Introduction (1p)

(a) Problem

(b) Contributions:

   i. I present an implementation of automated Kantian ethics by formalizing the categorical imperative in deontic logic and implementing this formalization in Isabelle. My system improves on prior work by formalizing a sophisticated interpretation of the Formula of Universal Law and representing the idea of a maxim. This implementation contributes axioms and definitions such that a maxim, appropriately represented, can be shown to be permissible, prohibited, or obligatory, along with a list of facts used to complete the proof and, in some cases, an Isar-style readable proof.

   ii. I present a testing architecture to evaluate different implementations of the categorical imperative based on ethical intuitions and properties of Kantian ethics established in the literature. Based on my testing architecture, my implementation outperforms prior attempts at formalizing Kantian ethics. The tests include meta-ethical tests (which test the abstract properties of the ethical theory) and application tests (which test the judgements that my system would make in real-world dilemmas).

   iii. Should I include any of the philosophical contributions? Specifically, the writing on why Kantian ethics is the most natural to formalize is a philosophical contribution but not a technical one.

The Problem (1p)

(a) In order for AI agents to navigate the world responsibly, they need to perform ethical reasoning.

(b) This ethical reasoning should draw on existing work in philosophy and should be explainable.

(c) Maybe put something about computational ethics here?

My Idea (2p)

(a) Overview of the Top-Down, Theorem Prover approach

(b) Testing Framework (introduce the idea, metaethical tests, application tests)

(c) How it would fit into (a) an AI agent (b) philosophers' workflow

The Details (5p)

(a) Why Kantian Ethics (1-1.5p)

(b) Isabelle/HOL implementation (basic logical background and structure)

(c) Features: robust definition of a maxim, ability to distinguish between lies and jokes, reusability of testing framework (maybe include the table showing which tests my implementation passes vs prior implementations?)

(d) Philosophical insights discovered along the way (vacuous maxims example)

(e) Limitations (needs an input parser and a common sense database)

Related Work (1-2p)

(a) Talk about bottom-up vs top-down approaches

(b) I improve prior work by (1) staying faithful to philosophical literature (2) building an explainable system