Dear Editor and Reviewers,

We thank the reviewers for their comments. We have carefully revised the paper following these comments. The changes are highlighted in red in the revision.

The major changes include:

1. We rename the titles of Section 3.1 and Section 3.2 to provide readers with a clearer understanding of their relevance to the overall paper.

2. In Section 7.2.1, we show the details of generating the sequence of click events from the transition rule.

3. In Section 9.2.2, we use ‘’adb logcat’’ and ‘’adb shell top’’ commands to analyze the root causes of the reported crashes.

~~-------------------------------------~~--------------------------------

Below are the answers to the reviewer’s questions and comments.

**Review 2**

***Q1****. The validation of formal semantics heavily relies on auditing Android OS source code. However, Android OS has been known complex. The paper should justify how quality is ensured in the auditing process to make it more convincing.*

**A**. We do have 4 individuals auditing the source code of Android OS to ensure the reliability. We have supplied the detail in the first paragraph of Section 7.1.

***Q2****. The validation using self-designed app, i.e., ValApp, may cause questions on the reliability. The paper argues that "this way is largely credible since, after the options (e.g. launch modes and intent flags) are chosen, the rest of the validation process is the same as real-world Android apps". Does that mean ValApp only provides combination and sequence of interactive events? If so, can AMASS be extended to include an event scheduling to enable validation with real-world apps?*

**A**. AMASS can be extended to include the sequence of interactive events, but the sequence is hard to correspond to the transition rule, we show the details of generating the sequence from the transition rule in the second paragraph of Section 7.2.1.

**Review 3**

***Q1****. The addition of Chapter 3 significantly improves the clarity of the paper's motivation. However, the titles of Sections 3.1 and 3.2 appear to be somewhat arbitrary. I suggest renaming these sections to reflect the specific purpose or case studies they address, which would provide readers with a clearer understanding of their relevance to the overall paper.*

**A**. We rename the titles of these two sections.

***Q2****. Section 7.2 remains somewhat confusing regarding the selection of the 10 apps. The manuscript mentions the selection of 10 apps (line 1797) and later refers to 20 real-world apps (line 1890), but it is unclear how these groups are related. Are the 10 apps a subset of the 20, or are they entirely different sets? Additionally, while the authors explain that only 10 apps were selected because generating sequences of click events for transitions is still a manual process, this explanation is not fully convincing. The readers might struggle to understand why this process must be done manually and how labor-intensive it is, making it difficult to judge whether selecting only 10 apps is justified. I recommend that the authors reorganize this section to better clarify these points and provide more context on the manual process involved.*

**A**. We modify the first paragraph of Section 7.2 to make it clear that there are 20 real-world apps in all, and 10 apps are selected to validate the semantics of AMASS\_ACT and 10 apps are selected to validate the semantics of AMASS\_FRG. Moreover, we show the details of generating sequence of click events from the transition rule (the manual process) in the second paragraph of Section 7.2.1.

***Q3****. I still do not understand why the authors do not demonstrate the causality between the crashes and task/fragment-container unboundedness by analyzing the root causes of the reported crashes.*

**A**. We use ‘’adb logcat’’ and ‘’adb shell top’’ commands to analyze the root causes of the reported crashes in the last paragraph in Section 9.2.2, it shows that the crashes are mainly caused by the memory problems, the usage of memory will increase when creating the new activity instance or fragment instance.