The Australian National University
2600 ACT | Canberra | Australia

Australian National University

School of Computing

College of Engineering, Computing and Cybernetics (CECC)

A Hierarchical Constraint for Ethical Norms in Planning Problems

— 12 pt Honours project (S2/S1 2023)

A thesis submitted for the degree
Bachelor of Advanced Computing

By:
Harry Hart

Supervisors:
Dr. Alban Grastien
Dr. Michael Norrish

October 2023

Declaration:

I declare that this work:

- upholds the principles of academic integrity, as defined in the University Academic Misconduct Rules;
- is original, except where collaboration (for example group work) has been authorised in writing by the course convener in the class summary and/or Wattle site;
- is produced for the purposes of this assessment task and has not been submitted for assessment in any other context, except where authorised in writing by the course convener;
- gives appropriate acknowledgement of the ideas, scholarship and intellectual property of others insofar as these have been used;
- in no part involves copying, cheating, collusion, fabrication, plagiarism or recycling.

October, Harry Hart

Abstract

The rise of autonomous agents' relevance in everyday life work has increasingly focused on the question of ethical restraint. A common approach is to formalise these ethical norms into formal constraints which we impose upon the solving agent. Using these constraints the agent can solve for an optimal solution which adheres to these ethics, or can check after finding a plan whether it satisfies these constraints. Unfortunately ethical frameworks are numerous and varied, and most papers focus on one particular set of ethics (utilitarianism, act-based deontology, virtue ethics) and then produce a formalisation which is very specific to that philosophical model. In this project, we create a philosophically ambivalent operator that allows us to define a structure which we believe is intrinsic to many frameworks, a hierarchy of norms. The key difference between many philosophical frameworks is *how* they rank actions within the framework, how they judge bad, good, better, and best actions. The invariant property in this is the *existence* of the hierarchy. Thus instead of focusing on formalising how to rank these actions, we focus on communicating this ranking to the solver. Using our hierarchy operator, a structure can be defined over a sequence of norms, with which we can check for satisfaction from any given plan. We provide an initial (verbose) definition in first order logic, then a more concise definition using set comprehension and sequences of norms. Then using this sequence definition we can prove some interesting properties of hierarchies such as concatenation, removing duplicates, maintenance of prefix structure, and a partial order on the space of norms. We also provide an example showing how it correctly constrains a sample plan.

Table of Contents

Chapter 1

Introduction

Autonomous machines are increasingly coming into closer contact with people in their day to day lives. Food delivery robots and driverless taxis are starting to populate the streets of Los Angeles. But with the rise in human-robot interaction, there has also been the looming question of morality. The famous Moral Machine experiment tackled the question of morality in fatal accidents with self-driving cars and through collection of responses from 233 countries found strong trends in rating humans over animals, children over older people, and executives over homeless people (Awad et al., 2018). But despite these strong indications of the need for the ability to constrain autonomous planners within ethical constraints, there does not seem to be a standardised way to formalise an ethical framework for planners.

Many papers focus on the specifics of how different moral systems rank actions or states and use this to formalise them into a constraint. We take a more ambivalent approach to formalising it. Let the philosophers define how we rank the different rules; we just communicate the ranking to the planner. Given an ethical framework, our goal is to constrain our planner to this ethical framework. Consider Asimov's Laws of Robotics (Asimov, 1942), which go as follows:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

If we had a planner which we wanted to constrain by these principles, how would we go about that? If we try to naively constrain the plan by all three of them, the hierarchy of them is not made clear. Thus in the situation where we have two plans which satisfy disjunct parts of the framework, it is not clear which is better. In this project we will

formalise a method of specifying a hierarchy of these constraints in a plan, and show how they can be composed and translated to express different properties of the frameworks within the planning space.

Chapter 2

Background

2.1 Planning Formalism

2.1.1 Planning Problems

We will use the SAS+ (Bäckström and Nebel, 1995) plan formulation for our planning problems. In this formulation the "world" is represented by state variables which describe part of the world. A state variable v can take one of certain predefined values $\mathcal{D}_v$ or an undefined value u (indicating that the variable is not important). We denote the set of predefined values and undefined as $\mathcal{D}_v^+ = \mathcal{D} \cup \{u\}$. With this set of predefined values we can define a state space:

Definition 1 (State Space Variables)**.** Given a set of state variables $\mathcal{V}$ and their respective domains $\mathcal{D}_v$, we define $\mathcal{S}_\mathcal{V} = \prod_{v \in \mathcal{V}} \mathcal{D}_v$ as the "total state space" and $\mathcal{S}_\mathcal{V}^+ = \prod_{v \in \mathcal{V}} \mathcal{D}_v^+$ as the "partial state space". ■

From this we can define a planning problem:

Definition 2 (Planning Problem)**.** A planning problem is given by a tuple $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ where:

- $\mathcal{V} = \{v_1, \ldots, v_m\}$ is a set of state variables. Each variable $v \in \mathcal{V}$ has a domain $\mathcal{D}_v$ of possible values and an extended domain $\mathcal{D}_v^+ = \mathcal{D}_v \cup \{u\}$. Note that these implicitly define the total and partial state space $\mathcal{S}_\mathcal{V}, \mathcal{S}_\mathcal{V}^+$.
- $\mathcal{O}$ is the set of operators of the form $\langle pre, post, prv \rangle$. For every $o \in \mathcal{O}$:

$$\forall v \in \mathcal{V}, pre(o)[v] \neq u \implies pre(o)[v] \neq post(o)[v] \wedge post(o)[v] \neq u \quad (R1)$$
$$\forall v \in \mathcal{V}, post(o)[v] = u \vee prv(o)[v] = u \quad (R2)$$

- $s_0 \in \mathcal{S}_\mathcal{V}^+$ and $s_* \in \mathcal{S}_\mathcal{V}^+$ denote the initial and goal states.

■

Operators are defined by three partial states *pre*, *post*, and *prv*. The *pre* conditions expresses which state variables the operator will change and what values they should have for it to apply. The *post* conditions expresses the changed variables and what values they should have after the operator is applied. The *prv* conditions expresses variables which are not changed, but are required to have some value before the operator can be applied. Thus for an operator to be applied in state $S \in \mathcal{S}_v^+$, both the *pre* and *prv* conditions must be met in S.

2.1.2 Plans

In a planning problem $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$, a plan is a sequence of operators in $\mathcal{O}$. According to Bäckström and Nebel, a general plan is defined as:

Definition 3 (General Plan)**.** Given a set of variables $\mathcal{V}$ and a set of operators $\mathcal{O}$ over $\mathcal{V}$, a general plan over $\mathcal{O}$ is a sequence $\pi = \langle o_1, \ldots, o_n \rangle$ of operators such that $o_k \in \mathcal{O}$ for all $k \in [1, n]$. ■

This does not say anything for the plan to be good, valid or even admissible. It is simply a sequence of operators. To define these further notions first we will define some functions on the states.

Definition 4 (Subsumed)**.** A state $s \in \mathcal{S}_\mathcal{V}^+$ is subsumed by another state $t \in \mathcal{S}_\mathcal{V}^+$ if:

$$\forall v \in \mathcal{V}, (s[v] = u) \vee (s[v] = t[v])$$

We denote this $s \sqsubseteq t$ ■

Definition 5 (Union)**.** For two state $s, t \in \mathcal{S}_\mathcal{V}^+$ we denote a new state $s \sqcup t$ defined as:

$$\forall v \in \mathcal{V}, (s \sqcup t)[v] = \begin{cases} s[v] & \text{if } t[v] = u \\ t[v] & \text{if } s[v] = u \end{cases}$$

■

Using these operators we can define what it means for an operator to be admissible in any particular state.

