university of groningen

Languages and Machines

Lecture 0: Introduction

Jorge A. Pérez

Bernoulli Institute for Mathematics, Computer Science, and AI
University of Groningen, Groningen, the Netherlands

April 18, 2023

L&M, In a Nutshell

- A course on **automata theory and formal languages**
- Lecturers: Dr. Dilek Düstegör, Dr. Jorge Pérez
- Lectures (at least one per week) and tutorials (one per week)
- We assume you have passed (and still remember!)
  - Introduction to Logic
  - Discrete Structures (in particular: proofs by induction)
- Assessment: Three homeworks and a final exam
- Helpdesk email: `lm23.rug[at]gmail.com`

The Foundations of Computation

Basic questions:

- What does it mean for a function to be computable?
- Are there any non computable functions?
- Computational power $\leftrightarrow$ Programming constructs?

The Foundations of Computation

Basic questions:

- What does it mean for a function to be computable?
- Are there any non computable functions?
- Computational power $\leftrightarrow$ Programming constructs?

Looking for answers $\rightarrow$ Fundamental concepts

- State
- Transition
- Non-determinism
- Undecidability
- $\ldots$

Persistent concepts, despite many (and frequent) technology changes

Models of Computation

In order of increasing power:

(a) Finite Memory:
Finite automata; regular expressions

(b) Finite Memory with stack:
Pushdown automata

(c) Unrestricted:
Turing machines (terminating and non-terminating)

Grammars and Languages

The **Chomsky hierarchy** - in order of increasing complexity:

