Optimizing a Structural Constraint Solver for Efficient Software Checking

Junaid Haroon Siddiqui
The University of Texas at Austin

Darko Marinov
The University of Illinois at Urbana Champaign

Sarfraz Khurshid
The University of Texas at Austin

Introduction

- Constraint solving is an enabling technology for several static and dynamic analysis techniques.
- To analyze code that manipulates structurally complex data, the solver must support structural constraints
- Solving such constraints is expensive due to many aliasing possibilities.
- Our technique, called *focused generation*, reduces the number of test cases to be generated

Focused Generation

- Focused generation is bounded exhaustive testing where the exhaustive nature is focused on specific fields.
- Finds only one valid assignment for every focused field for a given assignment of other fields.
- Allows optimizing bounded exhaustive testing when objects to be tested contain previously tested objects.
- Focused Generation allows testing of subject classes assuming correctness of classes it depends on.
- It allows focusing tests to some aspect of a structure.
- Considering a red-black tree, we can focus test generation on ordering, coloring, or basic structure.
- The reduced state space for a binary search tree with testing focused on structure is shown below.
- As soon as a test case is found, focused generation backtracks over all not-focused fields.

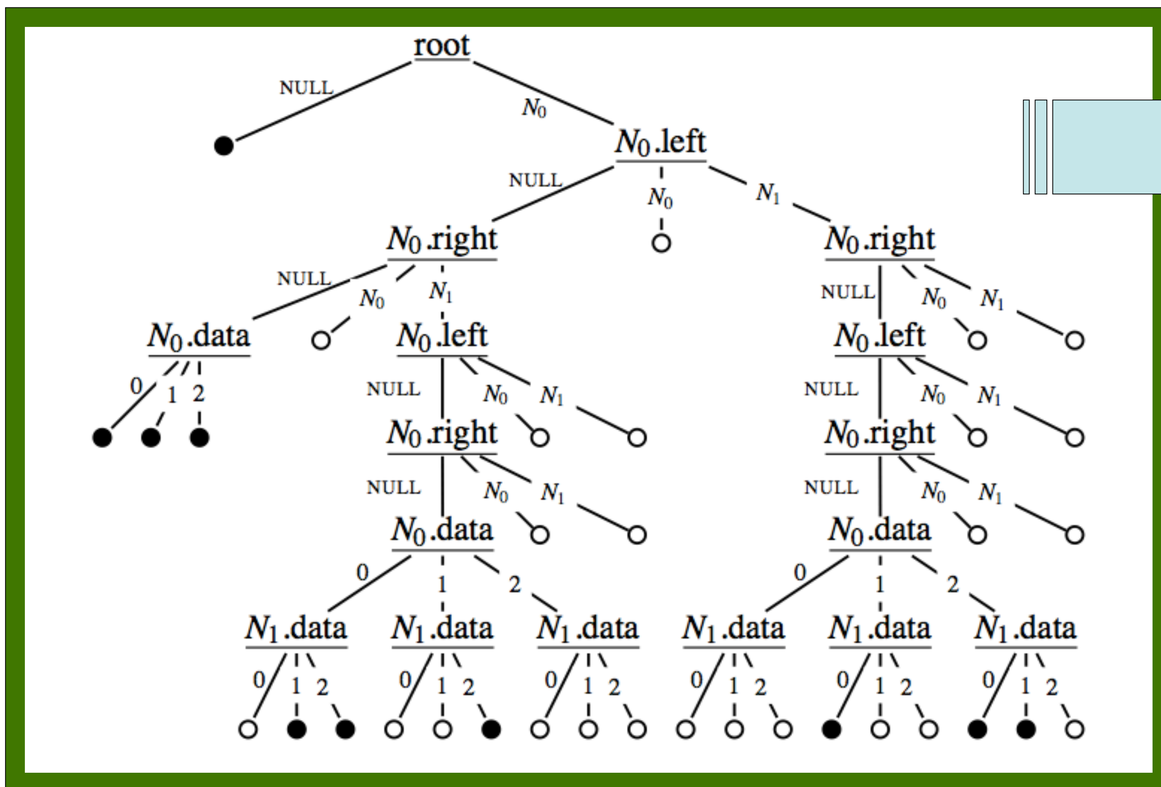
Contributions

- **Focused Generation**: A way to allow some constraints solved exhaustively while others for a single solution
- **Implementation**: We implemented focused generation on top of the Korat constraint solver.
- **Evaluation**: We compared number of tests generated and performance for Korat with and without focused generation.