Definition 6 (Admissible Operator)**.** Let $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ be a planning problem. Given an operator $o \in \mathcal{O}$ and some partial state $s \in \mathcal{S}_\mathcal{V}^+$ we say that the operator is admissible in that state if:

$$(pre(o) \sqcup prv(o)) \sqsubseteq s$$

■

Note that the union of two states is only defined when the intersection of defined variables in the two states is empty. This works for the *pre* and *prv* sets of operators because they

are defined to be disjoint. But for more general states we further define that a state s is *updated* by a state t:

Definition 7 (Updated)**.** For two state $s, t \in \mathcal{S}_{\mathcal{V}}^{+}$ we denote a new state $s \oplus t$ defined as:

$$\forall v \in V, (s \oplus t)[v] = \begin{cases} t[v] & \text{if } t[v] \neq u \\ s[v] & \text{otherwise} \end{cases}$$

And we say that s is updated by t. ■

With this we can define what the result of a plan is:

Definition 8 (Result)**.** For a general plan $\pi = \langle o_1, \ldots, o_n \rangle$ over $\mathcal{O}$ and $\mathcal{V}$, the function *result* takes a state s and the plan π and returns:

$$\begin{aligned} \mathit{result}(s, \langle\rangle) &= s, \\ \mathit{result}(s, (\pi; o)) &= \begin{cases} \mathit{result}(s, \pi) \oplus \mathit{post}(o) & \text{if } o \text{ is admissible in } \mathit{result}(s, \pi) \\ s & \text{otherwise} \end{cases} \end{aligned}$$

■

Then finally, using this definition of the function *result* and the notation that π/k denotes $\langle o_1, \ldots, o_k \rangle$, we can define what an admissible plan is:

Definition 9 (Admissible Plan)**.** A general plan $\pi = \langle o_1, \ldots, o_n \rangle$ over $\mathcal{O}$ and $\mathcal{V}$ is admissible in a state $s \in \mathcal{S}_v^{+}$ if:

- $\pi = \langle\rangle$, or
- $\forall k, 1 \leq k \leq n$ we have that o_k is applicable in $\mathit{result}(s, \pi/(k-1))$

■

But in this project we will use the term "plan" to refer more specifically to a solution:

Definition 10 (Plan)**.** Given a planning problem $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ a plan is a general plan $\pi = \langle o_1, \ldots, o_n \rangle$ over $\mathcal{V}$ and $\mathcal{O}$ such that π is admissible in s_0 and $s_* \sqsubseteq \mathit{result}(s_0, \pi)$. ■

2.1.3 Modelling Constraints

In this project we construct complext constraints with which we restrict plans. We define a constraint as:

Definition 11 (Constraint)**.** Given a planning problem $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ a constraint is a partial state $c \in \mathcal{S}_{\mathcal{V}}^{+}$ such that $c \sqsubseteq s_*$. ■

Now the content of this project considers limiting a planner by a constraint. If we wish to constrain our planner by some rule which says "Don't injure humans", then implicitly we want that constraint to hold true for all steps in the plan. So we can formalise this by defining the "models" operator:

Definition 12 (Models)**.** Given a planning problem $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ and a constraint $c \sqsubseteq s_*$, we say a plan π models a constraint if:

$$\forall o_k \in \pi, c \sqsubseteq (post(o_k) \sqcup prv(o_k))$$

We denote it $\pi \models c$. ■

Chapter 3

Related Work

The work most closely related to ours is (Lindner et al., 2020). In this work they explore formalising what it means for a plan to be morally permissible under different ethical frameworks. They offer formalizations within planning for utilitarianism under consequentialism, act-based and goal-based deontological principles, and the *principle of double effect*. Each of these ethical frameworks in formalization fall into one of two categories: qualitative and quantitative absolute judgements. By this we mean that for each framework they either have some a priori list of "moral" actions/states and then to implement the framework they ensure that none of the actions taken are immoral, or the goal state is not immoral, etc. The second category is that there is some assumption of a utility function u which can take an action/state and give it a quantitivate "moral value". Then using this they can perform similar checks as before on the plans. The limitation of this work that we address is that it does not consider disjunct plans with competing moral satisfaction. It may be the case that there are two plans which result in different immoral actions being taken. According to a principle like their *Asimovian Principle* as long as there is no other plan which results in fewer immoral actions being taken in the goal state, then it is morally permissible. But there may in fact be different degrees of immoral actions and satisfying some are more important.

Another related work is (Grastien et al., 2021). They focus on constraining an agent to actions which are *unambiguously* permissible. They define a set of actions which are permissible, then a set of observations, and a plan is *acceptable* (unambiguously permissible) if the probability of it being an impermissible plan given the observations is less than some epsilon.

$$Pr(\text{impermissible} \mid \sigma) = \sum_{\pi' \in \Pi_i(\mathbb{P})} Pr(\pi' \mid \sigma) \leq \epsilon$$

The idea of this work is that it is not only important for an agent to be acting ethically, but also unambiguously ethical. Autonomous agents should be acting in such a way

that any observer can be reasonably confident that they are acting ethically without further investigation. This is key to establishing trust in autonomous systems and for easy explanation and assessment of plans executed by these agents. This type of ethical formulation is closest to deontology where actions are judged as intrinsically good as opposed to some form of consequentialism. This runs into some complexities with regards to some actions which may be good in some contexts, but not in others. The authors address this by noting that actions that are morally ambiguous in this way can be "differentiated with a more fine grained description"(Grastien et al., 2021). For example splitting the action "pick fruit from a tree" into "pick fruit from a tree that is yours" and "pick fruit from a tree that is not yours" which splits it into two actions which are permissible and impermissible respectively. My work will hopefully open up the possibility for more varied ethical frameworks to be defined as acceptable through the hierarchy constraint. Instead of actions being permissible or impermissible, we may define them as a gradient of morality in relation to the other choices in the problem.

A final interesting piece of work in a similar vein is (Govindarajulu et al., 2019). The authors use an ethical framework which lends itself much less to formalisation in planning, virtue ethics. Virtue ethics determines that the best action in a situation is the one that a virtuous person would perform. They formalise notions of "admiration", "traits", and "learning traits". Using this they define an n-virtuous agent to be one which is admired or considered an "exemplar" by n other agents. Then an n-virtue is a trait possessed by at least n virtuous agents. This presents an interesting approach of agents "learning" in some sense the implicit ethical framework present in other agents (or in fact humans). This presents an interesting use case in that many agents may learn some implicit "virtuous traits" from observing how a human acts in the problem, then emulate those by generalising them to virtues internally. This is not the approach we take in this work, where we explicitly lay out our ethical framework;Govindarajulu et al. compute it through observation, but it represents an ethical framework separate from consequentialism and deontology which may be difficult to formalize using this work.

Chapter 4

The Hierarchical Constraint for Ethical Norms

4.1 Motivating Example

Before we begin with the definitions, let's consider a motivating example so that we know what we are striving for. Consider the classic trolley problem, modified such that we have some autonomous agent working the lever to switch tracks. The track switching machine in this case is a planner and is given the timetable for the trains as well as information about the track status. It interprets the timetable as orders from its human masters and then tries to satisfy them exactly by creating plans for track switching as trains approach.

Now our moral problem arises when a nefarious agent has tied someone to the tracks that the trolley is heading down. If the track is not switched it will head down track A and kill the person on the tracks, yet the trolley is large enough and is going fast enough to continue down the track and arrive at Station A on time. If the planner switches the tracks to save the person the trolley will head down track B and will arrive at the wrong station throwing off the whole timetable. See Figure 4.1 for the state of the problem.

Now obviously we value human life over the timetable as the human planners, and so we want to encode this basic set of ethics into our track switching robot. We will use a well known set of rules to do this, Asimov's Laws of Robotics from his 1942 short story "Runaround" (Asimov, 1942):

(A) A robot may not injure a human being or, through inaction, allow a human being to come to harm.
(B) A robot must obey the orders given it [*sic*] by human beings except where such orders would conflict with the First Law.

