Fundamentals of Data Processing and Distributed Knowledge

Pierre Genevès (CNRS) and Jérôme Euzenat (Inria)

University Grenoble Alpes

Course Introduction

Modern computing increasingly takes advantage of large amounts of distributed data and knowledge.

This is grounded on theoretical principles borrowing to several fields of computer science such as

- programming languages
- data management
- logic and artificial intelligence

Goals: present some of the most essential theoretical principles, the problems that they solve and those that they uncover.

Course Objectives

Introducing Fundamentals about:

• Expressing information	→ Languages
• Processing it	→ Algorithms
• in the most correct, efficient	→ Logic
and meaningful way	→ Semantics

Organization: Important URLs

Course

- Course website: `https://moex.inria.fr/teaching/fdk/`
- Slides for this part: `http://pierre.geneves.net/teaching.html`

Project proposals

- Time to start looking for an internship
- Pcarre website: `http://im2ag-pcarre.e.ujf-grenoble.fr/`
- Do not hesitate to look around on your own. E.g. in our team (tyrex.inria.fr), topics at the crossroads between PL and AI (graph information extraction, neuro-symbolic queries, etc.)

→ Project must be defended in June to qualify for PhD scholarships on academic merit given by the Doctoral School MSTII (`https://edmstii.univ-grenoble-alpes.fr`)

Two Perspectives on Data and Knowledge

1. **Foundations for Processing Trees (15h), Pierre Genevès (DR CNRS)**

2. Distributed Knowledge (15h), Dr. Jérôme Euzenat (DR Inria)

Introduction to First Part

Tree-shaped data

→ data model very widely used on the web (and crucial in Computer Science)

- Two particularities: **order** et **hierarchy**, make trees fundamentally different from more classical relational structures such as tables.
- This part introduces foundations for processing trees:
  - → How to effectively query these structures
  - → Foundations / theoretical and algorithmic tools (tree automata, tree logics) at the heart of theoretical computer science
  - → Concrete examples for the analysis of expressive queries, checking data consistency, etc. illustrated with XML technologies

Introduction to First Part

Tree-shaped data

→ data model very widely used on the web (and crucial in Computer Science)

- Two particularities: **order** et **hierarchy**, make trees fundamentally different from more classical relational structures such as tables.
- This part introduces foundations for processing trees:
  - → How to effectively query these structures
  - → Foundations / theoretical and algorithmic tools (tree automata, tree logics) at the heart of theoretical computer science
  - → Concrete examples for the analysis of expressive queries, checking data consistency, etc. illustrated with XML technologies

Introduction to First Part

Tree-shaped data

→ data model very widely used on the web (and crucial in Computer Science)

- Two particularities: **order** et **hierarchy**, make trees fundamentally different from more classical relational structures such as tables.
- This part introduces foundations for processing trees:
  - → How to effectively query these structures
  - → Foundations / theoretical and algorithmic tools (tree automata, tree logics) at the heart of theoretical computer science
  - → Concrete examples for the analysis of expressive queries, checking data consistency, etc. illustrated with XML technologies

Introduction to First Part

Tree-shaped data

→ data model very widely used on the web (and crucial in Computer Science)

- Two particularities: **order** et **hierarchy**, make trees fundamentally different from more classical relational structures such as tables.
- This part introduces foundations for processing trees:
  - → How to effectively query these structures
  - → Foundations / theoretical and algorithmic tools (tree automata, tree logics) at the heart of theoretical computer science
  - → Concrete examples for the analysis of expressive queries, checking data consistency, etc. illustrated with XML technologies

Introduction to First Part

Tree-shaped data

→ data model very widely used on the web (and crucial in Computer Science)

- Two particularities: **order** et **hierarchy**, make trees fundamentally different from more classical relational structures such as tables.
- This part introduces foundations for processing trees:
  - → How to effectively query these structures
  - → Foundations / theoretical and algorithmic tools (tree automata, tree logics) at the heart of theoretical computer science
  - → Concrete examples for the analysis of expressive queries, checking data consistency, etc. illustrated with XML technologies

Introduction to First Part

Tree-shaped data

→ data model very widely used on the web (and crucial in Computer Science)

- Two particularities: **order** et **hierarchy**, make trees fundamentally different from more classical relational structures such as tables.
- This part introduces foundations for processing trees:
  - → How to effectively query these structures
  - → Foundations / theoretical and algorithmic tools (tree automata, tree logics) at the heart of theoretical computer science
  - → Concrete examples for the analysis of expressive queries, checking data consistency, etc. illustrated with XML technologies