

# Foundations of XML Types: An Introduction

Pierre Genevès  
CNRS

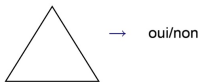
University of Grenoble Alpes, 2016–2017

# 3 Essential Tasks & Related Languages

## Validation

Check that an XML document is valid w.r.t. a given type

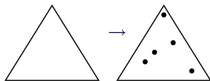
DTD, XML Schema...



## Navigation/Extraction

Select a set of nodes from an XML document

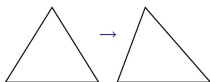
XPath



## Transformation

Build a new document from an existing one

XQuery (or XSLT)



## Some Essential Questions

- Can I express structural requirement X using XML type language Y? (Expressive Power)
- Can I check forward-compatibility when my XML file format evolves? Can I check type inclusion? (Operations over XML Types)
- Can we make sure my XML manipulating program will never output an invalid document? (Static Type-Checking)

## Some Essential Questions

- Can I express structural requirement X using XML type language Y? (Expressive Power)
- Can I check forward-compatibility when my XML file format evolves? Can I check type inclusion? (Operations over XML Types)
- Can we make sure my XML manipulating program will never output an invalid document? (Static Type-Checking)

For all these problems, we must know more about XML Types...

## Objectives of this Session

- Get a higher level perspective on XML type languages (DTD, XML Schema...)
- Know more about their theoretical foundations (where they do come from)
- Discover a general framework in which all these languages fit
- Understand better their relative expressive power
- Get an overview of some helpful theoretical tools, enrich your algorithmic toolbox
- Get a flavour of research ongoing in this area
- Get some elements for answering preceding questions...

# Preliminary Question

How to measure expressive power?...

# Preliminary Question

How to measure expressive power?...

## Some Answers

- Computational complexity classes provide a graduation
- Corresponding known problems (e.g. classes of logical formulas) give hints, and also provide help to prove that something cannot be expressed in a given language

# Recall: Computational Complexity