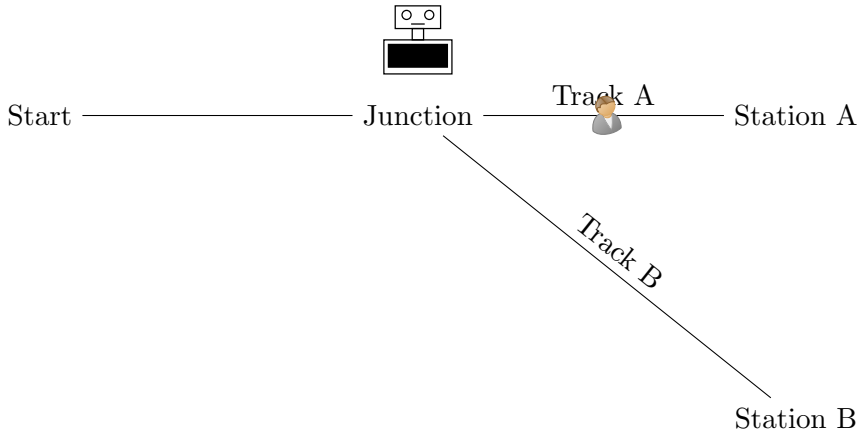

Figure 4.1: Our simplified trolley problem

(C) A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

So we want some way to specify the hierarchy (A) → (B) → (C). But this only captures part of the hierarchy determined in the rules. Some combinations of the rules are more favourable than others, e.g. $\pi \models (A) \wedge (B)$ is better than $\pi \models (A) \wedge (C)$. We can visualise the full hierarchy if we imagine that the space of plans which model each norm as a coloured region, then the full hierarchy can be seen in Figure 4.2.

The first step in defining this hierarchy is capturing the idea of doing a constraint to the best of its ability. Consider we have a norm for a car driving on the road "Don't break any road rules". This is certainly true for most driving, but in a potentially life threatening situation, we certainly would prefer that the driver breaks this norm instead of injuring someone. So if we take the world of admissible plans W which don't injure someone, we want to be able to specify "If possible given a set of plans, don't break any road rules". This will bring us to our first and most fundamental construct, try.

Definition 13 (Try)**.** For any plan $\pi \in W$ for some world W of admissible plans, given a constraint $c \in \mathcal{S}^+_{\mathcal{V}}$ we say that $\pi \models \mathsf{try}(c, W)$ if:

$$(\exists \pi' \in W, \pi' \models c) \implies \pi \models c$$

■

So this models the "if possible" principle. Another piece of notation used in further definitions is

Definition 14 (Constraint Set)**.** Given some set of plans W and some constraint c we have:

$$W|_c = \{\pi \in W \mid \pi \models c\}$$

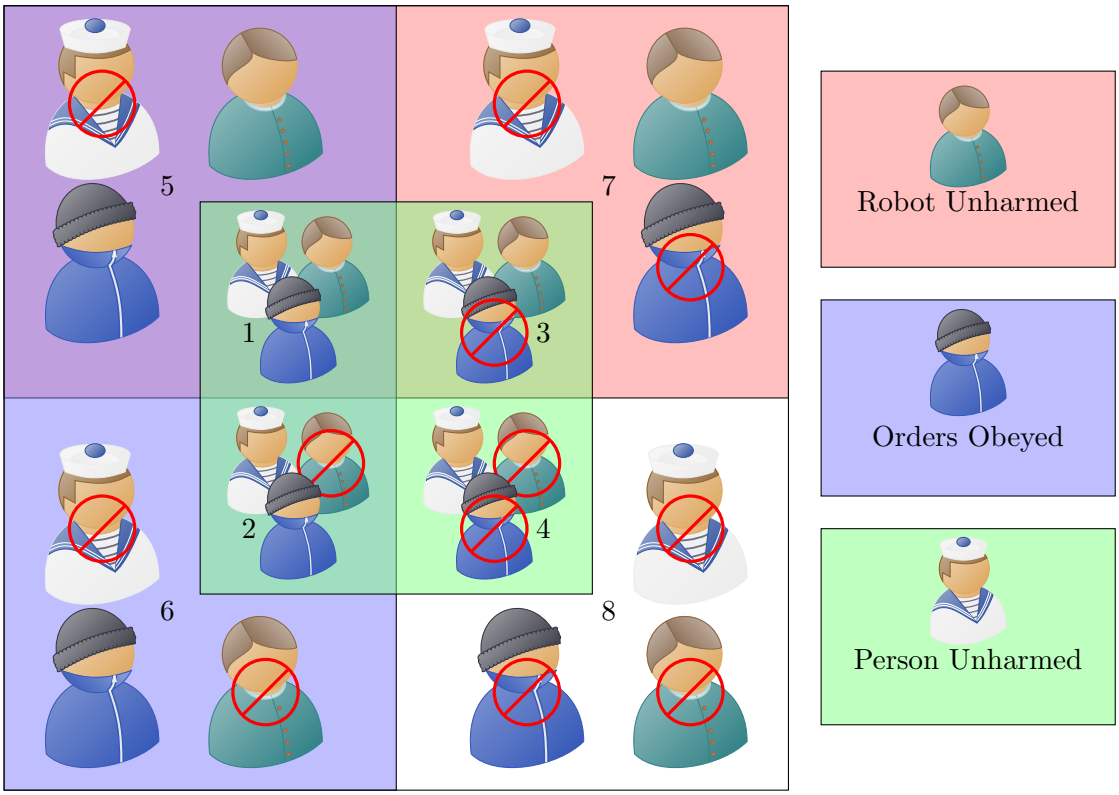

Figure 4.2: Visualising the Hierarchy of Asimov's Laws. Each region is numbered in order of preference with 1 being the most preferable plan and 8 the least preferable. It is composed of three overlapping coloured regions representing a region of plans where some condition is true (e.g. "Person Unharmed"). The state of each condition is also represented symbolically with a small icon.

■

Now we must extend it to a hierarchy.

4.2 First Order Logic Formulation

The first formulation of this hierarchy is purely built on first order logic. It is first defined in terms of two norms α and β representing constraints to create a definition for some ordering function:

$$\pi \models \lhd(\alpha, \beta, W)$$

This function defines in first-order-logic that we would prefer to achieve α over β. It is defined as follows

Definition 15 (Supercedes)**.** For some planning problem $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$, given a set of admissible plans W and a plan $\pi \in W$. For two constraints $\alpha, \beta \in \mathcal{S}_{\mathcal{V}}^{+}$ we say that π models α *supercedes* β if the following holds:

$$(\mathsf{try}(\alpha, W)) \wedge \tag{4.1}$$
$$((\exists \pi' \in W, \pi' \models (\alpha \wedge \beta)) \implies \pi \models \beta) \wedge \tag{4.2}$$
$$(((\forall \pi' \in W, \pi' \not\models \alpha) \wedge (\exists \pi' \in W, \pi' \models \beta)) \implies \pi \models \beta) \tag{4.3}$$

We denote this $\pi \models \lhd(\alpha, \beta, W)$. ■

This quite a cumbersome expression, but we can express the clauses in natural language as:

1. If possible, you must satisfy α (no matter what).
2. If it's possible to satisfy α and β, you must satisfy β.
3. If it's not possible to satisfy α but it is possible to satisfy β, you must satisfy β.

Using this we are able to specify a two constraint hierarchy. We can extend this to more stages of hierarchy, for example we can define $\pi \models \lhd(\alpha, \beta, \gamma, W)$:

$$\begin{aligned}
&(\mathsf{try}(\alpha, W)) \wedge \\
&((\exists \pi' \in W, \pi' \models (\alpha \wedge \beta)) \implies \pi \models \beta) \wedge \\
&(((\forall \pi' \in W, \pi' \not\models \alpha) \wedge (\exists \pi' \in W, \pi' \models \beta)) \implies \pi \models \beta) \wedge \\
&((\exists \pi' \in W, \pi' \models (\alpha \wedge \beta \wedge \gamma)) \implies \pi \models \gamma) \wedge \\
&(((\forall \pi' \in W, \pi' \not\models \beta) \wedge (\exists \pi' \in W, \pi' \models (\alpha \wedge \gamma))) \implies \pi \models \gamma) \wedge \\
&(((\forall \pi' \in W, \pi' \not\models \alpha) \wedge (\exists \pi' \in W, \pi' \models (\beta \wedge \gamma))) \implies \pi \models \gamma) \wedge \\
&(((\forall \pi' \in W, \pi' \not\models \alpha \wedge \pi' \not\models \beta) \wedge (\exists \pi' \in W, \pi' \models \gamma)) \implies \pi \models \gamma)
\end{aligned}$$

This method is essentially enumerating all the possible states of the world of plans, and then specifying preferences that way. Clearly this grows at roughly 2^n where n is the number of constraints in the hierarchy.

