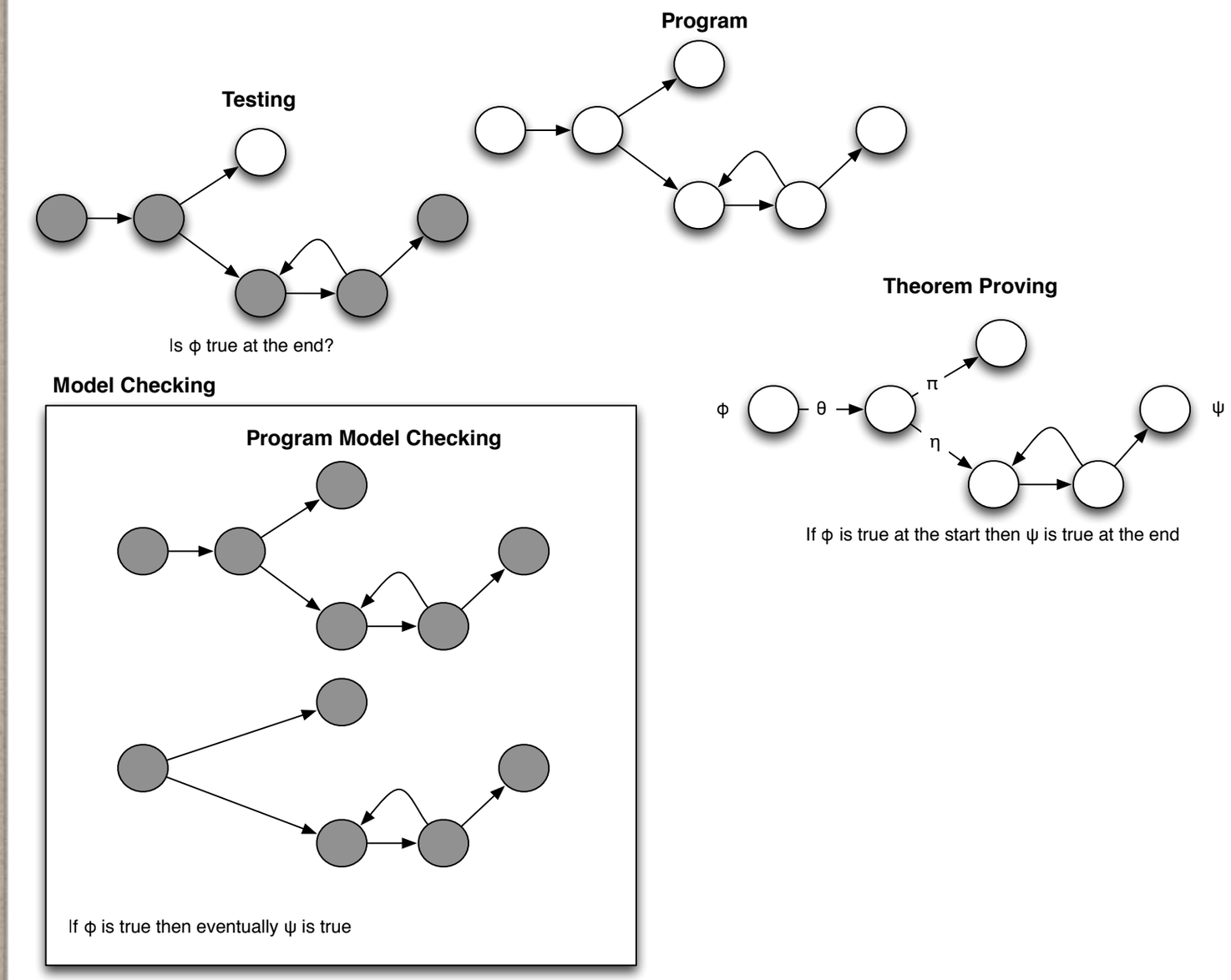
