

# ETR Case Generator

## High-level summary

The problem generation process is based on a randomized depth-first search for a list of View objects which will form the premises of a problem. The search incrementally chooses a random View object to append to the list, by means of randomly applying mutations to a random 'seed problem' and filtering for Views which keep the ETR-derived conclusion of the problem non-trivial. New View objects are incrementally added until the sum of the atom counts across the list is equal to one of a specified set of counts, representing a desired degree of complexity. If the greatest permitted atom count is exceeded, the last few View objects are backtracked. After too many failed attempts, the entire list is discarded and we start from scratch.

There are some other filtering steps which may be tuned. When using `generate_etr.py`, by default the problem is required to have a 'categorical' conclusion under ETR, (meaning the View amounts to a conjunction of atoms without any disjunctions). This check intervenes during the depth-first search, causing the search to continue if the condition is not met. The script `generate_etr.py` applies further logic to filter the generated problems, by default this is used to restrict to problems that have logically fallacious ETR-predicated conclusions.