

International Conference on Automated

Software Engineering, Urbana-Champaign,

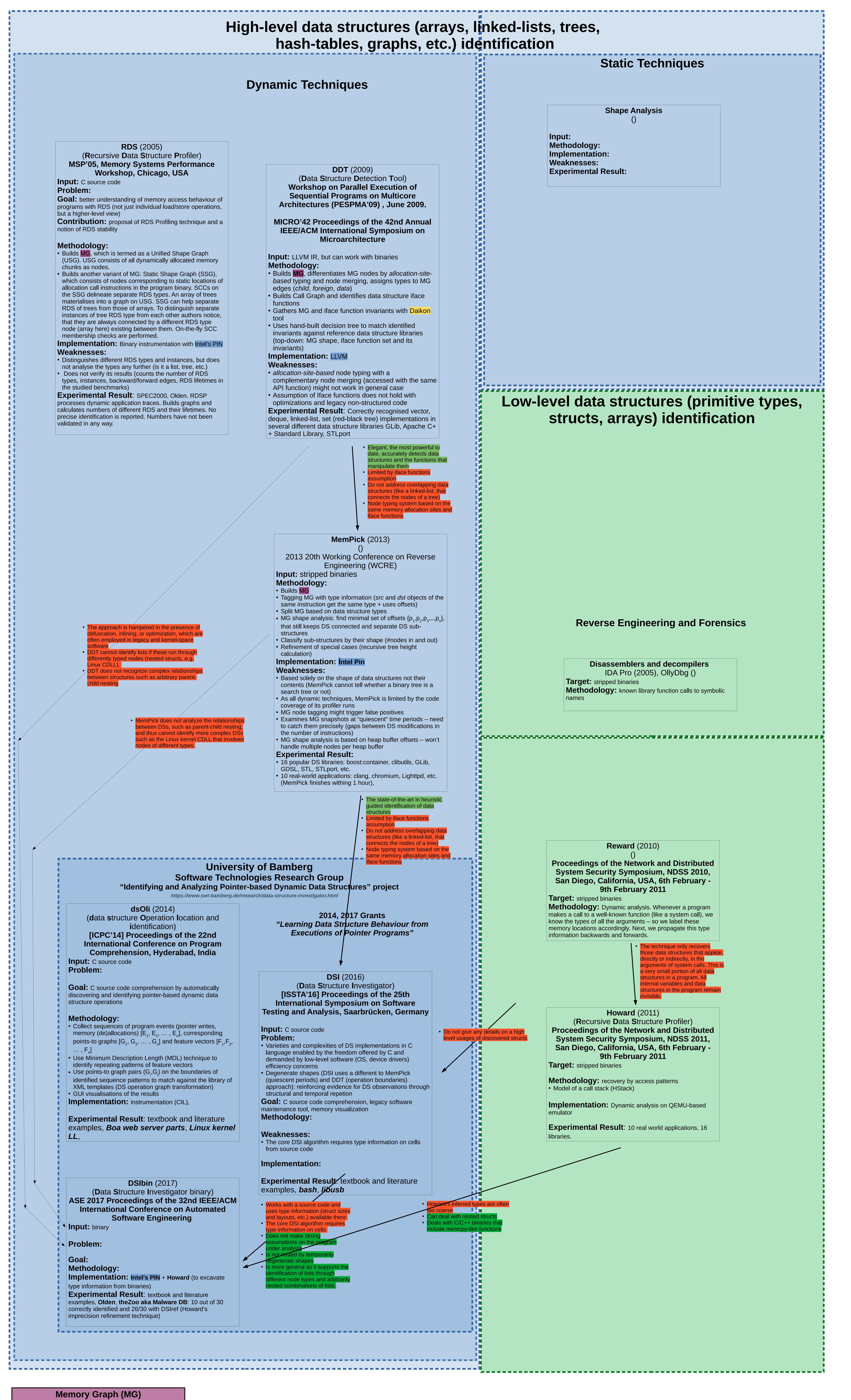
IL, USA

Methodology:

Implementation:

Experimental Result:

Nodes: add new nodes to the graph on dynamic memory allocations, remove graph nodes on deallocations Edges: store instructions (with destination and source operands being addresses pointing to locations withing previously allocated memory nodes/NULL values) create/remove MG edges **Limitations:** Composite data structures (like array of binary trees) chunk up in the single graph and are hard to distinguish, data structure overlays suffer from the same problem Multiple data structure nodes allocated in the same memory chunk are undistinguishable • Graphs evolve during a program execution, data structure manipulation operations break graph invariant properties



Nodes: add new nodes to the graph on dynamic memory allocations, remove graph nodes on deallocations

Edges: store instructions (with destination and source operands being addresses pointing to locations withing previously allocated memory

nodes/NULL values) create/remove MG edges

Limitations:

• Composite data structures (like array of binary trees) chunk up in the single graph and are

Multiple data structure nodes allocated in the same memory chunk are undistinguishable
Graphs evolve during a program execution, data structure manipulation operations break graph invariant properties

hard to distinguish, data structure overlays

suffer from the same problem