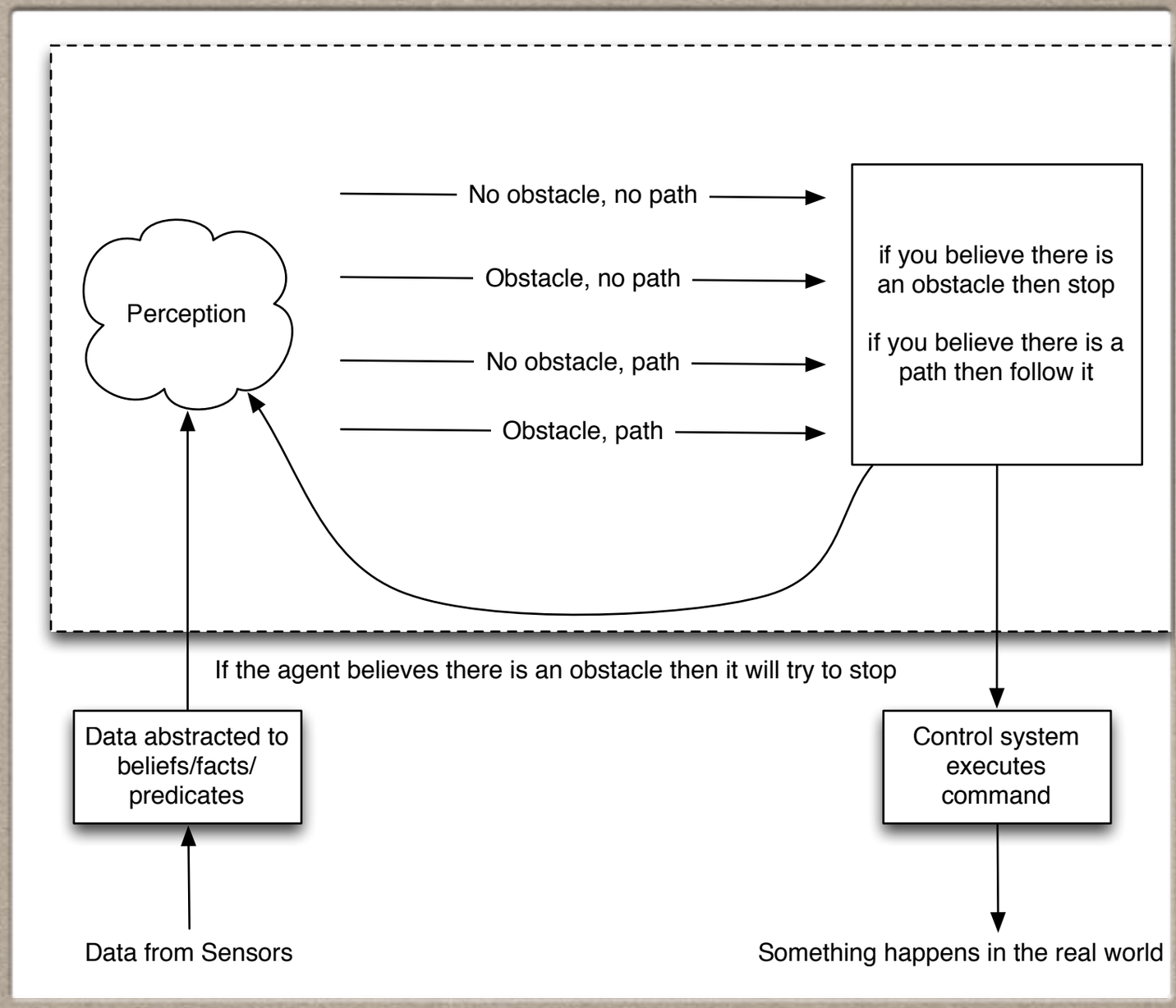
VERIFYING MACHINE ETHICS

