# Existing Capabilities

Infer is a static analysis tool which given a Java or C/C++/Objective-C source code is able to produce a list of potential bugs and patterns that may result in memory leaks. By design, Infer reports null pointer exceptions and resource leaks in Android and Java code.

After a successful Infer run, a user is presented with a short report regarding the bugs found. One may explore Infer’s reports in more details by running the command inferTraceBugs from the same directory the original test was running.

An Infer run is separated into two main phases: the capture phase and the analysis phase. In the capture phase, source code files are converted into Infer’s internal intermediate language. This translation process is similar to compilation, as Infer captured the compilation commands in order to perform its translation. Infer stores the intermediate files in the results directory that is reachable by the user.

In the analysis phase, the intermediate files results directory are analyzed by Infer. Infer analyzes each function and method separately. In the case that Infer encounters an error while analyzing a function or method, it will stops the analysis for that specific method or function and continue the analysis to other methods or functions. This means that after fixing reported errors, future runs of Infer may result in additional errors reported.

Infer reports errors to the standard output and also saves the results to the results directory. Infer serializes OCaml data structures that contain a control flow graph for each function or method implemented in the source code. By traversing the control flow graph Infer is able to search for bugs and memory leaks.

In Android, Infer is able to report if any Context is reachable from a static field. Thus it is able to catch static activity memory leaks in Android since Activity is a subclass of Context (Fig. #)

java.lang.Object

↳ android.content.Context

↳ android.content.ContextWrapper

↳ android.view.ContextThemeWrapper

↳ android.app.Activity

Fig. #: The Activity class hierarchy which originates from Context

Infer is also able to report inner class memory leaks in Android. Since inner class instances maintain a reference to the outer class instance, a static reference to such instances means the existence of a static field that can be used to reach the outer activity. Infer will consider it as a reachable Context and report the error.

Currently, Infer does not handle static view memory leaks not anonymous class memory leaks. This is due to how these components maintain a reference to their context in Android. While both introduce a persistent reference chain that leaks their context, Infer does no report it as a bug. In this paper, we plan to extend infer to cover such possible leaks in Andorid.

# Android Lifecycle

An activity is the single screen in Android, similar to a window or frame in Java UI. The Activity class provides a group of callback methods that allow the activity to understand that its state has changed. For example, understand that the system is creating, stopping, or resuming the activity, or destroying the entire process in which the activity resides. These states are the components that define the Android lifecycle which is controlled by seven methods: onCreate(), onStart(), onResume(), onPause(), onStop(), onRestart(), and onDestroy(). Each of these methods describes how the activity will behave at the different corresponding states. Within each lifecycle callback methods, a programmer can define how the activity works when the user or system interacts with the activity. A memory leak will occur if the method onDestroy() finishes and some reference to the destroyed Activity still exists.



Fig. #. A simplified illustration of the activity lifecycle

Source: https://developer.android.com/guide/components/activities/activity-lifecycle.html