4.2.1 Limitations

The main limitation of this method is the complexity of the extension to more constraints as the hierarchy grows. There is no concise way to define the full form of the constraint and the number of terms grows exponentially. Ideally, even if the performance of the constraint is the same we would like a concise and equivalent definition. This leads us into a new approach for defining our hierarchy which is suitable for defining any number of constraints.

4.3 Sequence Definition

4.3.1 Definition

Consider this new definition for hierarchical norms in terms of a sequence. Let our norms be ordered in a sequence:

$$(a_n)_{n=1}^{N}$$

This order describes the order of importance that we put on each of the norms. Placing the most important at the start a_1 and least important at a_N. Now we can define the hierarchy of norms as such:

Definition 16 (Hierarchy)**.** For some planning problem $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$, some set of plans W, and some plan $\pi \in W$. Given an ordered sequence of constraints $A = (a_n)_{n=1}^{N}, N \geq 1$ we say that the plan π models the hierarchy A if it models the relative constraint $H : W \to \mathcal{S}_{\mathcal{V}}^{+}$ defined recursively as:

$$\pi \models H((a_i)_{i=1}^{N}, W) \iff \begin{cases} \pi \in W|_{\mathsf{try}(a_1,W)} & N = 1 \\ \pi \models H\left((a_i)_{i=2}^{N}, W|_{H((a_i)_{i=1}^{1},W)}\right) & N > 1 \end{cases}$$

■

This recursive definition allows us to use sequence proofs to give properties that we might want. Note that the set notation $W|_{H((a_n)_{n=1}^{k})}$ is the set of plans $\pi \in W$ such that $\pi \models H((a_n)_{n=1}^{k})$ is true. We can also expand this definition into a more verbose version:

$$\pi \models H((a_i)_{i=1}^{N}, W) \iff \pi \in W|_{\mathsf{try}(a_1,W)}|_{\mathsf{try}(a_2,W|_{\mathsf{try}(a_1,W)})}|\cdots|_{\mathsf{try}(a_N,W|_{\mathsf{try}(a_1,W)}|\cdots)}$$

Using this we can expand series immediately, for example $\pi \models H((a,b,c),W)$:

$$\pi \models H((a,b,c),W) = \pi \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})}|_{\mathsf{try}(c,W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})})}$$

4.3.2 Example

To show the basic ability of this hierarchy constraint we will use the example trolley problem. So that we can formally explore this, we will define our trolley problem domain. Our state variables will be:

- $p = \{\text{alive}, \text{dead}\}$
- $t = \{A, B\}$
- $tp = \{\text{start}, \text{end}\}$

These represent the state of the person on the tracks, which track the trolley is on, and whether the trolley is at the start or end of the tracks. Then our trolley has three operators available to it $\mathcal{O} = \{\text{switch}, \text{advance1}, \text{advance2}\}$ which are defined as:

$$\begin{aligned}\text{switch} &= \langle\{t = A\}, \{t = B\}, \{p = \text{alive}\}\rangle \\ \text{advanceA} &= \langle\{tp = \text{start}, p = \text{alive}\}, \{tp = \text{end}, p = \text{dead}\}, \{t = A\}\rangle \\ \text{advanceB} &= \langle\{tp = \text{start}, p = \text{alive}\}, \{tp = \text{end}, p = \text{alive}\}, \{t = B\}\rangle\end{aligned}$$

Finally the initial state is $s_0 = \{p = u, t = A, tp = \text{start}\}$ and the goal state is $s_* = \{p = u, t = u, tp = \text{end}\}$. Note that the goal is just to get from the start to the end, not specifying which track to end on or whether to save the person. Now encoding Asimov's laws into this would imply a hierarchy:

1. $p = \text{alive}$
2. $t = A$

Because the first law states not to harm a human, but the second law states to follow any human's orders (the timetable requiring the trolley to arrive at station A). Then we can encode this framework by requiring that any solution to this problem $\pi \in W$ must be constrained by:

$$\pi \models H((p = \text{alive}, t = A), W)$$

To see how this constrains us to the laws, let's consider the plans:

$$\begin{aligned}\pi_1 &= \langle\text{advanceA}\rangle \\ \pi_2 &= \langle\text{switch}, \text{advanceB}\rangle\end{aligned}$$

Now obviously according to our human understanding of the laws, we prefer π_2, but do our constraints align with that?

$$\begin{aligned}\pi_1 &\models H((p = \text{alive}, t = A), W) \\ &\models H((t = A), W|_{H((p=\text{alive}),W)}) \\ &\models H((t = A), W|_{\mathsf{try}(p=\text{alive},W)}) \\ &\implies \pi \in W|_{\mathsf{try}(p=\text{alive},W)}|_{\mathsf{try}(t=A,W|_{\mathsf{try}(p=\text{alive},W)})} \\ &\implies \pi \in \{\pi' \in W|_{\mathsf{try}(p=\text{alive},W)} | \pi' \models \mathsf{try}(t = A, W|_{\mathsf{try}(p=\text{alive},W)})\}\end{aligned}$$

Since $\pi_2 \notin W|_{\mathsf{try}(p=\text{alive},W)}$ and $W|_{\mathsf{try}(p=\text{alive},W)} \subseteq W|_{\mathsf{try}(p=\text{alive},W)}|_{\mathsf{try}(t=A,W|_{\mathsf{try}(p=\text{alive},W)})}$ then:

$$\begin{aligned}
&\implies \pi_1 \notin W|_{\mathsf{try}(p=\text{alive},W)}|_{\mathsf{try}(t=A,W|_{\mathsf{try}(p=\text{alive})})} \\
&\implies \pi_1 \not\models H((p=\text{alive}, t=A), W)
\end{aligned}$$

Without expanding any further, here we can see that π_1 is already eliminated because there *are* other plans which model $p = \text{alive}$ (namely π_2) but it does *not* satisfy this constraint.

4.3.3 Relevant Theorems

Using this definition we can prove some nice properties of these hierarchical norms.

Equivalence with Original Definition

The first and most pertinent theorem to tying this work together is whether this new definition is equivalent to the first order formulation in the 2-case. Let's prove this.

Theorem 1 (Equivalence)**.** *Let $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ be a planning problem with a set of admissible plans W. Given $a, b \in \mathcal{S}^+_{\mathcal{V}}$ constraints and $\pi \in W$ an admissible plan we have:*

$$\pi \models \lhd(a, b, W) \iff \pi \models H((a, b), W)$$

Proof. We will go from left to right:

$$\begin{aligned}
LHS &= \pi \models \lhd(a, b, W) \\
&= \pi \models \mathsf{try}(a, W) \\
&\quad \wedge ((\exists \pi' \in W, \pi' \models (a \wedge b)) \implies \pi \models b) \\
&\quad \wedge (((\forall \pi' \in W, \pi' \not\models a) \wedge (\exists \pi' \in W, \pi' \models b)) \models \pi \models b) \\
&= \pi \models \mathsf{try}(a, W) \\
&\quad \wedge ((\exists \pi' \in W|_a, \pi' \models b) \implies \pi \models b) \\
&\quad \wedge (((\forall \pi' \in W, \pi' \not\models a) \wedge (\exists \pi' \in W, \pi' \models b)) \models \pi \models b)
\end{aligned}$$

We need to reduce this to a try clause, so we will use a lemma:

Lemma 1 (Try Set Expansion)**.** *For a plan $\pi \in W$, constraint a, b the following holds:*

$$\begin{gathered}
((\exists \pi' \in W|_a, \pi' \models b) \implies \pi \models b) \wedge (((\forall \psi' \in W, \psi' \not\models a) \wedge (\exists \pi' \in W, \pi' \models b)) \implies \pi \models b) \\
\iff \\
\pi \models \mathsf{try}(b, W|_{\mathsf{try}(a,W)})
\end{gathered}$$

Proof.