LOUISE DENNIS

IJCAI 2018 SUMMER SCHOOL



WHAT IS VERIFICATION?



VERIFYING AGENT BASED AUTONOMOUS SYSTEMS

BRAKE FAIL ON LINE UP

ϕ_1 = do not damage own aircraft (1),

ϕ_2 = do not collide with airport hardware (2),

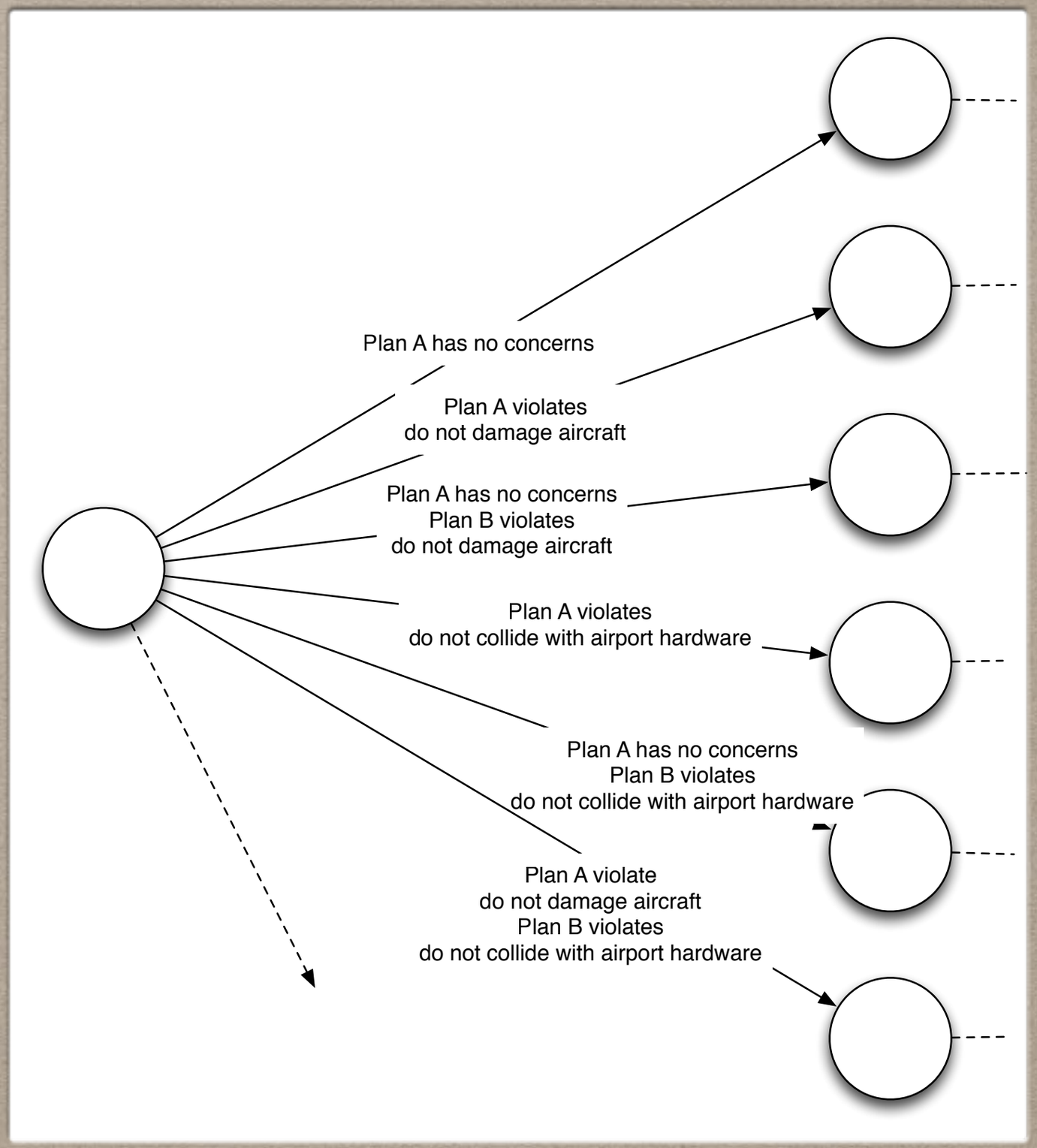
ϕ_3 = do not collide with people (3),

ϕ_4 = do not collide with manned aircraft (4).

- Turn Left (damages the aircraft and airport hardware)
- Turn Right (damages the aircraft and risks colliding with people)
- Continue (risks collision with a manned aircraft)

AIRCRAFT TURNS LEFT

VERIFYING THE PROGRAM



PROPERTIES

- If selected plan collides with a manned aircraft then all other plans collided with manned aircraft.
- If the selected plan collides with people then all other plans collided with people or manned aircraft.
- If the selected plan damages airport hardware then all other plans damaged airport hardware or collided with people or manned aircraft
- If the selected plan damages unmanned aircraft then all other plans damaged unmanned aircraft or airport hardware or collided with people or manned aircraft.

INTRUDER AIRCRAFT

ϕ_1 = do not violate turn right rule (2);

ϕ_2 = do not stay above 500 feet rule (2);

