Clone Detection

Software Evolution — L5T3

Dr. Vadim Zaytsev aka @grammarware, March 2021

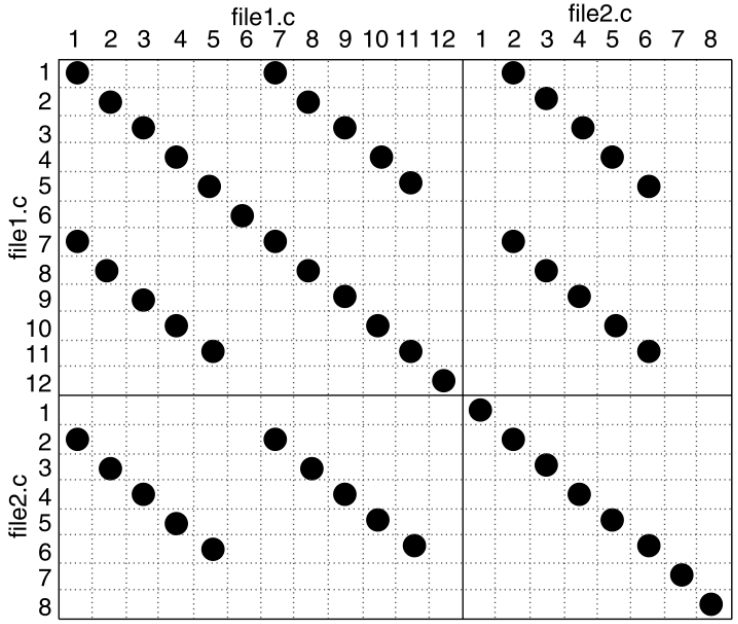
UNIVERSITY OF TWENTE.
Formal Methods & Tools

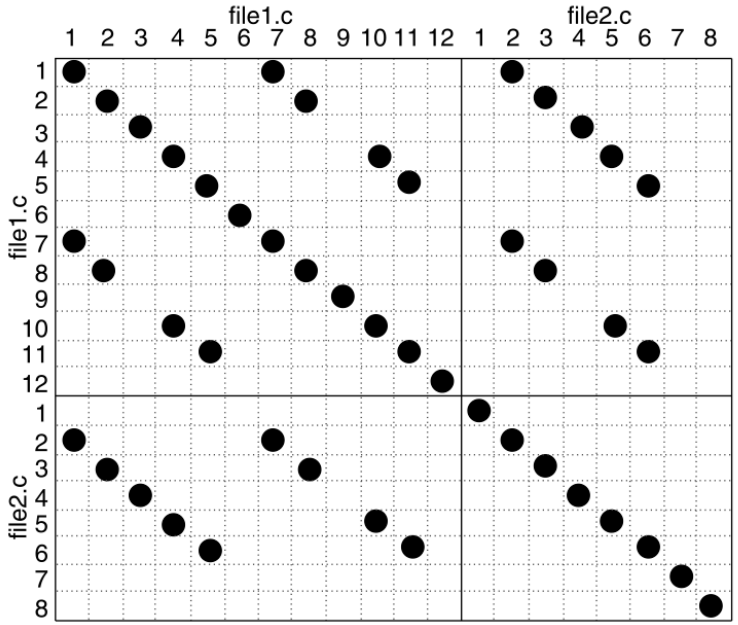
Definitions

- **Clone**
  - fragment of code that is duplicated elsewhere
- **Clone pair**
  - two code fragments that are duplicates of each other
- **Clone class**
  - any number of code fragments that are all duplicates of one another

Clone Detection on Str

- Compare each line with each line
- Use hashing to speed up
- Extend cloned lines into bigger fragments

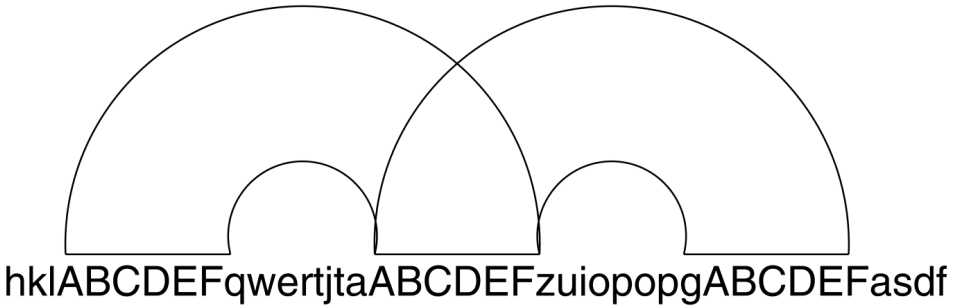




recall L3P1; *Software Evolution*, Chapter 2

Clone Detection on Tok

- Apply lexical analysis to obtain tokens
- Normalise tokens
- Use an efficient algorithm (e.g., suffix trees)
- Variants will work on
  - **Lex** – without normalisation
  - **Reg** – balancing normalisation



recall L3P1; *Software Evolution*, Chapter 2

Clone Detection on Ast

- Parse the code fully to construct an AST
- Find identical subtrees
- Ignoring leaves will detect Type 2 clones
- Bespoke matching of some nodes will get Type 3
- Staying with **Ptr**/**Cst** brings no benefits

recall L3P1; *Software Evolution*, Chapter 2

Clone Detection on Dia

- The most popular is PDG
  - construct the control flow graph
  - construct the data flow graph
- Find isomorphic subgraphs
  - NP-hard! approximations possible
- High quality Type 3 clones

recall L3P1; *Software Evolution*, Chapter 2

Clone Detection with Metrics

- Split the code into fragments
- Calculate a vector of metrics about each fragment
  - Size, complexity, number of calls, …
- Measure distances between each two vectors

recall L3P1; *Software Evolution*, Chapter 2

A Tool for Each Technique!

- (**Str**) Simian http://www.harukizaemon.com/simian
- (**Tok**) CCFinder http://www.ccfinder.net
- (**Tok**) SourcererCC
  - https://github.com/Mondego/SourcererCC
- (**Ast**) Deckard https://github.com/skyhover/Deckard
- (**CFG**) JDiff http://javadiff.sf.net
- (**PDG**) Duplix, CodeSurfer, CC#, CCGraph
- . . .

Conclusion

- Clone detection can be reasonable on any level
  - text, tokens, trees, graphs, models, metrics, …
- Can be a good exercise to implement
- Clone removal is refactoring
- Q&A Sessions @ Canvas
  - ⇒ v.zaytsev@utwente.nl
  - ⇒ https://discord.gg/n7VQAPNBPD