

FIT2014

Theory of Computation

Lecture I Introduction

Slides by David Albrecht (2011), modified by Graham Farr (2013).

All materials produced for teaching this course of study, including all lectures and any supplementary materials are protected by copyright. You are permitted to use these materials only for your personal study and research. Use of the materials for any other purposes, including sale of your personal lecture notes, without express permission of the copyright owner, may infringe copyright. The copyright owner may take action against you for infringement.



Overview of today's lecture

- Important information
- Language
- Examples
- Terminology
- Definitions, Theorems, Proofs
- Other things we will cover in the course.

People

Lecturers (Clayton campus)

Prof. Graham Farr

Graham.Farr@monash.edu

Room 139, Building 63

Dr Rebecca Robinson

Rebecca.Robinson@monash.edu

Lecturer (Malaysia campus)

Assoc. Prof. KokSheik Wong

Wong.KokSheik@monash.edu

Head Tutor (Clayton)

Dr Rebecca Robinson

Tutors (Clayton)

Harald Bögeholz

Pooja Pancholi

Ben Jones

Grant Sinclair

Roger Lim

Rebecca Robinson

Raphael Morris

Rebecca Young

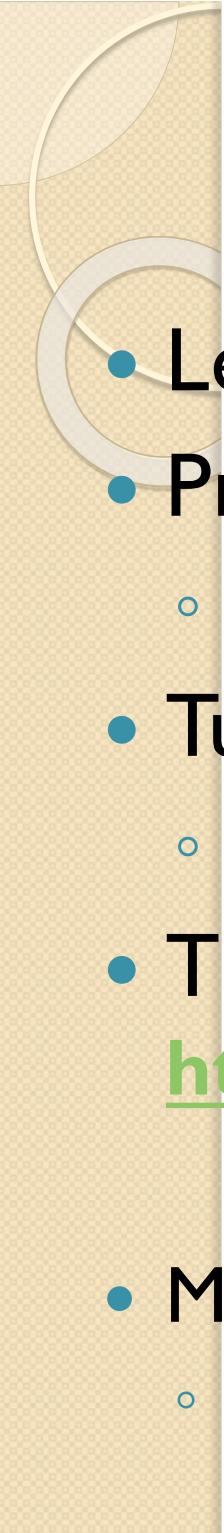
Text Books

Recommended Text:

- Michael Sipser, ***Introduction to the Theory of Computation***, PWS Publishing Company, 2006

Also useful:

- Daniel I.A. Cohen, ***Introduction to Computer Theory***, 2nd Edition, Wiley, New York, 1997.
 - ISBN-10: 0471137723.



Lectures, Pracs and Tutes

- Lectures are on the web
- Pracs (2 hours) - start week 2 (Lab 0, on Linux)
 - Weeks 2, 4, 10,
- Tutes (2 hours) - start week 3 (Tute 1)
 - Weeks 3, 5, 7, 8, 9, 11, 12
- Timetable:
<http://www.monash.edu.au/timetables/allocate/>
- Mid-Semester Test – week 6
 - No tutes/labs that week

Assessment

- Tutorial preparation (5% total)
 - all tutorials: weeks 3, 5, 7, 8, 9, 11, 12.
 - Each tutorial has a nominated preparation exercise.
 - You must make a serious attempt at this question, although it does not need to be entirely correct.
 - Your attempt will be assessed at the start of each tutorial. You get 1 mark for a serious and clear attempt; 0 marks otherwise.
 - Maximum 7 marks for the semester.
-

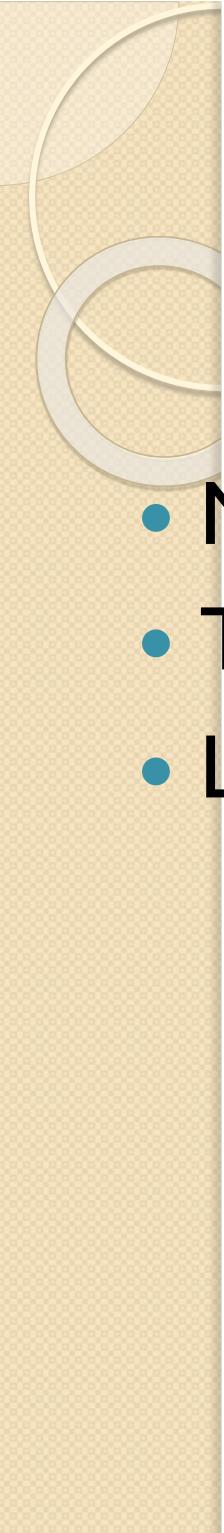
Assessment (continued)

