Lecture 9: Proving Languages not to be Regular

Thorsten Altenkirch
based on slides by Neil Sculthorpe

Room A10
School of Computer Science
University of Nottingham
United Kingdom
txa@cs.nott.ac.uk

13th March 2012
The Regular Languages are those that can be recognised by finite automata.

Such machines have a finite number states (i.e. finite memory).

But many languages are not regular.
The Chomsky Hierarchy

All languages

Recursively Enumerable Languages (Type 0)  Turing Machines

Recursive/Decidable Languages  Total Turing Machines / Deciders

Context-Sensitive Languages (Type 1)  Linear-Bounded Turing Machines

Context-Free Languages (Type 2)  Pushdown Automata

Regular Languages (Type 3)  Finite Automata
How do we prove a language is not regular?

One technique: Using The Pumping Lemma

Basic idea: Exploit the fact that, for any Regular Language, sufficiently long words are repetitive.
The Pumping Lemma for Regular Languages

Given a regular language $L$, there exists an $n \in \mathbb{N}$ such that all $w \in L$ of length at least $n$ can be split into three words ($w = xyz$) satisfying:

- $y \neq \varepsilon$
- $|xy| \leq n$
- $\forall k \in \mathbb{N}. \ xy^kz \in L$
Recommended Reading

- Introduction to Automata Theory, Languages, and Computation (3rd edition), pages 127–131
- G52MAL Lecture Notes, pages 29–31