Machine Ethics

Hands-On Session

The HERA Project

The goal of the HERA (Hybrid Ethical Reasoning Agents) project is to provide novel, theoretically well-founded and practically usable machine ethics tools for implementation in physical and virtual moral agents such as (social) robots and software bots. The research approach is to use advances in formal logic and modelling as a bridge between artificial intelligence and recent work in analytical ethics and political philosophy .

Collaborators

Martin Mose Bentzen, Technical University of Denmark

Felix Lindner, Ulm University

Software

The HERA machine ethics library is now maintained on GitHub. For more information and tutorials, we thus refer to our GitHub repository: https://github.com/existenzquantor/ethics

- Two implementations available one in Python, one in Prolog.
- Python 3+ is required, check additional information on the course page for installation details.

Tutorials

- *HERA Utility Based Causal Agency Models Tutorial
- *HERA Kantian Causal Agency Models Tutorial
- HERA Moral Planning Domain Definitions Tutorial
- HERA Explainable Ethical Reasoning Tutorial

Check the zip folder on Learn to access the tutorials.

Principles

- **Deontological Principles:** Action-Focused Deontology, Intention-Focused Deontology, Patient-focused Deontology via the second formulation of Kant's Categorical Imperative
- **Consequentialist Principles:** Utilitarian Principle, Do No Harm Principle, Do No Instrumental Harm Principle
- **Principle of Double Effect**

Patient-focused Deontology via the second formulation of Kant's Categorical Imperative

According to the second formulation of Kant's categorical imperative (also called "Humanity Formula"), an act is permissible if and only if everyone who is treated as a means by this act is also treated as a goal by the same act, that is, **nobody is treated *merely* as a means**. This principle takes individual moral patients and their benefits explicitly into account, and relates these to the agent's goals and means.

Utilitarian Principle

The *utilitarian principle* focuses on consequences of actions. It says that an agent ought to perform the action amongst the available alternatives with the **overall maximal utility**. We adopt an **act-utilitarian interpretation** which does not distinguish between doing and allowing, i.e. the causal structure of the situation is not taken into account. Thus the action which the agent ought to perform is the one which leads to the best possible situation, i.e. **the highest utility**, regardless of what the agent causes and intends.

Do No Harm Principle

The *Do No Harm principle* says that **an agent may not perform an action which has any negative consequences.** The Do No Harm principle is fulfilled in case the agent remains inactive as there will then be no negative consequences and since we regard the act token of remaining inactive itself as neutral. The distinction between doing and allowing is relevant to this principle, as it is the causal consequences of an action which are considered. The intentions of the agents are not considered ethically relevant for our interpretation of this principle.

Principle of Double Effect

The *Principle of Double Effect* says that an action is permissible if 4 conditions hold.

These are:

1. The act itself must be **morally good** or **indifferent**.
2. The positive consequence must be **intended** and the negative consequence may not be intended.
3. The negative consequence may not be a means to obtain.
4. There must be proportionally grave reasons to prefer.

Two Examples

- The dilemma of a rescue robot – Utility-based Causal Agency Models:
  actions, consequences, utilities, intentions
- Giving flowers – Kantian Causal Agency Models:
  actions, consequences, goals, affects

The Dilemma of a Rescue-Robot

A recent experiment conducted by Alan Winfield and colleagues shows that rescue robots may enter into ethical dilemmas, see [1]. In the experiment, A (for Asimov), a robot, is saving (robot stand-ins for) human beings who are about to move into a dangerous area. This the robot does by moving in front of them, which causes them small discomfort but also has the effect that they turn away from danger. However, in case of exact symmetry in terms of distance between the human beings to be saved, the robot may dither between saving one or the other and thus fail to save anyone.

[1] Alan FT Winfield, Christian Blum, and Wenguo Liu. Towards an ethical robot: internal models, consequences and ethical action selection. In M. Mistry, A. Leonardis, M.Witkowski, and C. Melhuish, editors, Advances in Autonomous Robotics Systems, pages 85–96. Springer, 2014.