$$\begin{aligned}
&\pi \models \mathsf{try}(b, W|_{\mathsf{try}(a,W)}) \\
&\quad \iff (\exists \pi' \in W|_{\mathsf{try}(a,W)}, \pi' \models b) \implies \pi \models b \\
&\quad \iff (\exists \pi' \in \{\psi \in W \mid (\exists \psi' \in W, \psi' \models a \implies \psi \models a)\}, \pi' \models b) \implies \pi \models b
\end{aligned}$$

Now we can split the set into the two cases:

$$\begin{aligned}
&\iff ((\exists \psi' \in W, \psi' \models a) \implies ((\exists \pi' \in \{\psi \in W \mid \psi \models a\}, \pi' \models b) \implies \pi \models b)) \\
&\qquad \wedge ((\forall \psi' \in W, \psi' \not\models a) \implies ((\exists \pi' \in W, \pi' \models b) \implies \pi \models b))
\end{aligned}$$

But since if $\neg(\exists \psi' \in W, \psi' \models a)$ then $(\exists \pi' \in \{\psi \in W \mid \psi \models a\}, \pi' \models b)$ is false. Thus we can remove the redundant clause.

$$\begin{aligned}
&\iff ((\exists \pi' \in \{\psi \in W \mid \psi \models a\}, \pi' \models b) \implies \pi \models b) \\
&\qquad \wedge ((\forall \psi' \in W, \psi' \not\models a) \implies ((\exists \pi' \in W, \pi' \models b) \implies \pi \models b))
\end{aligned}$$

Now we can use the fact $a \implies (b \implies c)$ is equivalent to $(a \wedge b) \implies c$.

$$\begin{aligned}
&\iff ((\exists \pi' \in \{\psi \in W \mid \psi \models a\}, \pi' \models b) \implies \pi \models b) \\
&\qquad \wedge (((\forall \psi' \in W, \psi' \not\models a) \wedge (\exists \pi' \in W, \pi' \models b)) \implies \pi \models b)
\end{aligned}$$

□

Thus from Lemma 1 we have that

$$\begin{aligned}
&= \pi \models \mathsf{try}(a, W) \\
&\qquad \wedge ((\exists \pi' \in W, \pi' \models (a \wedge b)) \implies \pi \models b) \\
&\qquad \wedge (((\forall \pi' \in W, \pi' \not\models a) \wedge (\exists \pi' \in W, \pi' \models b)) \models \pi \models b)
\end{aligned}$$

Is equivalent to:

$$= \pi \models \mathsf{try}(a, W) \wedge \pi \models \mathsf{try}(b, W|_{\mathsf{try}(a,W)})$$

Then we can reduce the second clause down to a try:

$$\begin{aligned}
&= \pi \in W|_{\mathsf{try}(a,W)} \wedge \pi \in W|_{\mathsf{try}(b, W|_{\mathsf{try}(a,W)})} \\
&= \pi \in W|_{\mathsf{try}(a,W)} \cap W|_{\mathsf{try}(b, W|_{\mathsf{try}(a,W)})} \\
&= \pi \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b, W|_{\mathsf{try}(a,W)})} \\
&= \pi \models H((a, b), W)
\end{aligned}$$

□

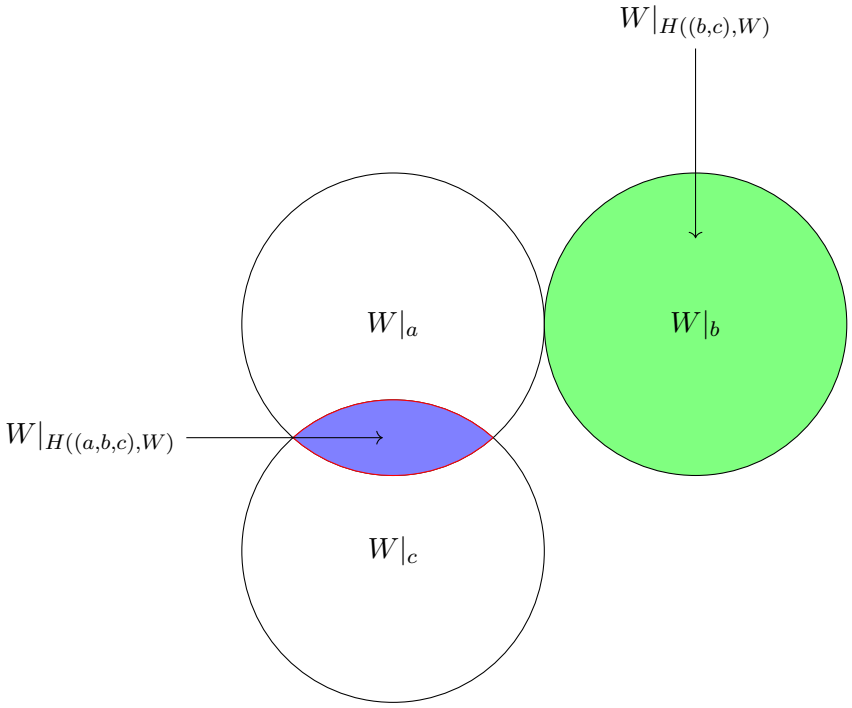

Figure 4.3: Counter-Example to the notion $\pi \models H((a,b,c),W) \implies \pi \models H((b,c),W)$.

Thus we have that the two definitions are equivalent in the most basic sense of the hierarchy of two elements. So from this we can conclude that our expansion to the sequence definition does in fact follow our original formulation reasoning. But with this sequence definition we can expand further with more properties.

Sub-hierarchy Theorems

These results relate to whether the integrity of the hierarchy is preserved when only subsequences are taken. This is important for the application to ethical frameworks, because it directly asks the question of what structure is preserved with only a subset of the norms. If $\pi \models H((a,b,c),W)$ is it true that:

1. $\pi \models H((a,b),W)$
2. $\pi \models H((b,c),W)$
3. $\pi \models H((a,c),W)$

Straight away we can invent counter-examples for (2) and (3). A counter example for (2) can be seen in Figure 4.3 and for (3) see the transitivity counter Figure 4.5.

To understand why these cases fall apart, consider the different ethical structures of:

A	**B**	**C**
1. Follow God's laws 2. Preserve your own life 3. Follow the laws of your country	1. Preserve your own life 2. Follow the laws of your country	1. Follow God's laws 2. Follow the laws of your country

In the first set of rules (**A**), you may lay down your life for your God, but this is not permissible in the second set (**B**). Then in the third set (**C**) you may lay down your life for your country's war, but in the first set (**A**) you may not if your God is ambivalent.

But we do have one structure preserved, which is the prefix for any hierarchy. Another way to think of this is concatenation with another series. For any two series $A = (a_n)_{n=1}^N$ and $B = (b_n)_{n=1}^M$ we denote the series $(a_0, \ldots, a_N, b_0, b_1, \ldots, b_M)$ obtained from concatenation as $(A; B)$.

Theorem 2 (Prefix)**.** *For a given planning problem $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ and a world of admissible plans W, let $A = (a_n)_{n=1}^N$ and $B = (b_n)_{n=1}^M$ be sequences of constraints. Then:*

$$\pi \models H((A; B), W) \implies \pi \models H(A, W)$$

Proof. Here we will prove that any plan π satisfying a hierarchy $(A; B)$ will also satisfy any arbitrary prefix of that hierarchy A. We can show this through showing a subset inclusion:

$$W|_{H((A;B),W)} \subseteq W|_{H(A,W)}$$

To start on the left hand side:

$$\begin{aligned} LHS &= W|_{H((A;B),W)} \\ &= \{\pi \in W \mid \pi \models H((A; B), W)\} \\ &= \{\pi \in W \mid \pi \models H(A, W) \wedge \pi \models H(B, W|_{H(A,W)})\} \\ &= W|_{H(A,W)}|_{H(B,W|_{H(A,W)})} \\ &\subseteq W|_{H(A,W)} \end{aligned}$$

