

# 卢静波

## 首席科学家

✉ jingbo.lu@sectrend.com.cn    💬 diluteammonia    🌐 lujingbol.github.io    📍 中国, 北京

## 🔍 研究兴趣

我的主要研究领域是编译器与编程语言。最近的研究集中在静态程序分析，着眼于开发底层的静态和动态程序分析技术和工具，以提高复杂软件系统的可用性，可靠性和安全性。

## 📄 工作经历

2023 至今 中国, 北京	<b>首席科学家</b> 上海安势信息技术有限公司 负责静态程序分析团队的组建管理和技术规划实施。内容包括 C++静态分析引擎、java 静态分析引擎、软件安全应用测试平台、AI 代码审计与漏洞修复、软件成分分析漏洞可达等
2020 - 2022 澳大利亚, 悉尼	<b>研究助理</b> 新南威尔士大学 (UNSW) 负责人: 薛京灵 卓越教授

## 🎓 教育经历

2016 - 2020 澳大利亚, 悉尼	<b>计算机科学与工程 博士</b> 新南威尔士大学 (UNSW) 导师: 薛京灵 卓越教授
2014 - 2016 澳大利亚, 悉尼	<b>信息技术 硕士</b> 新南威尔士大学 (UNSW)
2009 - 2013 中国, 天津	<b>应用物理 理学学士</b> 南开大学 (NKU) 导师: 王新宇 副教授

## </> 项目经历

2022	<p>QILIN, 细粒度上下文敏感 Java 程序分析平台</p> <p><a href="#">开源项目</a></p> <p><b>描述:</b> QILIN 是一种支持当前探索细粒度上下文敏感性的研究趋势（其中方法中的不同变量/对象可以在变量级别的不同上下文抽象下（即，通过）不同的上下文抽象进行分析）的通用（现代）选择方案，精确、高效和模块化。为了满足这四个设计目标，QILIN 采用命令式编程，由具有参数化上下文敏感性的细粒度指针分析内核组成，支持动态调用图构建和异常分析，其迭代解决基于一个精心设计的基于增量传递的约束求解器（包含复杂的 Java 特性处理）。</p>
2021	<p>CONCH, 用于加速对象敏感指针分析的上下文去膨胀技术</p> <p><a href="#">开源项目</a></p> <p><b>描述:</b> CONCH 通过验证三个线性条件（对真实应用中需要上下文敏感分析的对象来说几乎总是必要条件）来识别上下文独立对象，然后通过限制这些对象组合出新上下文（组合后的上下文不能提升指针分析精度）达到显著提升指针分析速度和可扩展性同时几乎不丢失分析精度的效果。</p>
2021	<p>SELECTX, 轻量级高精度调用地址敏感指针分析加速技术</p> <p><a href="#">开源项目</a></p> <p><b>描述:</b> SELECTX 通过变量和对象的上下文敏感性和其对任意程序点的虚假指针关系关联性分析以使得 <math>k</math>-CFA 可以选择性上下文敏感。</p>
2021	<p>TURNER, 在加速对象敏感指针分析的现有预分析技术中，分析精度和分析效率之间达到更佳平衡点的技术</p> <p><a href="#">开源项目</a></p> <p><b>描述:</b> TURNER 能够使对象敏感指针分析跑得比保精度的加速技术显著快同时比现有不保精度的加速技术更加精确。TURNER 是用于选择影响分析精度的变量/对象的第一个过程内预分析技术。</p>
2019	<p>EAGLE, 轻量级保精度对象敏感指针分析加速技术</p> <p><a href="#">开源项目</a></p> <p><b>描述:</b> EAGLE 使得 <math>k</math>-obj 对一个方法进行部分上下文敏感分析, 即, 上下文敏感性仅针对选定的变量和分配地址。EAGLE 通过一个借助上下文无关语言可达性（特别为 <math>k</math>-obj 创造的新形式）推理的轻量级的预分析来做出这些选择。</p>

**描述：**DRUID 是一个整合了一系列主流上下文敏感技术的指针分析框架。我们结合过去众多主流平台的优点，单针对上下文敏感指针分析的需求对控制流程作了重新设计。它的易用性和可拓展性都得到了提升。

- ECOOP'24 Dongjie He, **Jingbo Lu**, and Jingling Xue. A CFL-Reachability Formulation of Callsite-Sensitive Pointer Analysis with Built-in On-the-Fly Call Graph Construction. In 38th European Conference on Object-Oriented Programming(ECOOP'24), 2024.
- ASE'23 **Jingbo Lu**, Dongjie He, Wei Li, Yaoqing Gao, and Jingling Xue. Automatic Generation and Reuse of Precise Library Summaries for Object-Sensitive Pointer Analysis. In 38th IEEE/ACM International Conference on Automated Software Engineering (ASE'23), 2023.
- TOSEM'23 Dongjie He, **Jingbo Lu** and Jingling Xue. IFDS-based Context Debloating for Object-Sensitive Pointer Analysis. In ACM Transactions on Software Engineering and Methodology (TOSEM'23), 2023.
- TSE'22 Dongjie He, **Jingbo Lu**, Yaoqing Gao and Jingling Xue. Selecting Context-Sensitivity Modularly for Accelerating Object-Sensitive Pointer Analysis. In IEEE Transactions on Software Engineering (TSE'22), 2022.
- ECOOP'22 Dongjie He, **Jingbo Lu** and Jingling Xue. Qilin : A New Framework for Supporting Fine-Grained Context-Sensitivity in Java Pointer Analysis. In 36th European Conference on Object-Oriented Programming (ECOOP'22), 2022.
- ASE'21 Dongjie He, **Jingbo Lu** and Jingling Xue. Context Debloating for Object-Sensitive Pointer Analysis. In 36th IEEE/ACM International Conference on Automated Software Engineering (ASE'21), pages 79 – 91, 2021.
- SAS'21 **Jingbo Lu**, Dongjie He and Jingling Xue. Selective Context-Sensitivity for k-CFA with CFL-Reachability. In 28th International Static Analysis Symposium (SAS'21), pages 261 – 285, 2021.
- ECOOP'21 Dongjie He, **Jingbo Lu**, Yaoqing Gao, and Jingling Xue. Accelerating Object-Sensitive Pointer Analysis by Exploiting Object Containment and Reachability. In 35th European Conference on Object-Oriented Programming (ECOOP'21), LIPIcs, Vol. 194, pp. 16 :1 – 16 :31, 2021.
- TOSEM'21 **Jingbo Lu**, Dongjie He and Jingling Xue. Eagle : CFL-Reachability-based Precision-Preserving Acceleration of Object-Sensitive Pointer Analysis with Partial Context Sensitivity. In 2019 ACM Transactions on Software Engineering and Methodology (TOSEM'21), 30(4), 2021.
- OOPSLA'19 **Jingbo Lu** and Jingling Xue. Precision-Preserving Yet Fast Object-Sensitive Pointer Analysis with Partial Context Sensitivity. In 2019 ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'19), pages 148 :1 – 148 :29, Athens, 2019.