Spec in YAML

rescue-robot.yaml — examples (git: master)

```yaml
description: The Rescue Robot Dilemma
actions: [a_save_h1, a_save_h2, a_remain_inactive]
background: [b_save_people]
consequences: [saved_h1, discomfort_h1, saved_h2, discomfort_h2]
mechanisms:
    saved_h1: And("b_save_people", "a_save_h1")
    discomfort_h1: a_save_h1
    saved_h2: And("b_save_people", "a_save_h2")
    discomfort_h2: a_save_h2
utilities:
    saved_h1: 10
    discomfort_h1: -4
    saved_h2: 10
    discomfort_h2: -4
    Not('saved_h1'): -10
    Not('discomfort_h1'): 4
    Not('saved_h2'): -10
    Not('discomfort_h2'): 4
intentions:
    a_save_h1: [a_save_h1, saved_h1]
    a_save_h2: [a_save_h2, saved_h2]
    a_remain_inactive: [a_remain_inactive]

```

Example in action: Jupyter Notebook

importing the required modules

The evaluation will be done with respect to two moral principles: the principle of double effect and the utilitarian principle.

In [ ]:
```
from ethics.cam.principles import DoubleEffectPrinciple, UtilitarianPrinciple, DoNoHarmPrinciple
from ethics.cam.semantics import CausalModel
```

setting up the models

In [ ]:
```
models = []
m1 = CausalModel("./examples/rescue-robot.yaml", {"a_save_h1": 1, "a_save_h2": 0, "a_remain_inactive": 0, "b_save_pe
m1.__str__= "save h1"

m2 = CausalModel("./examples/rescue-robot.yaml", {"a_save_h1": 0, "a_save_h2": 1, "a_remain_inactive": 0, "b_save_pe
m2.__str__= "save h2"

m3 = CausalModel("./examples/rescue-robot.yaml", {"a_save_h1": 0, "a_save_h2": 0, "a_remain_inactive": 1, "b_save_pe
m3.__str__= "remain inactive"

models.extend([m1,m2,m3])
```

definining models as alternatives of each other

In [ ]:
```
for m in models:
    m.alternatives = models
```

evaluation of the models

In [ ]:
```
for m in models:
    res1 = m.evaluate(DoubleEffectPrinciple)
    res2 = m.evaluate(UtilitarianPrinciple)
    res3 = m.evaluate(DoNoHarmPrinciple)
    print(m.__str__)
    print(" Principal of Double Effect: ", res1, "\n Utilitarianism: ", res2, "\n Do-No-Harm: ", res3, '\n')
```

