

# Logical dependencies between classes: how to find them and how to use them ?

Adelina Diana Stana

Department of Computer and Information Technology, Politehnica University of Timisoara, Romania

## Definition

A **dependency** is a relationship that shows that an element, or set of elements, requires other elements for their specification or implementation. [ UML Specification]

## Structural dependencies

Structural dependencies are the result of source code analysis and can be extracted from : members, call parameters, local variables.

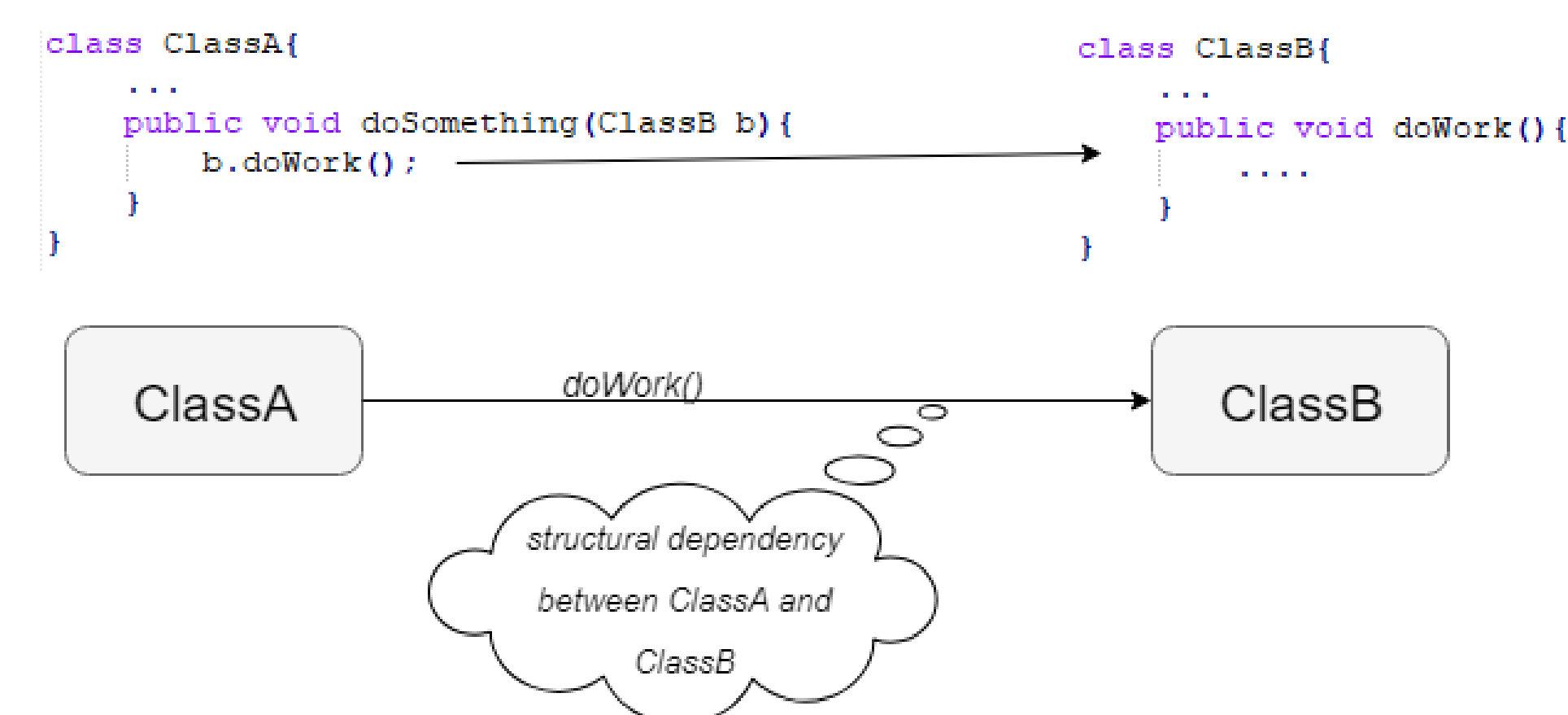


Figure 1: Example of structural dependency between two classes

## Logical dependencies

Logical dependencies are the result of software history analysis and can reveal relationships that are not present in the source code code (structural dependencies).

