Lecture #6: Planning as SAT and CSP

What is a compilation-based approach to solve planning problems? Why is it useful?

Why cannot we encode the planning problem as a (single) CSP?

What is the core concept of compiling planning to SAT or CSP?

What is the difference between encoding fluents and rigid atoms in SAT?

Do we need to encode rigid atoms in the SAT formula? Is there another way to take them in account? Hint: how (where) are the rigid atoms used?

Why do we need frame axioms?

Describe several approaches to encode frame axioms.

If we use the successor state axiom, do we still need effect axioms?

How do we encode that exactly one action is performed in each time step?

Describe how planning graph can be encoded in SAT.

What is a multi-valued state representation? How does it differ from classical state representation (any advantage?)?

Can multi-valued state representation be converted to classical state representation? How?

Can any CSP is be translated to an equivalent SAT problem? And vice versa?

How do CSP/SAT solvers work? Describe the core principles.