(i) Right-linear grammars

(ii) Context-free grammars

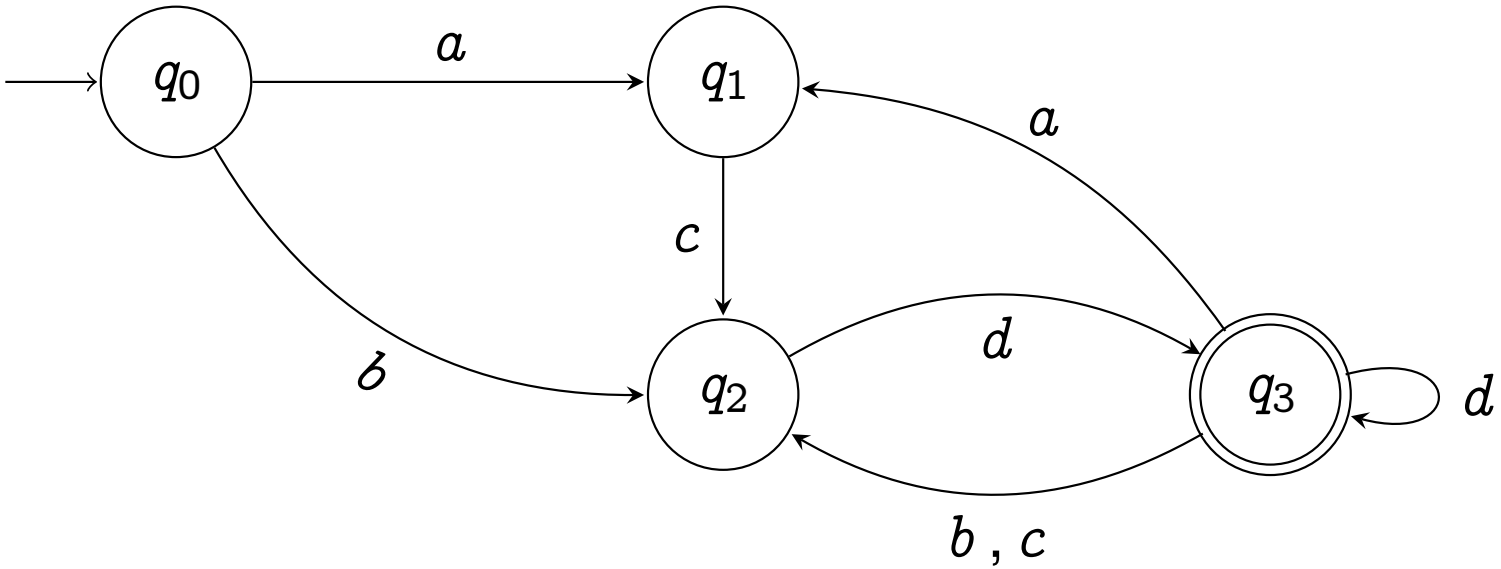
(iii) Unrestricted grammars

Grammars and Machines

- Superficially very different
- Parsing a sentence in a language quite similar to computation
- Grammar types (i)-(iii) are **equivalent** to machines (a)-(c)!

State-based systems are everywhere!

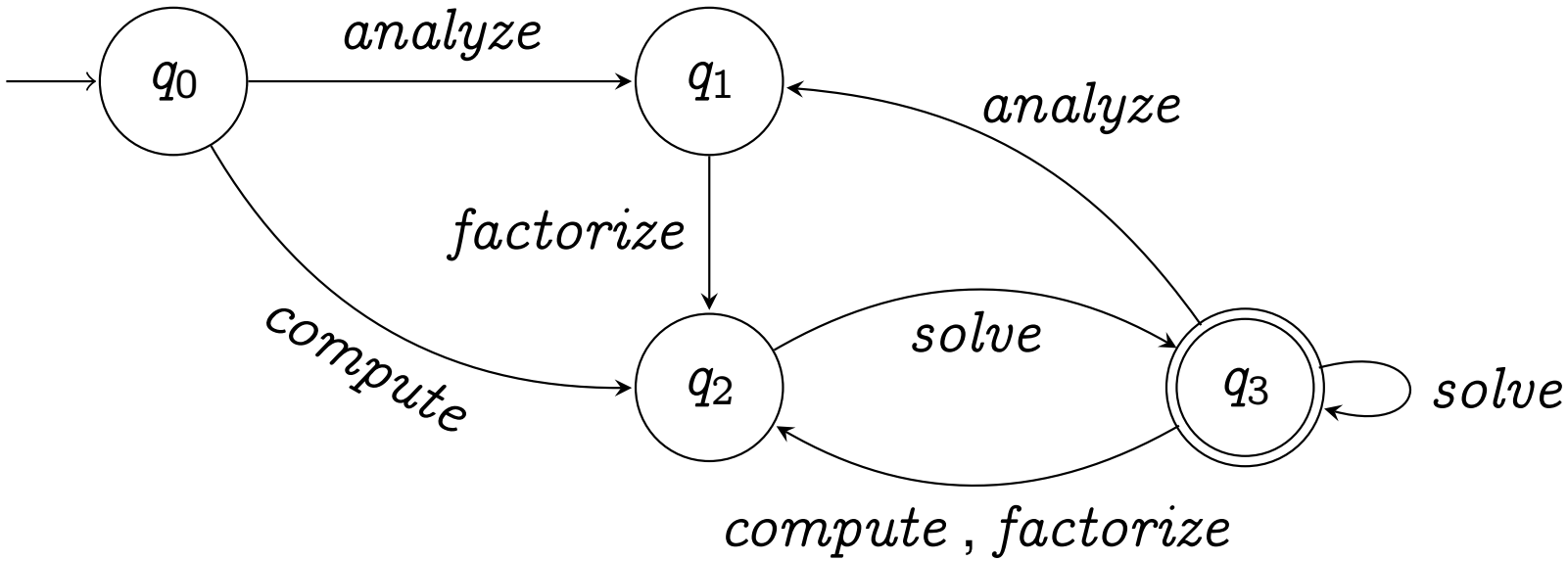
A finite-state machine



- Above, q_0, q_1, q_2, q_3 are states of the machine.
  Symbols a, b, c, d are recognized by moving between states.
  The machine recognizes a certain language (set of strings).

