Checking applications for vehicular UI constraints

- **Vehicular UI constraints:** No visual & cognitive distractions to the driver
- **Goal:** Find violations of best practice design guidelines in the application
- **Problem:** Currently requires manual testing -> Too time consuming & difficult
- **Solution:** Use model checking
  - Unassisted exploration of the application
  - Find UI violations -> Report to user via annotated screenshots

**AMC (Android Model Checker) Overview**

**Automatically identify state and its user actions**

![](image1)

**Thoroughly explore app state space**

![](image2)

**Report violations**

![](image3)

**Challenges:**
State explosion problem

**Our Solution:**
- Use heuristics to collapse similar states into one
- Structure hash & 0,1,many heuristics

**Challenges:**
Non-unique transitions

**Our Solution:**
- Explore using only unique transitions
- When necessary, use AMC’s exploration log to reach a state

**Challenges:**
False positives/negatives

**Our Solution:**
- Flag as ‘maybe’ and perform manual verification later

**Preliminary results & evaluation**

- Tested 5 non-vehicular apps -> As expected, numerous violations
- Tested 6 apps designed for vehicular use
  - **Surprisingly, also numerous violations; no better than non-vehicular apps!**
- Comparing AMC’s result with human expert
  - False positives: 4% False negatives: 4% -> Accuracy: 92%
- Average runtime: 1 hr 33 mins