- Tutorial preparation (5%) (see previous slide)
- Assignment 1 (5%)
 - due Fri 11:59pm in week 4
- Mid-Semester Test (10%)
 - on Wed 4 September, at 3pm, in week 6
- Assignment 2 (20%)
 - due Fri 11:59pm in week 10
- Exam (60%)



Assessment *Hurdles*

- Total of non-Final-Exam assessments:
 - two Assignments
 - + Mid-Semester Test
 - + tute preparation:
at least 40% of available marks
- Final Exam: at least 40% of available exam marks
- Overall: at least 50%
- Submit all assignments.



Where to get Information

- Moodle Page
- Tutors
- Lecturer

Languages

- Natural Languages
 - English, Chinese, French, Auslan, etc.
- Programming Languages
 - Java, C, MIPS, Python, BASIC, Fortran, Pascal, etc.
- Mathematics
- State Diagrams
- Music
- Feynman Diagrams



Components

- Alphabet
 - Basic elements
- Rules
 - Grammar
 - Tell you what words belong to the language
 - syntax
- Meaning
 - semantics



Definitions

- **Alphabet**
 - Fundamental building blocks
 - set of letters or characters
- **Word**
 - a finite string of characters belonging to the language
- **Language**
 - set of words (vocabulary)



English: Words

- Alphabet

a b c .. z A B .. Z ' -

- Words

- all the words in a standard dictionary
- e.g.

aardvark, zygote, woot,

[but not w00t]

don't, context-free, regular, ...



English: Sentences

- Alphabet
 - English words
 - punctuation marks
?! , ‘ ’ ;:
 - blank space
 - Words
 - sentences
- What is on the exam?**
- The quick brown fox jumps over the lazy dog.**

Java

- Alphabet
 - **Unicode characters**
- Words
 - **programs**

```
class HelloWorld {  
    public static void main (String args[ ]) {  
        System.out.println("Hello World!");  
    }  
}
```

Assumptions

- Alphabet
 a b
- Words
 a b ab abbb bab ...
- Notation
 a^2 means aa
 b^3 means bbb
- Empty Word ϵ
- Empty Language ϕ



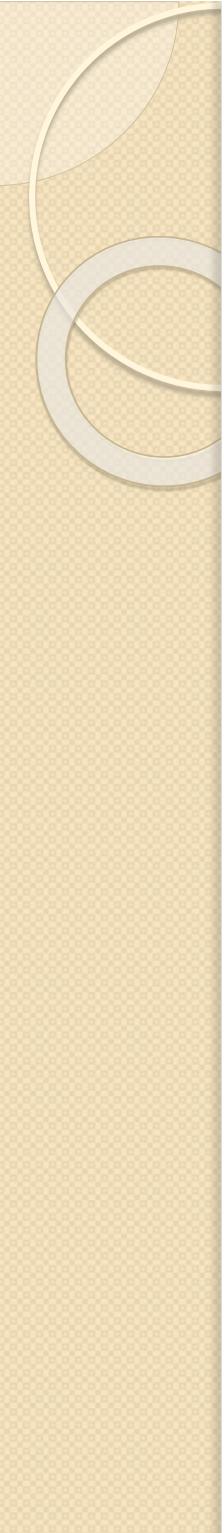
EVEN-EVEN

- All the strings that contain an even number of **a**'s and an even number of **b**'s.
- E.g.
 - ε aa bb aaaa aabb abab abba baab**



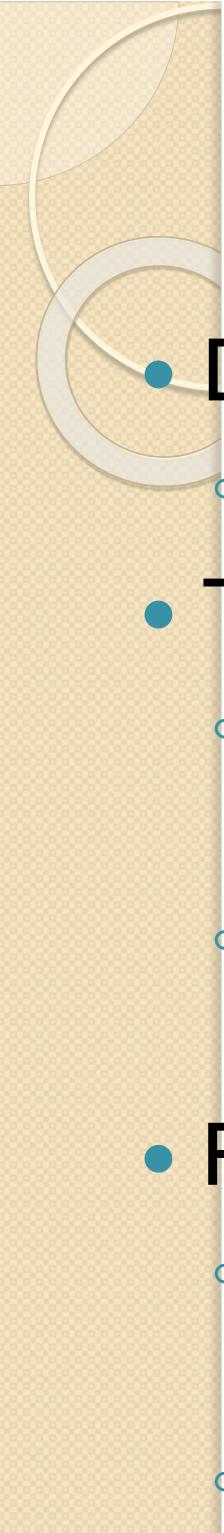
PALINDROME

- All the strings which are the same if they are spelt backwards
- E.g.
 $\epsilon \text{ a b aa bb aaa aba bab bbb}$