State-based systems are everywhere!

A finite-state machine... can also be a rigorous specification for verifying object-oriented programs!



- Above, q_0, q_1, q_2, q_3 are states of the machine.
- By interpreting symbols a, b, c, d as class methods, we can specify the sequences of allowed invocations.
  (This is called a *typestate*.)

Many applications

- Programming language design and implementation
  (Compiler construction, domain specific languages, etc)
- Software and hardware verification
  (Model checking, run-time verification, etc)
- Learning and AI
- Bioinformatics
- Security
- . . .

Overview

In this course, we will describe, analyse, and classify the languages that can be read by machines and the machines that can read them.

The course does not concern the interpretation of such languages.

Roughly, seven parts:

1. Regular and context-free languages
2. Finite state machines
3. Properties of regular languages
4. Pushdown machines
5. Properties of context-free languages
6. Turing machines
7. Decidability issues

Learning Outcomes (Ocasys)

1. The student learns to understand and apply:
   (a) The basic theory of finite state, pushdown, and Turing machines, and of the regular, context-free, and decidable and semi-decidable languages.
   (b) The relationships between machines and languages, and the translation algorithms between the various representations (e.g. regular expressions, normal forms of grammars).
2. The student obtains an elementary understanding of decidability, undecidability, semi-decidability, computability, time complexity, the classes P and NP, and the Chomsky hierarchy.

Material

"If it ain't broke, don't fix it"

- Lecture Notes "Languages and Machines" by Wim Hesselink
  Contains many exercises, some of which are done at the tutorials
  PDF available in Brightspace: you comments are welcome!
- Course slides follow those by Jasper van de Gronde

Many good textbooks around!

Teaching Method

On our side:

- In-person lectures (at most twice per week).
- Tutorials (once per week).

Schedules in Brightspace (subject to changes).

Teaching Method

On our side:

- In-person lectures (at most twice per week).
- Tutorials (once per week).

Schedules in Brightspace (subject to changes).

On **your side**: self-study!

- Studying the reader **before** the lecture
  Consult textbooks as needed
- Work on the exercises **before** the tutorials, and bring questions
- Look into topics not covered (or partially covered) in the course

... but also:

- Handing homeworks on time
- Proactive, constructive feedback to the lecturer and TAs

Grading (Ocasys)

Components

1. H: Individual homeworks
   Four homeworks: Only the first three are mandatory (graded).
2. E: Exam

Your Final Grade

$$F = 0.6 \times E + 0.4 \times H$$

There is also a resit. Notice that H does not count at the resit.

Important Dates

Four homeworks:

- Deadline on Fridays, 10h:
  April 28, May 12, May 26, and June 9.
- See Brightspace for instructions / modifications.

Exam and Resit

- See the rooster for dates/times.

Contact Us

1. General questions: send us an email
   lm23.rug[at]gmail.com
2. Specific questions, feedback, requests for (online) meetings
   j.a.perez[at]rug.nl, d.dustegor@rug.nl

These are confusing times. Feel free to reach out to the academic advisor for Computing Science (Korrie / Femke):

- Email: academicadvisor.cs[at]rug.nl
- Book your own appointment via
  https://fse.as.me/KorrieBonnema
  https://fse.as.me/femkeschouten

Plagiarism and Scientific Integrity

Study Guide Computer Science, section on “Fraud Prevention & Scientific Integrity” — https://student.portal.rug.nl/infonet/studenten/fse/programmes/bsc-cs/:

> *Plagiarism is not accepted at this university nor elsewhere in the scientific community.*
> *In all cases in which plagiarism is found or suspected, the examiner will inform the Board of Examiners.*
>
> *Possible consequences:*
>
> - *Warning*
> - *Exclusion from exams for the relevant course for 1 academic year*
> - *Exclusion from exams for several courses for 1 academic year*
> - *Exclusion from programme*

The End