□

At first glance you might assume that:

$$\pi \models H((A; B), W) \iff \pi \models H(A, W) \wedge \pi \models H(B, W)$$

But this is not the case because of the disjunct case. Consider Figure 4.4

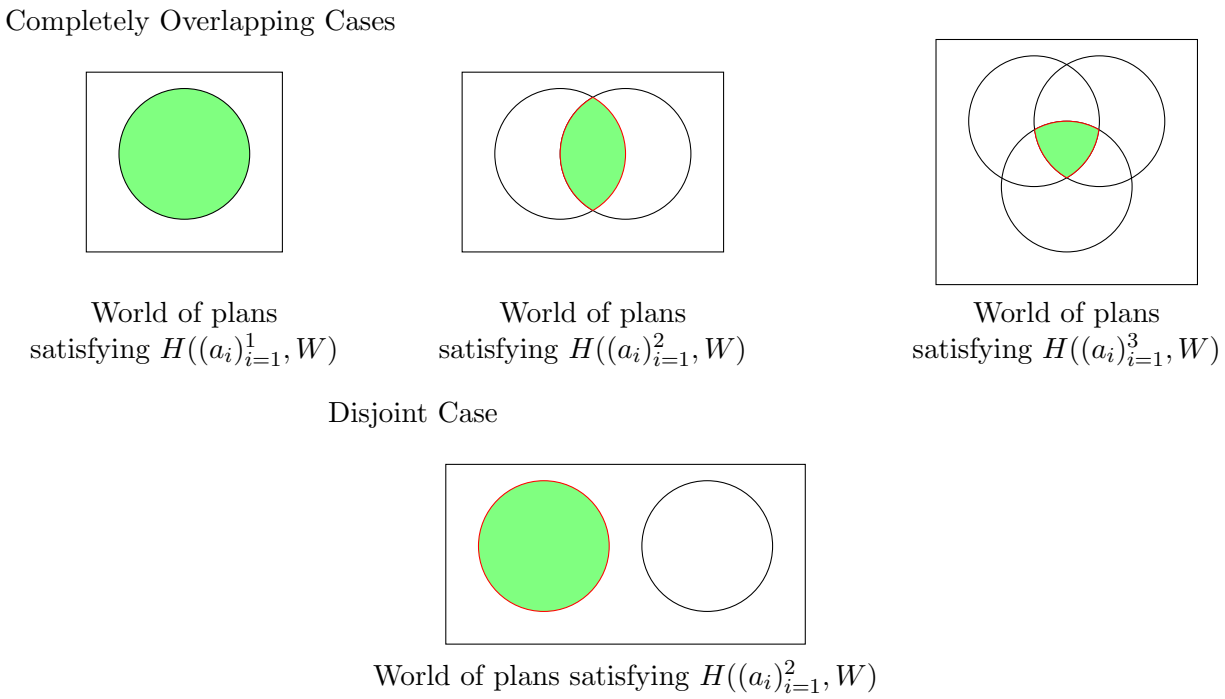

Figure 4.4: How the H function pares down the world of plans W. The green highlighted section indicates the plans satisfying the constraint. Crucially when a lower preference norm does not overlap with a higher preference norm, the "if possible" operator ensures we take the preferable set.

Theorem 3 (Subsequent Duplicates)**.** *Let $A = (a_n)_{n=1}^N$ be some sequence of norms, where $\exists i \in [1, N]$ such that $a_i = a_{i+1}$. Then we can define a sequence with the duplicate removed $A' = (a_1, \ldots, a_i, a_{i+2}, \ldots, a_N)$. These are equivalent in terms of plan satisfaction, that is for some plan $\pi \in W$ we have:*

$$\pi \models H(A, W) \iff \pi \models H(A', W)$$

Proof. This will follow from a lemma about the how the sets are pared down.

Lemma 2 (Duplicate Paring)**.** *For some set of plans W and some constraint a then:*

$$W|_{H((a),W)}|_{H((a),W|_{H((a),W)})} = W|_{H((a),W)}$$

Proof.

$$\begin{aligned}
W|_{H((a),W)}|_{H((a),W|_{H((a),W)})} &= \{\pi \in W|_{H((a),W)} \mid \pi \models H((a), W|_{H((a),W)})\} \\
&= \{\pi \in W|_{H((a),W)} \mid \pi \in W|_{\mathsf{try}(a,W|_{H((a),W)})}\} \\
&= \{\pi \in W|_{H((a),W)} \mid \pi \in \{\psi \in W \mid \psi \models \mathsf{try}(a, W|_{H((a),W)})\}\} \\
&= \{\pi \in \{\phi \in W \mid \phi \models H((a), W)\} \mid \pi \in \{\psi \in W \mid \psi \models \mathsf{try}(a, W|_{H((a),W)})\}\} \\
&= \{\pi \in W \mid \pi \models H((a), W) \wedge \pi \models \mathsf{try}(a, W|_{H((a),W)})\} \\
&= \{\pi \in W \mid \pi \models H((a), W) \wedge ((\exists \pi' \in W|_{H((a),W)}, \pi' \models a) \implies \pi \models a)\}
\end{aligned}$$

Since any $\pi \in W$ such that $\pi \models a$ will be in $W|_{H((a),W)}$ we have that $(\exists \pi' \in W|_{H((a),W)}, \pi' \models a)$ is equivalent to $(\exists \pi' \in W, \pi' \models a)$.

$$\begin{aligned}
&= \{\pi \in W \mid \pi \models H((a), W) \wedge ((\exists \pi' \in W, \pi' \models a) \implies \pi \models a)\} \\
&= \{\pi \in W \mid \pi \models H((a), W) \wedge \pi \models \mathsf{try}(a, W)\} \\
&= \{\pi \in W \mid (\pi \in \{\psi \in W \mid \psi \models \mathsf{try}(a, W)\}) \wedge \pi \models \mathsf{try}(a, W)\} \\
&= \{\pi \in W \mid \pi \models \mathsf{try}(a, W) \wedge \pi \models \mathsf{try}(a, W)\} \\
&= \{\pi \in W \mid \pi \models \mathsf{try}(a, W)\} \\
&= W|_{H((a),W)}
\end{aligned}$$

□

Now if we expand the left hand side:

$$\begin{aligned}
&\pi \models H(A, W) \\
&\quad \iff \pi \in W|_{\mathsf{try}(a_1,W)}|_{\mathsf{try}(a_2,W|_{\mathsf{try}(a_1,W)})}|\ldots|_{\mathsf{try}(a_i,W|_{\mathsf{try}(a_1,W)}|\ldots)}|_{\mathsf{try}(a_{i+1},W|_{\mathsf{try}(a_1,W)}|\ldots|_{\mathsf{try}(a_i,W|_{\mathsf{try}(a_1,W)}|\ldots)})}|\ldots \\
&\quad \iff \pi \in W|_{\mathsf{try}(a_1,W)}|_{\mathsf{try}(a_2,W|_{\mathsf{try}(a_1,W)})}|\ldots|_{\mathsf{try}(a_i,W|_{\mathsf{try}(a_1,W)}|\ldots)}|_{\mathsf{try}(a_i,W|_{\mathsf{try}(a_1,W)}|\ldots|_{\mathsf{try}(a_i,W|_{\mathsf{try}(a_1,W)}|\ldots)})}|\ldots
\end{aligned}$$

Now by Lemma 2 we can remove the duplicated element.

$$\begin{aligned}
&\iff \pi \in W|_{\mathsf{try}(a_1,W)}|_{\mathsf{try}(a_2,W|_{\mathsf{try}(a_1,W)})}|\ldots|_{\mathsf{try}(a_i,W|_{\mathsf{try}(a_1,W)}|\ldots)}|\ldots \\
&\iff \pi \models H(A', W)
\end{aligned}$$

□

Theorem 4 (Duplicates)**.** *Let $A = (a_n)_{n=1}^N$ be some sequence of norms, where $\exists i, j \in [1, N]$ such that $a_i = a_j$ and $i < j$. Then we can define a sequence with the duplicate removed $A' = (a_1, \ldots, a_i, \ldots, a_{j-1}, a_{j+1}, \ldots, a_N)$. These are equivalent in terms of plan satisfaction, that is for any plan $\pi \in W$ we have:*

$$\pi \models H(A, W) \iff \pi \models H(A', W)$$

Proof. Consider the expansion of the left hand side:

$$\pi \models H(A, W) \iff \pi \in \{\psi \in W|_{\mathsf{try}(a_1,W)}|\ldots \mid \psi \models \mathsf{try}(a_N, W|_{\mathsf{try}(a_1,W)}|\ldots)\}$$

