## 2/12 - Forge to Boolean Logic

Garbage collected languages (context for lab):

- Languages where there is automated memory management
- There are algorithms that do that! Lab is about reference counting strategy
- Reference counting:
  - Counts # of references for each object
  - Once it's zero we can clean it up!
  - But it fails! You'll model this in lab.

What is the variables, clauses, state, etc.?

How do we get from Forge bounds + constraints to boolean logic?

- In tic-tac-toe: we have 1 Board, 3 Indices, 2 Players
- In a relation Board -> Index -> Index -> Player: how many possible rows are there?
  - 1 possible Board \* 3 poss. Indices \* 3 poss. Indices \* 2 poss. Players = 18
- The variables represent the presence or absence of these rows in the relation in the instance
  - Similarly for sets: one boolean variable that says if the Atom is present in the world (and what the Atom is)
- How do we get rid of the quantifier in all i: Index | i in A or i in B or i in C
  - We can just turn all into a bunch of and of the boolean variables since everything is bounded!
  - Similarly for every other quantifier and relational operator

Why doesn't Forge show us all the instances?

- There is symmetry breaking going on!
- Forge won't show you instances that are *isomorphic* to the ones you've already seen
- We can change how much symmetry breaking there is with option sb 20
  - Higher is more symmetry breaking

The pipeline: Forge constraints and bounds Boolean logic SAT Solver