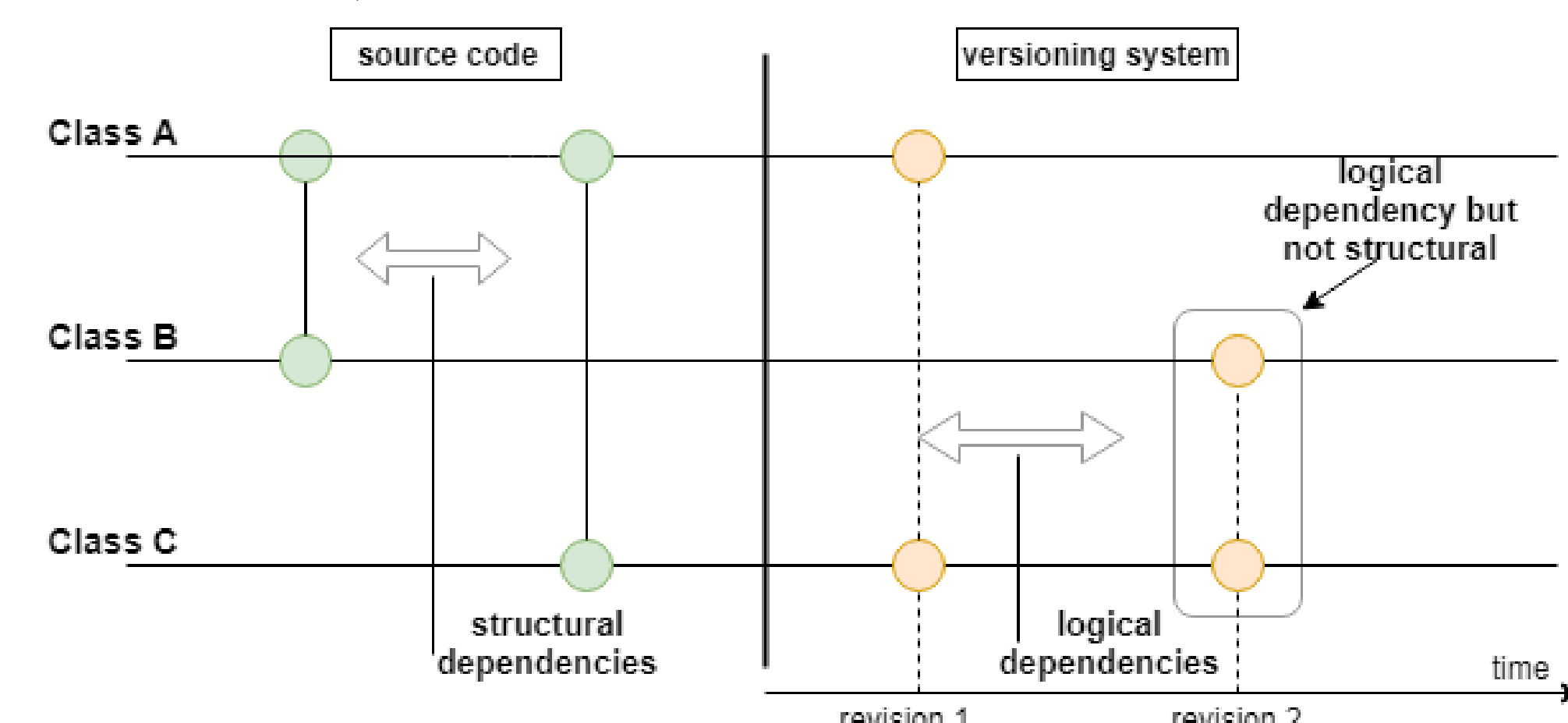


Figure 2: Example of logical and structural dependencies

## Research questions

We build logical dependencies based on three questions :

**Question 1:** How the number of files changed in a commit can influence the logical dependencies of the system?

**Question 2:** Considering comment changes as valid changes can lead to additional logical dependencies ?

**Question 3:** One occurrence of a logical dependency is enough to consider it as valid ?

## Tool for measuring software dependencies

In order to answer these research questions, we have built a tool that extracts structural and logical dependencies.

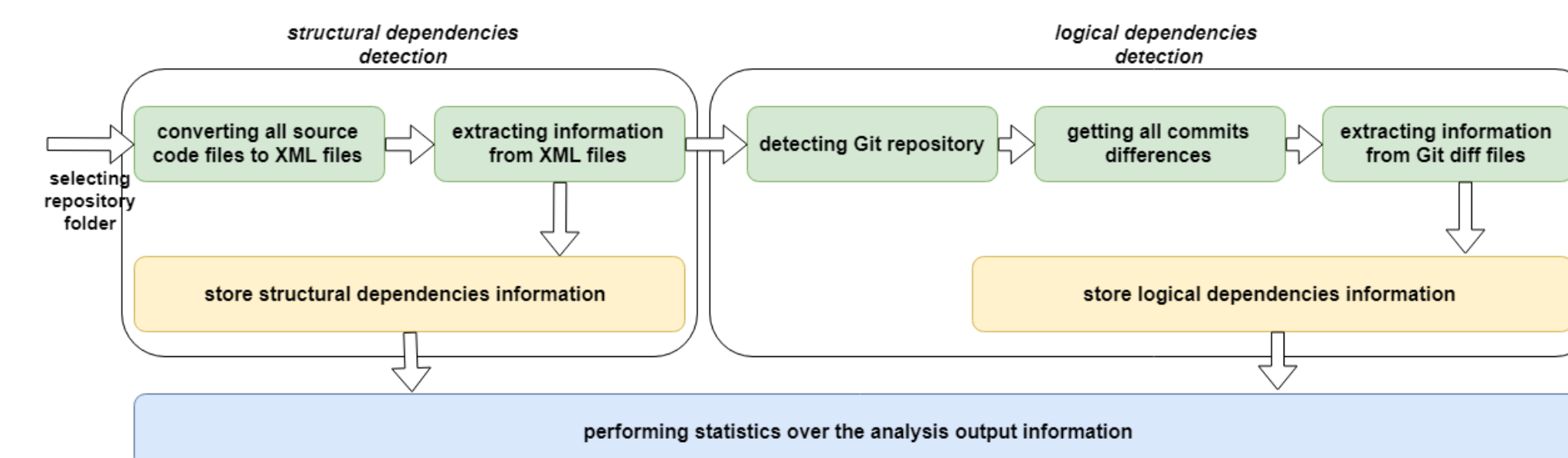


Figure 3: Workflow diagram of the tool

The workflow can be delimited by three major steps as it follows:

Extracting structural dependencies.

