 and 10

Non-regular languages are not closed under most operations

if L_1 and L_2 are regular,

then so are

$L_1 \cup L_2$

$L_1 . L_2$

L_1^*

if L_1 and L_2 are not regular,

then

$L_1 \cup L_2$

$L_1 . L_2$

L_1^*

may or may not be regular!

$(L_1)^C$ is not regular

non RL are closed under complement

This week is all about

Context-Free Languages

a superset of Regular Languages