שם הקורס: Program Analysis for Developers

ניתוח קוד לצרכי פיתוח

סמינר במדמ״ח 1

מספר הקורס: 236801

סמסטר: אביב תשע״ט

מרצה:	שחר יצחקי
שעות הרצאה:	
שעת תרגול:	אין
דרישות קדם:	לוגיקה
אתר הקורס:	

תאור הקורס

הסמינר יסקור פיתוחים מרכזיים בתחום הניתוח האוטומטי של קוד תוכנה בשלושים השנים האחרונות, בדגש על יישומיהם בכלי פיתוח עכשוויים. הנושאים יכללו: הרצה סימבולית (execution symbolic), checking model Bounded, אימות בעזרת Theories Modulo SAT, בדיקות בזמן ריצה, בדיקות אוטומטיות, ניתוח קוד multithreaded, ניתוח קוד GPU.
שימושים של טכניקות אלה משפרות יכולות debugging ואימות אוטומטי של תכניות מחשב, במטרה להוביל למערכות מחשב יציבות, אמינות, ויעילות יותר.

דרישות הקורס

נוכחות חובה
קריאה והצגה של 1-2 מאמרים אקדמיים
ניהול דיון שאלות ותשובות בנושא שהוצג
הגשת סיכום (עמוד אחד) של הנושא כפי שהוצג במאמרים ובדיון

רשימת ספרות

King, J.C. “Symbolic execution and program testing.” Commun. ACM 19, 7 (july 1976), 385–394.

Boonstoppel, P., Cadar, C. and Engler, D. “RWset: attacking path explosion in constraint-based test generation.” in Proceedings of TACAS’08, (mar–apr 2008).

clarke, l.a. a program testing system. in Proceedings of the 1976 Annual Conference, 488–491.

de Moura, L. and Bjørner, N. “Z3: An Efficient SMT Solver”

Nieuenhuis, R. and Oliveras, A. “Solving SAT and SAT Modulo Theories: From an Abstract Davis–Putnam–Logemann–Loveland Procedure to DPLL(T).”

Cadar, C., Dunbar, D. and Engler, D. “KLEE: unassisted and automatic generation of high-coverage tests for complex systems programs.” in Proceedings of OSDI’08, (dec 2008).
Cadar, C. and Engler, D. execution generated test cases: how to make systems code crash itself (invited paper). in Proceedings of SPIN’05, (aug 2005).
Cadar, C., Ganesh, V., Pawlowski, P., Dill, D. and Engler, D. “EXE: automatically generating inputs of death.” in Proceedings of CCS’06, (oct–nov 2006).
Chipounov, V., Kuznetsov, V. and Candea, G. “S2E: A Platform for In-Vivo Multi-Path Analysis of Software Systems”
Babic, D. and Hu, A. J. “Calysto: Scalable and Precise Extended Static Checking”
Li, G., Li, P., Sawaga, G, Gopalakrishnan, G., Ghosh, I. and Rajan, S.P. “GKLEE: Concolic verification and test generation for GPUs.” in Proceedings of PPoPP’12.
Hastings, R. and Joyce, B. “Purify: fast detection of memory leaks and access errors.” in Proceedings of Winter USENIX Conference, 1992.
Nethercote, N. and Seward, J. “Valgrind: a program supervision framework.” Electronic Notes in Theoretical Computer Science 89, 2 (2003).
Clarke, E., Biere, A., Raimi, R. and Zhu, Y. “Bounded Model Checking Using Satisfiability Solving.”
Kuznetsov, V., Kinder, J., Bucur, S. and Candea, G. “Efficient state merging in symbolic execution.” In Proceedings of PLDI’12, (jun 2012).
Sen, K., Necula, G., Gong, L., Choi, W. “MultiSE: Multi-Path Symbolic Execution using Value Summaries.”
Marinescu, P. D. and Cadar, C. “KATCH: High-Coverage Testing of Software Patches.”
Anand, S., Godefroid, P., Tillmann, N. “Demand-Driven Compositional Symbolic Execution.”
Trabish, D., Mattavelli, A., Rinetzky, N., Cadar, C. “Chopped Symbolic Execution.” In ICSE’18 (2018).
Bergan, T., Grossman, D. and Ceze, L. “Symbolic Execution of Multithreaded Programs from Arbitrary Program Contexts.”
Cui, H., Wu, J. Che Tsai, C. and Yang, J. “Stable deterministic multithreading through schedule memoization.” in Proceedings of OSDI’10.
Avgerinos, T., Cha, S.K., Hao, B.L.T. and Brumley, D. “AEG: automatic exploit generation.” in Proceedings of NDSS’11, (feb. 2011).
Bethea, D., Cochran, R. and Reiter, M. “Server-side verification of client behavior in online games.” in Proceedings of NDSS’10, 2010.