explanation of the models

In [ ]:
```
print(m1.explain(DoubleEffectPrinciple))
```

In [ ]:
```
print(m1.explain(UtilitarianPrinciple))
```

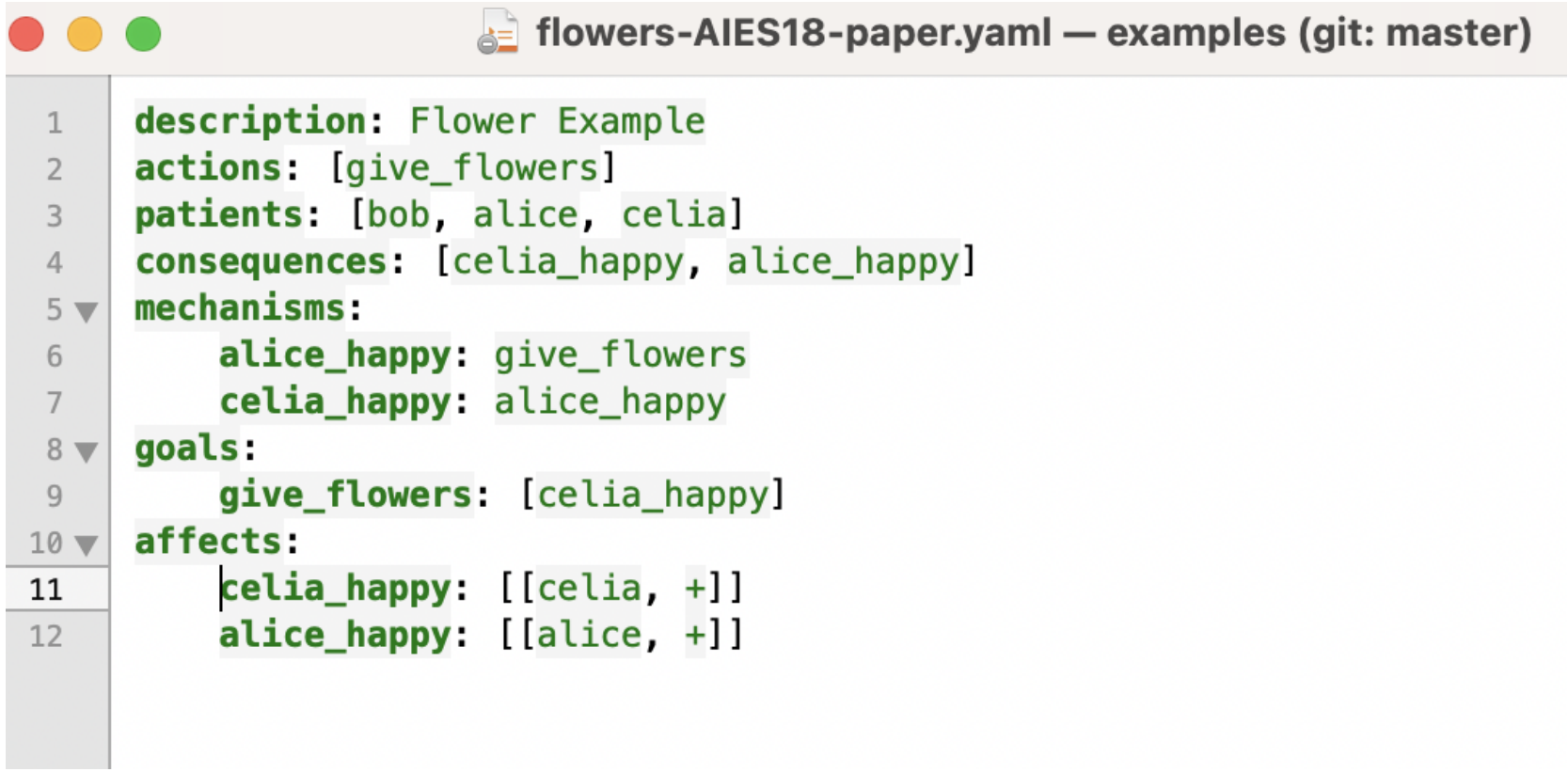
In [ ]:
```
print(m1.explain(DoNoHarmPrinciple))
```

Giving Flowers

Bob gives Alice flowers in order to make Celia happy when she sees that Alice is thrilled about the flowers. Alice being happy is not part of the goal of Bob's action.

Paper: https://www.aies-conference.com/2018/contents/papers/main/AIES_2018_paper_110.pdf

Spec in YAML


flowers-AIES18-paper.yaml — examples (git: master)
1 description: Flower Example
2 actions: [give_flowers]
3 patients: [bob, alice, celia]
4 consequences: [celia_happy, alice_happy]
5 mechanisms:
6 alice_happy: give_flowers
7 celia_happy: alice_happy
8 goals:
9 give_flowers: [celia_happy]
10 affects:
11 celia_happy: [[celia, +]]
12 alice_happy: [[alice, +]]

Example in action: Jupyter Notebook

Scenario

Bob gives Alice flowers in order to make Celia happy when she sees that Alice is thrilled about the flowers. Alice being happy is not part of the goal of Bob's action.

Paper: https://www.aies-conference.com/2018/contents/papers/main/AIES_2018_paper_110.pdf

In [ ]:
```
from ethics.cam.principles import KantianHumanityPrinciple
from ethics.cam.semantics import CausalModel
```

initalizing a model

In [ ]:
```
model = CausalModel("./examples/flowers-AIES18-paper.yaml", {"give_flowers":1})
perm = model.evaluate(KantianHumanityPrinciple)
print(perm)
```

explaining the model

In [ ]:
```
reason = model.explain(KantianHumanityPrinciple)
print(reason)
```

In [ ]:
```

```

DIY

- Check the paper: Bentzen, Martin Mose, and Felix Lindner. "A Formalization of Kant's Second Formulation of the Categorical Imperative." *ISAIM*. 2018.
- Implement Example 1 (Suicide) and Example 3 (False Promise) in the paper.

Action!