MILCOM 2016

SECURE COMMUNICATIONS AT THE SPEED OF CYBER

Graph Compactification for Efficient Program Comprehension and Analysis

Suresh C. Kothari
Richardson Professor
Department of Electrical and Computer Engineering

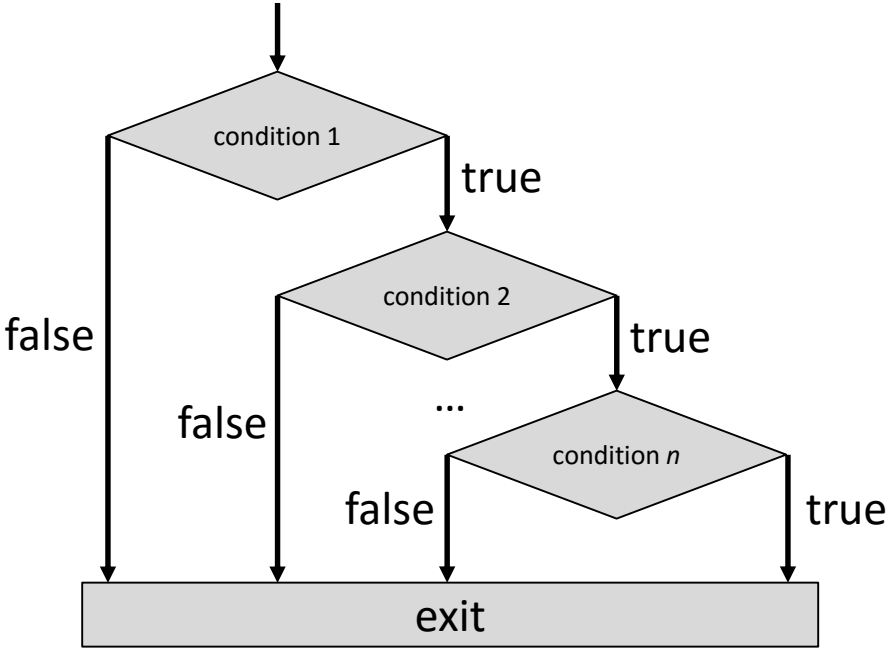
Ben Holland, Iowa State University

Acknowledgement: **Team members** at Iowa State University and EnSoft, DARPA contracts FA8750-12-2-0126 & FA8750-15-2-0080

BALTIMORE, MD • NOVEMBER 1–3, 2016

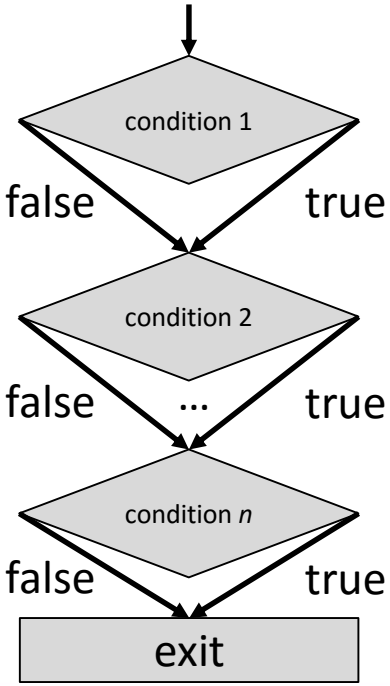
Counting Paths

- How many paths are possible for *n* nested conditions?
  - Answer: *n*+1 paths



Counting Paths

- How many paths are possible for *n* non-nested conditions?
  - Answer: 2^n paths

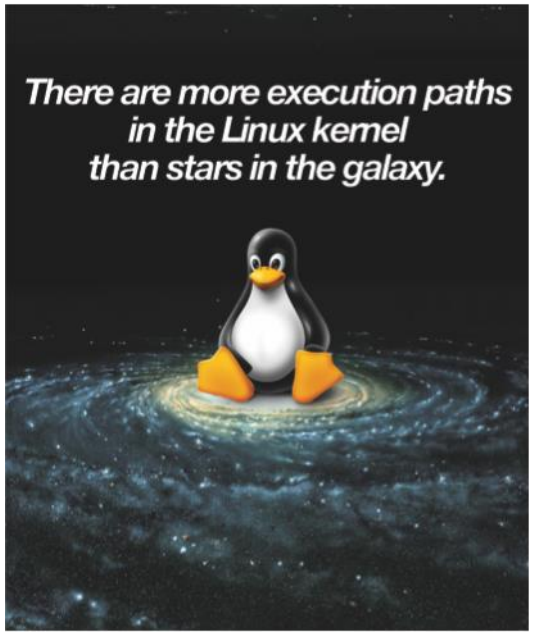


Counting Paths

- How many paths are *feasible* if c1 == c2?
  - i.e. How many paths could produce valid runtime execution traces?
  - More or less?

Counting Paths

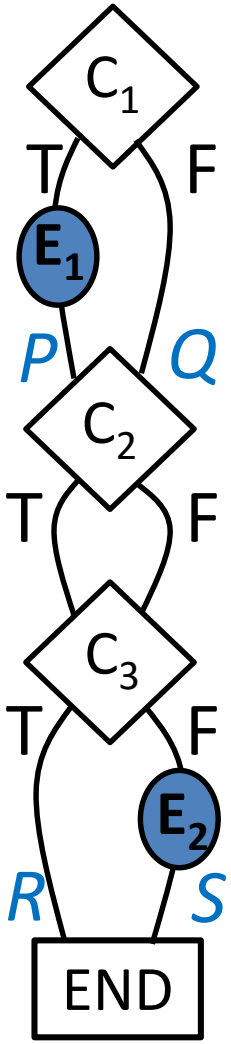
- In the worst case all conditions are non-nested and all paths are feasible.
  - Number of paths to consider in software is exponential!
  - In reality the number of feasible paths is much smaller.



Intuition: Efficient Path-Sensitive Analysis

- A large number of paths could be partitioned into a small number of groups.
- All Paths in a group are equivalent – have the same execution behavior w.r.t. the property to be verified.
- Efficient computation by examining only one path from each group.
- Challenge: How can the groups be formed without examining each path at least once?

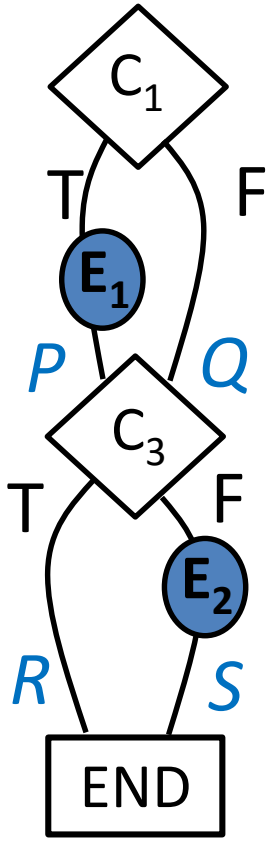
Irrelevant Branch Conditions



C_2 irrelevant to path-sensitive analysis w.r.t. E_1 and E_2

Remove the irrelevant branch conditions to avoid unnecessary path explosion & simplify the path feasibility check.

paths reduced from 8 to 4



conditions for feasibility check reduced from 3 to 2