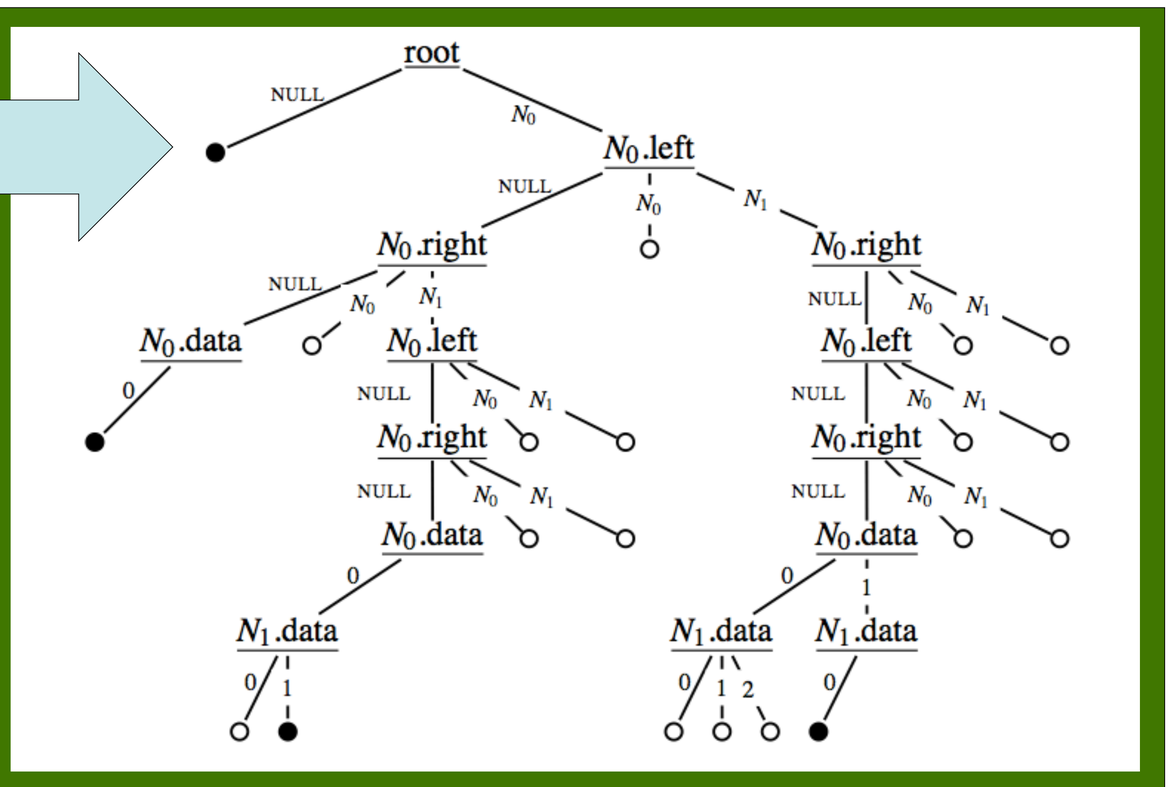
Experiment

- We compared baseline Korat and Korat with focused generation.
- With a 10 minute timeout we compared different sized sorted singly linked lists and red-black trees.
- The results show that Focused generation significantly reduces test cases generated.
- This also enables more pruning of search space and reduced test generation time.
- Test execution time decreases as a consequence of reduced number of tests.

Subject	Size	Baseline		Focused	
		Tests	Time	Tests	Time
Sorted Singly Linked List focused on structure	20	1 M	478s	20	0.00s
	25	N/A	> 600s	25	0.01s
	100	N/A	> 600s	100	2s
	200	N/A	> 600s	200	32s
Red-black tree focused on structure and coloring	10	5 M	26s	377	9s
	11	20 M	110s	707	36s
	12	85 M	478s	1395	142s
	13	N/A	> 600s	2835	563s



Binary Search Tree Exploration by Baseline Korat



Binary Search Tree Exploration with Focused Generation

- For a given structure, only one assignment of data fields is found with focused generation.