DOUBLEWORD

- All the strings are formed by two copies of a string joined together.
- {ss : s is a string of **a** and **b**}
- E.g.
ε aa bb aaaa abab baba bbbb



Definitions, Theorems, Proofs

- Definition
 - specifies the precise meaning of certain things.
- Theorem
 - a mathematical statement that has been proved to be true.
 - has some close but “less significant” relatives:
Proposition, Lemma
- Proof
 - A step-by-step argument that establishes, logically and with certainty, that something is true.
 - Should be verifiable.

Examples of theorems and proofs

Theorem

English has a palindrome.

(i.e., a word that is the same forwards as backwards)

Proof

'rotator' is an English word and also a palindrome. Q.E.D.

Existential statement ...

*There exists a palindrome in English
... just requires one suitable example for a proof.*

Most proofs are not this short ...



Examples of theorems and proofs

Theorem

Every English word has a vowel or a 'y'.

Proof

'aardvark' has a vowel.

'aardwolf' has a vowel.

'aasvogel' has a vowel.

...

...

'syzygy' has a 'y'.

...

'zygote' has a vowel. Q.E.D.



Theorems and proofs

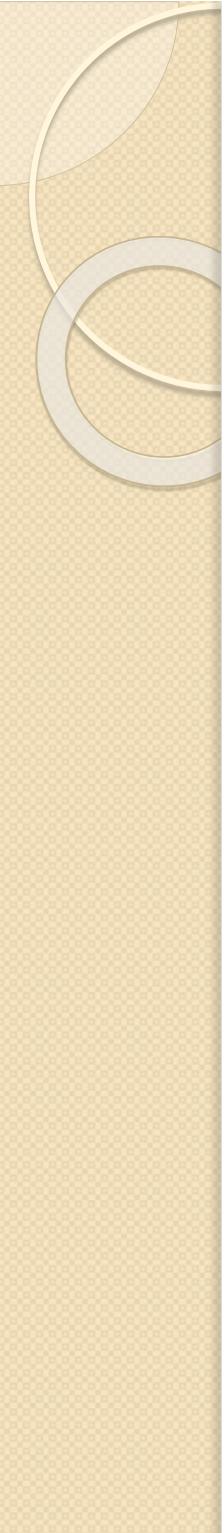
To prove a *universal* statement ...

For every English word, it has a vowel
... you need to cover every possible case.

One way is to go through all possibilities, in turn, and check each one.

But the number of things to check may be huge, or infinite.

So usually we want to reason in a way that can apply to many different possibilities at once.



Another example Theorem ...

Theorem

DOUBLEWORD \subseteq EVEN-EVEN

Another example Theorem ...

Theorem

$$\text{DOUBLEWORD} \subseteq \text{EVEN-EVEN}$$

Non-Proof

The examples on the previous slide show that every member of DOUBLEWORD has an even number of a's and an even number of b's.

So every member of DOUBLEWORD is also a member of EVEN-EVEN.

~~Q.E.D.~~

“Proof by example” is not a proof.

... except where the Theorem just asserts the existence of a specific example!

Another example Theorem and Proof

Theorem

$\text{DOUBLEWORD} \subseteq \text{EVEN-EVEN}$

Another example Theorem and Proof

Theorem

$$\text{DOUBLEWORD} \subseteq \text{EVEN-EVEN}$$

Proof

Let $w \in \text{DOUBLEWORD}$

Then $w = xx$ for some word x .

So, # a's in $w = 2 \times (\# \text{a's in } x)$, so it's even.

Also, # b's in $w = 2 \times (\# \text{b's in } x)$, so it's even too.

So $w \in \text{EVEN-EVEN}$.

Q.E.D.



Halting Problem (Entscheidungsproblem)

- Consider the following problem:

Design a program HALT which takes as input a program, P, and data, D, and returns true if P halts on the data D, and false otherwise.



Other Topics

- Propositional logic
- Predicates, quantifiers
- Linux
- Regular languages, finite automata
- Grammars, context-free languages, pushdown automata
- Lexical Analysis
- Introduction to parsing
- Turing Machines
- Computability, decidability
- Computational complexity
- Classes P and NP
- NP-completeness



Reading

- Sipser, pp 13-14
 - strings and languages
- Sipser, § 0.3, pp 17-20
 - definitions, theorems, proofs