The Pumping Lemma

Regular Expressions

- We have seen that Regular Expressions can be used to describe the set of permitted lexemes in a programming language

  Identifier: [a-zA-Z][a-zA-Z]*

  Number: [1-9][0-9]*

- We have also seen how to represent REs by NFAs and DFAs and how this leads to an implementation for the matching problem
- Can we also use REs to go beyond single lexemes?
- Yes, we can describe an entire statement from a programming language

  abc = 344;

  by a RE like

  Statement: *Identifier* "=" *Number* ";"

- But there is a limit....

Limitation of Regular Languages

- Can we describe complex expressions by a RE, for example

  ((a+3)*2+4)/7 ?

- Let's look at a simpler problem: Can we give a RE for all strings consisting of a number of opening parenthesis followed by the same number of closing parenthesis, i.e.,

  (), (()), ((())), (((()))), ...

- More general: Is the language over alphabet $\Omega = \{\mathrm{a}, \mathrm{b}\}$

  $$\{ a^n b^n \mid n \in \mathbb{N}\}$$

  a regular language?

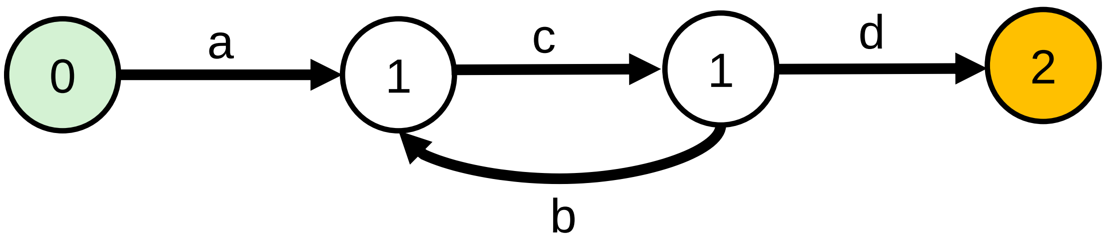
- The answer is: no!

The Pumping Lemma

- Let L be a regular language
- The Pumping Lemma says: There exists a natural number $p \geq 1$ for L such that every sequence of characters $s \in L$ with length $\geq p$ can be decomposed into three subsequences x, y, z in the form
$$s = x\ y\ z$$
with
  - length of $y \geq 1$
  - length of $x\ y \leq p$
  - for all $n \geq 0$: $x\ y^n\ z \in L$

Intuitive Proof of the Pumping Lemma

- If L is a regular language, there is a DFA with a finite number of states that accepts L
- If the DFA accepts a sequence of characters that is "very long", it must go through a loop somewhere in the DFA. Example:



The DFA accepts the sequence acd, and also (by "pumping up" acd) ac<u>bc</u>d, ac<u>bcbc</u>d, ac<u>bcbcbc</u>d, ac<u>bcbcbcbc</u>d...

Showing that $\{a^n b^n\}$ is not regular

- Assume $L = \{ a^n b^n \mid n \in \mathbb{N}\}$ is regular. Then there must be a natural number $p \geq 1$ satisfying the pumping lemma.
- Let's look at the sequence $a^p b^p \in L$. The pumping lemma says:
  - We can write $a^p b^p$ as xyz with length$(xy) \leq p$. This means that x and y only contain the letter a
  - y is not empty, i.e., $y = a^v$ where $v \geq 1$
  - We are allowed to repeat y as often as we want, i.e., $a^{p+v} b^p$ should be also in L

  $\Rightarrow$ Contradiction
- Conclusion: $L = \{ a^n b^n \mid n \in \mathbb{N}\}$ is not a regular language
- If we want to describe L in a formal way, we need something more powerful than regular expressions and NFAs/DFAs

- Warnung: If L is regular it can be "pumped", but the opposite is not true: There are "pumpable" languages that are not regular.