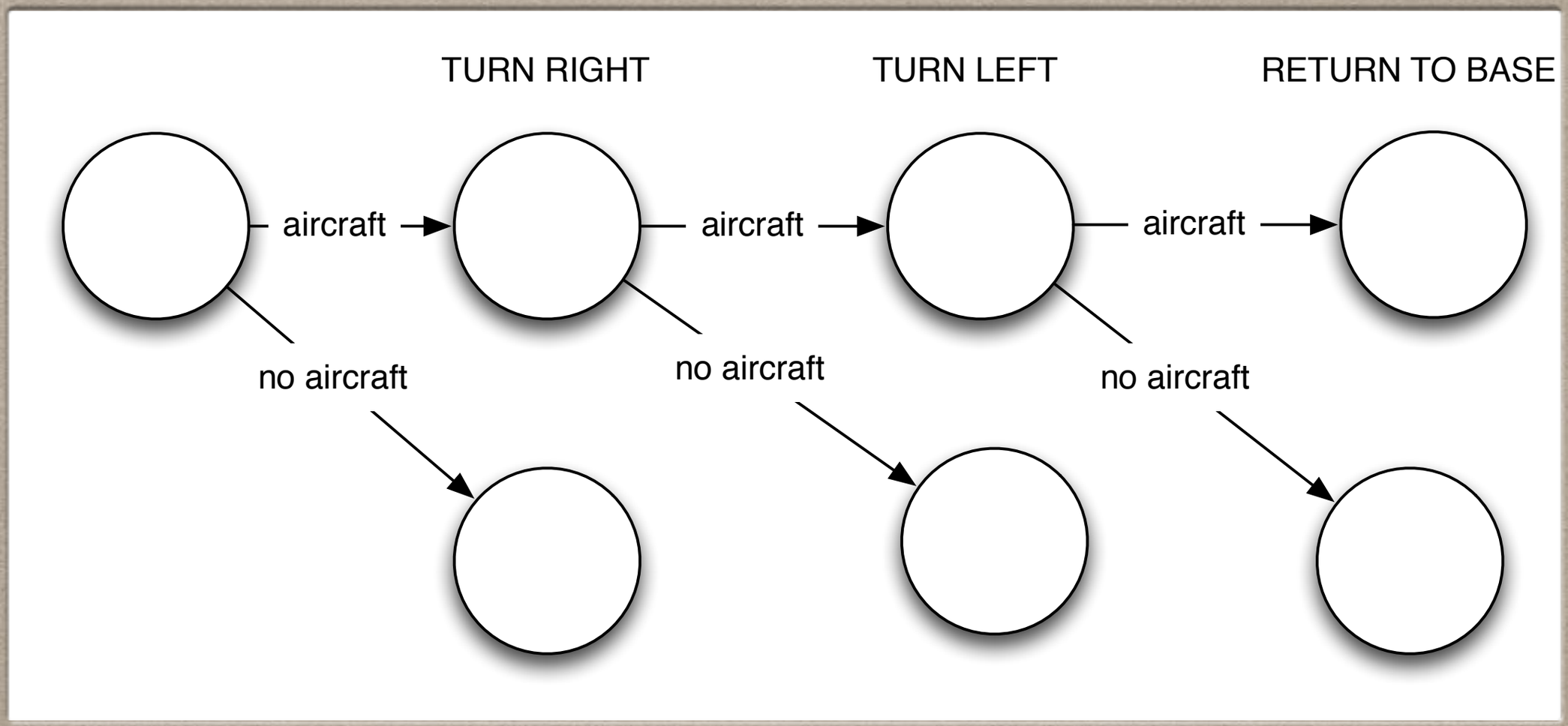
ϕ_3 = do not collide with objects on the ground (3);

ϕ_4 = do not collide with aircraft (4).

+! avoid_collision : {B flightPhase(eAvoid), ~B route(eAvoid, Route)} ←
plan(reqEmergRoute,turnRight), *route(eAvoid, R), enactRoute(R), wait;

+! avoid_collision	: {B flightPhase(eAvoid)} ←	enactRoute(turn_left); [ϕ_1]	1
+! avoid_collision	: {B flightPhase(eAvoid)} ←	enactRoute(emergency_land); [ϕ_2,ϕ_3,ϕ_4]	2
+! avoid_collision	: {B flightPhase(eAvoid)} ←	enactRoute(return_to_base); [ϕ_4]	3

VERIFYING THE PROGRAM



REMEMBER WINFIELD'S ROBOTS?

```
agent = nao_agent.NaoAgent()

add_pick_best_rule(AND(B('plans'), B('danger_close')), compare_plans_asimov_WD, update_plan_rule)
add_pick_best_rule(AND(B('plans'), NOT(B('danger_close'))), compare_plans_asimov_WT, update_plan_rule)

def compare_plans_asimov_WD(self, plan1, plan2):
        if ((plan1.robot_walking_dist < plan2.robot_walking_dist)
        and not (worse(plan1, plan2, 'robot_danger_dist'))
        and not (worse(plan1, plan2, 'robot_obj_dist'))
        and not (worse(plan1, plan2, 'human_danger_dist'))):
                return 1;
        else:
                if (worse(plan2, plan1, 'human_danger_dist')):
                        return 1;
                else:
                        if (worse(plan2, plan1, 'robot_obj_dist')
                        and not (much_worse(plan1, plan2, 'human_danger_dist'))):
                                return 1;
                        else:
                                if (worse(plan2, plan1, 'robot_danger_dist')
                                and not (worse(plan1, plan2, 'robot_obj_dist'))
                                and not (worse(plan1, plan2, 'human_danger_dist'))):
                                        return 1;
                                else:
                                        return 0;
def compare_plans_asimov_WT(self, plan1, plan2):
        if ((plan1.wait_time < plan2.wait_time)
        and not (worse(plan1, plan2, 'robot_danger_dist'))
        and not (worse(plan1, plan2, 'robot_obj_dist'))
        and not (worse(plan1, plan2, 'human_danger_dist'))):
                return 1;
```

