

Undo/redo test generation

Undo/redo is a mainstream feature that permits user interfaces to cancel (undo) and re-execute (redo) user actions. The core idea of this project is to automatically generate tests that check the correct behaviour of standard undo/redo operations.

Let us consider the following formal definitions of the standard undo and redo operations. Let s_0 the initial state of a software system, s_n the current state of the system (with $n \geq 0$), s_{n-1} the previous state of the system ($n > 0$), s_{n-m} the m -th previous state of the system (with $n \geq m \geq 1$). We can now define the function $undo : undo(s_n) = s_{n-1}$, and the function $redo : redo(s_{n-1}) = s_n$ with $n > 0$.

These two last definitions (undo and redo functions) can be considered as two oracles for undo/redo tests, i.e. a test can compare different states of the application when applying undo/redo to check for undo/redo issues.

In this project, we will automate as far as possible the generation of undo/redo tests of an application under test. The challenges are multiple and incremental, for example :

- what is the state of an application (e.g. the graphical rendering of the widgets, the underlying data model)? How can we get it automatically?
- How to compare states (e.g. comparing screenshots of the application before and after an undo operation, comparing some variables)?
- How to generate test sequences, i.e. successive actions executed on the user interfaces (e.g. randomly, manually, using an existing test suites, using a specific tool such as GUITAR)?
- How to customize these test sequences (e.g. their length, several undo/redo operations in one test sequence)?

You will target Java Swing applications. Java applications you can test are : JabRef, LaTeXDraw, ArgoUML, Ganttproject.