We move terms from the subscript on the left hand side to the right hand side such that we have up to a_i on the left.

$$\begin{aligned}\pi \models H(A, W) \iff \pi \in \{\psi \in W|_{\mathsf{try}(a_1,W)}|\ldots|_{\mathsf{try}(a_{i-1},W|\ldots)} \mid\ & \psi \models \mathsf{try}(a_i, W|\ldots) \wedge \ldots \\ & \wedge \psi \models \mathsf{try}(a_j, W|_{\mathsf{try}(a_1,W)}|\ldots) \wedge \ldots \\ & \wedge \psi \models \mathsf{try}(a_N, W|_{\mathsf{try}(a_1,W)}|\ldots)\}\end{aligned}$$

Note that for each world set in the sequence $W \supseteq W|_{\mathsf{try}(a_1,W)} \supseteq W|_{\mathsf{try}(a_2,W|_{\mathsf{try}(a_1,W)})} \supseteq \ldots$. Thus the world set in $\mathsf{try}(a_i, W|_{\mathsf{try}(a_1,W)}|\ldots)$ will be a superset of $\mathsf{try}(a_j, W|_{\mathsf{try}(a_1,W)}|\ldots)$. But consider the following lemma:

Lemma 3 (Try Superset)**.** *For some constraint $a \in \mathcal{S}^+_{\mathcal{V}}$ and some sets of plans W, W' such that $W' \subseteq W$, the following is true:*

$$\pi \models \mathsf{try}(a, W) \implies \pi \models \mathsf{try}(a, W')$$

Proof. Assume for contrapositive $\pi \not\models \mathsf{try}(a, W')$, then:

$$\begin{aligned}\pi \not\models \mathsf{try}(a, W') &\implies (\exists \pi' \in W', \pi' \models a) \wedge \pi \not\models a \\ &\implies (\exists \pi' \in W, \pi' \models a) \wedge \pi \not\models a \\ &\implies \pi \not\models \mathsf{try}(a, W)\end{aligned}$$

□

Thus by Lemma 3 we have that:

$$\mathsf{try}(a_i, W|_{\mathsf{try}(a_1,W)}|\ldots) \implies \mathsf{try}(a_j, W|_{\mathsf{try}(a_1,W)}|\ldots)$$

Thus:

$$\mathsf{try}(a_i, W|_{\mathsf{try}(a_1,W)}|\ldots) \wedge \mathsf{try}(a_j, W|_{\mathsf{try}(a_1,W)}|\ldots) = \mathsf{try}(a_i, W|_{\mathsf{try}(a_1,W)}|\ldots)$$

And so we can safely remove the $\mathsf{try}(a_j, W|_{\mathsf{try}(a_1,W)}|\ldots)$ duplicate clause and logically it will be equivalent.

$$\begin{aligned}\pi \models H(A, W) \iff \pi \in \{\psi \in W|_{\mathsf{try}(a_1,W)}|\ldots|_{\mathsf{try}(a_{i-1},W|\ldots)} \mid\ & \psi \models \mathsf{try}(a_i, W|\ldots) \wedge \ldots \\ & \wedge \psi \models \mathsf{try}(a_j, W|_{\mathsf{try}(a_1,W)}|\ldots) \wedge \ldots \\ & \wedge \psi \models \mathsf{try}(a_N, W|_{\mathsf{try}(a_1,W)}|\ldots)\} \\ \pi \models H(A, W) \iff \pi \in \{\psi \in W|_{\mathsf{try}(a_1,W)}|\ldots|_{\mathsf{try}(a_{i-1},W|\ldots)} \mid\ & \psi \models \mathsf{try}(a_i, W|\ldots) \wedge \ldots \\ & \wedge \psi \models \mathsf{try}(a_N, W|_{\mathsf{try}(a_1,W)}|\ldots)\} \\ \iff \pi \models H(A', W)\end{aligned}$$

□

Partial Order Theorems

We can think of specified chains of hierarchies as defining a partial order over the space of ethical norms. To define a partial order it would have to satisfy:

- Reflexivity $\forall \pi \in W, \pi \models H((a,a),W)$
- Antisymmetry $\pi \models H((a,b),W) \land \pi \models H((b,a),W) \implies a = b$
- Transitivity $\pi \models H((a,b),W) \land \pi \models H((b,c),W) \implies \pi \models H((a,c),W)$

But these direct translations will not hold immediately. We need to adjust them for the unique structure of our ethical norm world. We will encapsulate the idea of one norm $a \in \mathcal{S}_{\mathcal{V}}^{+}$ is "equal" to another $b \in \mathcal{S}_{\mathcal{V}}^{+}$ if they are of equal importance to our plan. Meaning we say that they are equivalent in $\pi \in W$ if:

$$\pi \models \mathsf{try}(a, W) \land \mathsf{try}(b, W)$$

This encapsulates that if the two norms are possible to satisfy in this world, the plan must equally satisfy both otherwise it is unsatisfiable. So with this we translate our reflexivity and antisymmetry properties to:

- Reflexivity $\pi \models H((a,a),W) \implies \pi \models \mathsf{try}(a, W)$
- Antisymmetry $\pi \models H((a,b),W) \land \pi \models H((b,a),W) \implies \pi \models \mathsf{try}(a,W) \land \mathsf{try}(b,W)$

Now we can prove these two properties.

Theorem 5 (Reflexivity)**.** *Let $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ be a planning problem with a set of admissible plans W. Given $a \in \mathcal{S}_{\mathcal{V}}^{+}$ constraints and $\pi \in W$ an admissible plan we have:*

$$\pi \in W, \pi \models H((a,a),W) \implies \pi \models \mathsf{try}(a, W)$$

Proof.

$$\begin{aligned}
\pi \models H((a,a),W) &\implies \pi \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(a,W|_{\mathsf{try}(a,W)})} \\
&\implies \pi \in \{\psi \in W|_{\mathsf{try}(a,W)} \mid \psi \models \mathsf{try}(a, W|_{\mathsf{try}(a,W)})\} \\
&\implies \pi \in \{\psi \in W|_{\mathsf{try}(a,W)} \mid (\exists \psi' \in W|_{\mathsf{try}(a,W)}, \psi' \models a) \implies \psi \models a\} \\
&\implies \pi \in \{\psi \in W|_{\mathsf{try}(a,W)} \mid (\exists \psi' \in W|_{\mathsf{try}(a,W)}, \psi' \models a) \implies \top\} \\
&\implies \pi \in \{\psi \in W|_{\mathsf{try}(a,W)} \mid \top\} \\
&\implies \pi \in W|_{\mathsf{try}(a,W)} \\
&\implies \pi \models \mathsf{try}(a, W)
\end{aligned}$$

□

Theorem 6 (Antisymmetry)**.** *Let $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ be a planning problem with a set of admissible plans W. Given $a, b \in \mathcal{S}_{\mathcal{V}}^{+}$ constraints and $\pi \in W$ an admissible plan we have:*

$$\pi \models H((a,b),W) \land \pi \models H((b,a),W) \implies \pi \models \mathsf{try}(a,W) \land \mathsf{try}(b,W)$$

Proof.

$$\begin{aligned}
&\pi \models H((a,b),W) \land \pi \models H((b,a),W) \\
&\quad\implies \pi \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})} \land \pi \in W|_{\mathsf{try}(b,W)}|_{\mathsf{try}(a,W|_{\mathsf{try}(b,W)})} \\
&\quad\implies \pi \in \{\psi \in W \mid \psi \models \mathsf{try}(a,W) \land \psi \models \mathsf{try}(b,W|_{\mathsf{try}(a,W)}) \land \psi \models \mathsf{try}(b,W) \land \psi \models \mathsf{try}(a,W|_{\mathsf{try}(b,W)})\} \\
&\quad\implies \pi \in \{\psi \in W \mid \psi \models \mathsf{try}(a,W) \land \psi \models \mathsf{try}(b,W)\} \\
&\quad\implies \pi \models \mathsf{try}(a,W) \land \mathsf{try}(b,W)
\end{aligned}$$

□

Having proven those, we now want to prove transitivity. Standard transitivity would be:

$$\pi \models H((a,b),W) \land \pi \models H((b,c),W) \implies \pi \models H((a,c),W) \tag{4.4}$$

But this does not hold in general because there is no implicit ordering of the norms thus $H((a,c),W)$ is not aware of the b norm. A counter-example showing this can be seen in the case shown in Figure 4.5.