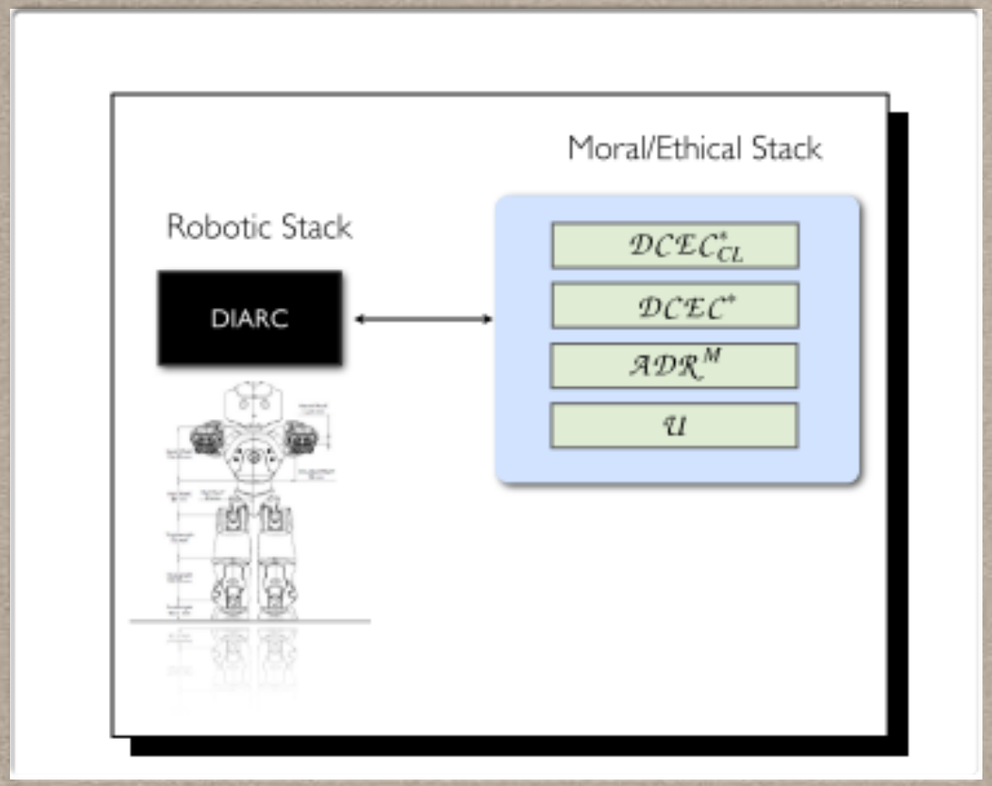
ETHICAL PYTHON CODE

PROPERTIES

- The selected plan does not put the human in more danger than the other plans.
- If the selected plan puts the robot further from its (human ordered) goal then the other plans put the human in more danger.
- If the selected plan puts the robot in danger then the other plans put the human in danger or placed the robot further from its goal.
- Eventually the agent selects a plan. (FALSE)

CONSTRAINED ENVIRONMENT

- Plan comparisons (worse) and walking time/ distance relations are transitive.
- compare_plans_asimov_WD etc are antisymmetric and transitive.



REASONING WITH EMBEDDED THEOREM PROVING/ MODEL CHECKING

REFERENCES

- Dennis et al (2016). Practical Verification of Decision-making in Agent-based Autonomous Systems. Automated Software Engineering, 23/3, pp 305-359.
- Dennis et al (2016). Formal Verification of Ethical Choices in Autonomous Systems. Robotics and Autonomous Systems, 77, 1-14.
- Dennis et al (2015). Towards Verifiably Ethical Robot Behaviour. 1st International Conference on AI and Ethics.
- Bremner et al (?). On Proactive, Transparent and Verifiable Ethical Reasoning for Robots. Under Revision for special issue on Machine Ethics for IEEE Transactions.