Introduction to Software Analysis

Question 1. Static vs. Dynamic Analysis

a. State one advantage of static analysis over dynamic analysis.

Answer: Static analysis may achieve soundness - it is possible to design an analysis that does not miss an error in a program, even if some of errors reported may be false positives.

Since static analysis does not require the program to be run, the cost to analyze a piece of software is proportional to its code size, not it's runtime.

Since static analysis does not require the program to be run, it can be performed on a machine without requiring that machine to be able to run the code.

b. State one advantage of dynamic analysis over static analysis.

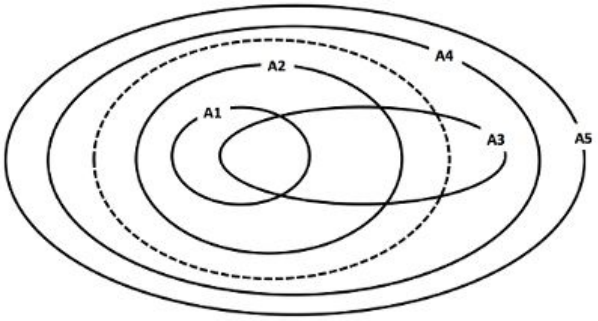
Answer: Dynamic analysis may achieve completeness - it is possible to design an analysis that will only report errors that can be triggered in a concrete execution of the program, even if some errors may be undiscovered due to limitations on the number of runs inspected.

Since dynamic analysis is performed by observing a concrete run of a program, bugs found using this method are typically easy to report / reproduce.

Note that there may be other acceptable answers to both parts of this question.

Question 2. We may classify each program as either safe or unsafe. For instance, we may say that a program is safe if it does not dereference a null pointer in any execution, and unsafe if there exists an execution in which it will dereference a null pointer.

A software analysis is considered sound if whenever it reports a program as safe, the program is indeed safe. However, due to the undecidability of software analysis, a sound analysis may reject some safe programs. A sound analysis A is more precise than a sound analysis B if whenever analysis B accepts a program, analysis A also accepts that program.



Consider the above figure. Inside of the dotted (---) oval lies the set of all safe programs, and the outside of the oval denotes the set of all unsafe programs. The inside of each solid oval, labeled A1 through A5, denotes the set of programs accepted by the corresponding analysis (e.g., these are five different null dereference checking analyses). Each analysis rejects all programs outside its corresponding oval.

Which of these five analyses are sound? List the sound analyses ordered by precision, i.e A6 > A7 > A8 indicates that A6, A7, and A8 are all sound, and A6 is more precise than A7, which is more precise than A8.

Answer: A1 and A2 are the only analyses that can be considered sound. The other analyses include unsafe programs (their oval covers regions outside the dotted oval) in the set of programs they will accept, which means the analysis is unsound.

A2 > A1, because A2 will accept any program that is safe that will also be accepted by A1, and more (A2 completely contains A1 in the diagram).