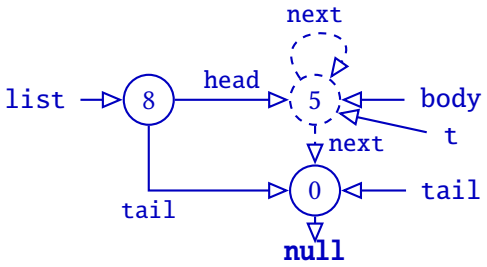
Exjobb: Precise Bug Detection with Points-To Analysis (in collaboration with Axis Communications)

Wanted: Two students for an M.Sc. project. You should have taken one of the Compilers courses, be familiar with C and (ideally) C++, and be curious about logic programming, the LLVM compiler infrastructure, and static program analysis.

Finding bugs earlier saves developers time and headaches. One of the main techniques for finding bugs is *static program analysis*, which can detect bugs by analysing the code structure, i.e., without ever running the code. In collaboration with Axis Communications, we have built a prototype tool that can detect incorrect uses of the `glib` library, which is a fundamental library used by the Gnome and GIMP Open Source projects, as well as by GStreamer, a central piece of Open Source software used by Axis.

Many of the more powerful program analyses have to understand pointers and what those pointers might point to in a given program. We can use *points-to analysis* to obtain this information.

```
makeList(len) {
[0]  tail = new()
[1]  tail.next = null
[2]  body = tail
[3]  for (; len > 0; -len) {
[4]    t = body
[5]    body = new()
[6]    body.next = t
[7]  }
[8]  list = new()
[9]  list.head = body
[10] list.tail = tail
[12] return list
```



Our existing tool already utilises a points-to analysis, but that analysis is relatively simple. Therefore, our tool sometimes mixes up two objects that it should consider separate, and reports false warnings. We thus want to replace the current analysis with a state-of-the-art points-to analysis called *cclyzer*. *Cclyzer* is implemented in the logic programming language Datalog, so combining it with our current (C++) code presents an implementation challenge.

In this project, you will:

- Familiarise yourself with our existing analysis tool and with the Phasar program analysis framework that underlies the tool
- Familiarise yourself with the cclyzer framework for points-to analysis
- Integrate cclyzer and Phasar to utilise the output of the cclyzer's points-to analysis to improve the precision of our tool's error reports
- Evaluate your changes through case studies and by sampling realistic Open Source software

This project is a collaboration between Axis Communications and the Software Development and Environments group.

Supervisors at Axis: David Svensson Fors (davidsf@axis.com) and Karin Hedlund
Supervisors at LTH: Christoph Reichenbach (christoph.reichenbach@cs.lth.se), and co-supervisor Alexandru Dura (alexandru.dura@cs.lth.se)

Contact us for details!