Extracting logical dependencies.

Processing the information extracted.

## Experimental conditions

For each system, we extracted its structural dependencies, its logical dependencies and determined the overlap between the two dependencies sets, in various experimental conditions:

- code comments
- size of commit
- number of LD occurrences

## Experimental results

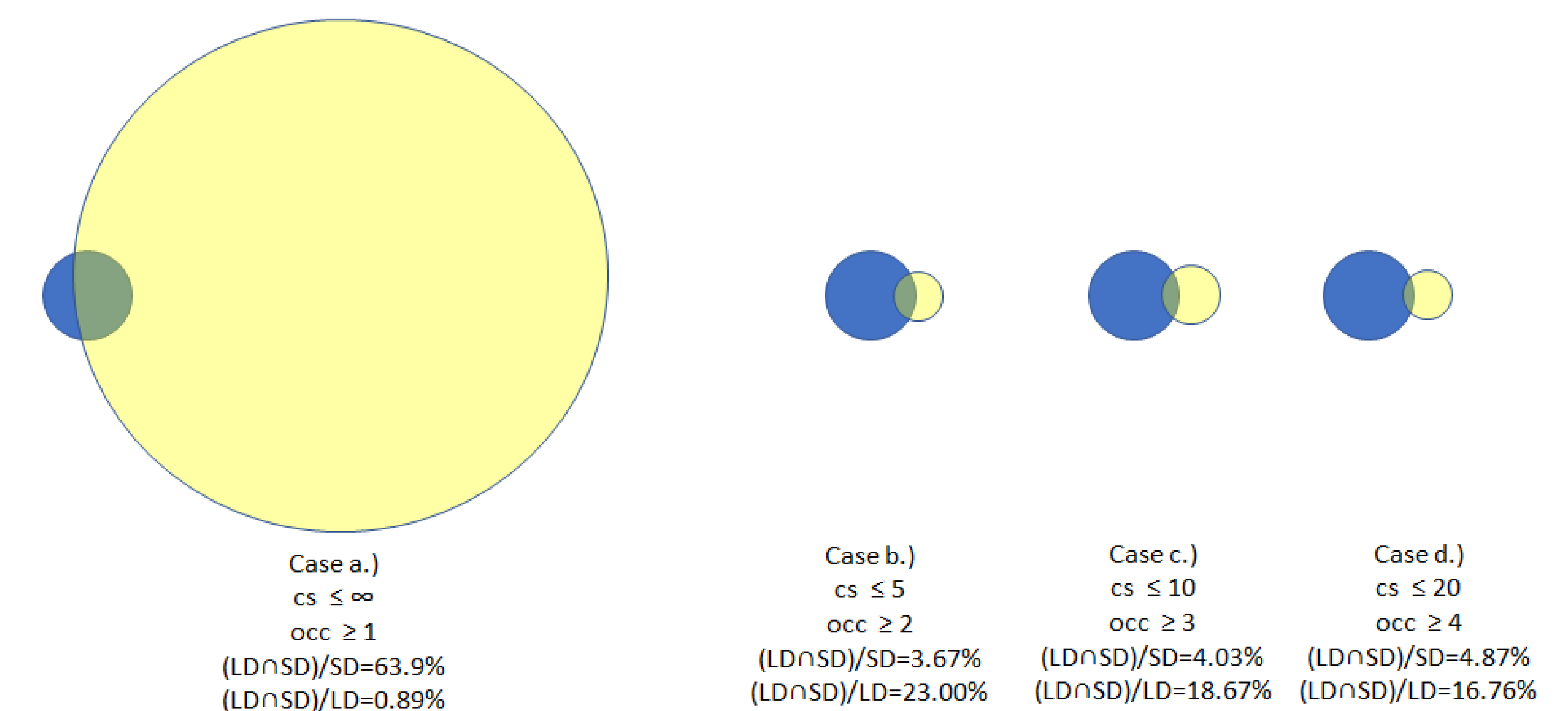


Figure 4: Intersections of logical and structural dependencies, in different cases defined by different combinations of filtering thresholds.

## Conclusions and Future work

- Large number of structural dependencies are not doubled by logical - systems partially stable
- + -3% for comments as a change
- The number of changed files taken into consideration influence the results
  - big threshold - not so relevant logical dependencies
  - small threshold (5 10) - more accurate results
- Filtering the logical dependencies after occurrences is good only for projects with a significant number of commits.

Investigate the cause for the large number of logical dependencies which are not overlapping with structural dependencies.