But we can make a slightly weaker statement along the same lines as transitivity but fully capturing the triplet relation. This is just a special case of concatenation.

Theorem 7 (Weak Transitivity/Concatenation)**.** *Let $\mathcal{P} = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ be a planning problem with a set of admissible plans W. Given $a, b, c \in \mathcal{S}_{\mathcal{V}}^{+}$ constraints and $\pi \in W$ an admissible plan we have:*

$$\pi \models H((a,b),W) \land \pi \models H((b,c),W) \implies \pi \models H((a,b,c),W)$$

Proof.

$$\begin{aligned}
&\pi \models H((a,b),W) \land \pi \models H((b,c),W) \\
&\quad\implies \pi \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})} \land \pi \in W|_{\mathsf{try}(b,W)}|_{\mathsf{try}(c,W|_{\mathsf{try}(b,W)})} \\
&\quad\implies \pi \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})} \cap W|_{\mathsf{try}(b,W)}|_{\mathsf{try}(c,W|_{\mathsf{try}(b,W)})} \\
&\quad\implies \pi \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})} \cap \{\psi \in W|_{\mathsf{try}(b,W)} \mid \psi \models \mathsf{try}(c,W|_{\mathsf{try}(b,W)})\} \\
&\quad\implies \pi \in \{\psi \in W|_{\mathsf{try}(a,W)} \mid \psi \models \mathsf{try}(b,W|_{\mathsf{try}(a,W)})\} \cap \{\psi \in W|_{\mathsf{try}(b,W)} \mid \psi \models \mathsf{try}(c,W|_{\mathsf{try}(b,W)})\} \\
&\quad\implies \pi \in \{\psi \in W \mid \psi \models \mathsf{try}(a,W) \land \psi \models \mathsf{try}(b,W|_{\mathsf{try}(a,W)})\} \\
&\qquad\quad \cap \{\psi \in W \mid \psi \models \mathsf{try}(b,W) \land \psi \models \mathsf{try}(c,W|_{\mathsf{try}(b,W)})\} \\
&\quad\implies \pi \in \{\psi \in W \mid \psi \models \mathsf{try}(a,W) \land \psi \models \mathsf{try}(b,W|_{\mathsf{try}(a,W)}) \land \psi \models \mathsf{try}(b,W) \land \psi \models \mathsf{try}(c,W|_{\mathsf{try}(b,W)})\}
\end{aligned}$$

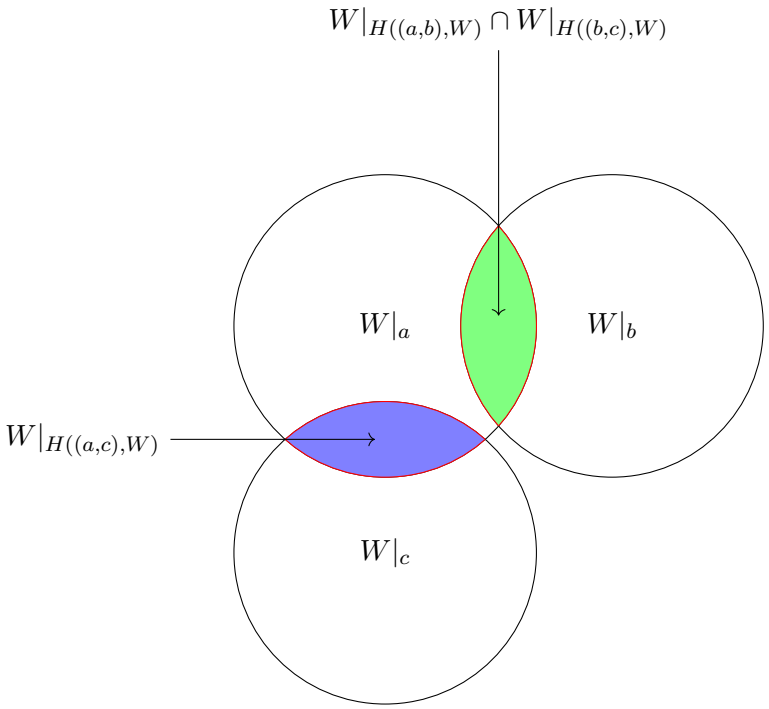

Figure 4.5: This is a counter-example to the simple transitivity statement given in Equation 4.4. The 3 regions represent the norms a, b, c. The left hand side of the transitivity statement is shown in green, note that since $W|_b$ and $W|_c$ don't overlap the region $W|_{H((b,c),W)} = W|_b$. The right hand side is shown in blue. Since these two regions do not overlap this is a direct counter-example to the simple transitivity statement.

Now we want to prove that the expression $\psi \models \mathsf{try}(a, W) \wedge \psi \models \mathsf{try}(b, W|_{\mathsf{try}(a,W)}) \wedge \psi \models \mathsf{try}(b, W) \wedge \psi \models \mathsf{try}(c, W|_{\mathsf{try}(b,W)})$ is equivalent to $(\exists \psi' \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})}, \psi' \models c) \implies \psi \models c$. We will do this as a proof by contradiction. Assume for contradiction:

$$(\exists \psi' \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})}, \psi' \models c) \wedge \psi \not\models c$$

So:

$$\begin{aligned}
&\exists \psi \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})}, \psi \models c \\
&\implies \exists \psi \in \{\psi \in W|_{\mathsf{try}(a,W)} \mid \psi \models \mathsf{try}(b, W|_{\mathsf{try}(a,W)})\}, \psi \models c \\
&\implies \exists \psi \in \{\psi \in W \mid \psi \models \mathsf{try}(a, W) \wedge \psi \models \mathsf{try}(b, W|_{\mathsf{try}(a,W)})\}, \psi \models c \\
&\implies \exists \psi \in \{\psi \in W \mid \psi \models \mathsf{try}(a, W) \wedge (\exists \psi' \in W|_{\mathsf{try}(a,W)}, \psi' \models b) \implies \psi \models b\}, \psi \models c
\end{aligned}$$

But since $\pi \models \mathsf{try}(a, W) \wedge \pi \models \mathsf{try}(b, W)$ we know that:

$$W|_a \cap W|_b \neq \emptyset \vee W|_b = \emptyset$$

Thus since we know there must be an overlap given by π, we have that this overlap must be the intersection $W|_a \cap W|_b$ or $W|_b = \emptyset$, both of these are trivially a subset of $W|_{\mathsf{try}(b,W)}$.

$$\begin{aligned}
&\exists \psi \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})}, \psi \models c \\
&\implies \exists \psi \in \{\psi \in W \mid (\exists \psi' \in W, \psi' \models b) \implies \psi \models b\}, \psi \models c \\
&\implies \exists \psi \in W|_{\mathsf{try}(b,W)}, \pi \models c \\
&\implies \psi \models c \\
&\implies \bot
\end{aligned}$$

Thus we have proven the equivalence by contradiction.

$$\begin{aligned}
&\implies \pi \in \{\psi \in W \mid (\exists \psi' \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})}, \psi' \models c) \implies \psi \models c\} \\
&\implies \pi \in W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})}|_{\mathsf{try}(c,W|_{\mathsf{try}(a,W)}|_{\mathsf{try}(b,W|_{\mathsf{try}(a,W)})})} \\
&\implies \pi \models H((a, b, c), W)
\end{aligned}$$

□

Thus we have the modified version of transitivity holding. So we have some form of a partial order upon the set of constraints. This is potentially useful as the hierarchies studied become more complex. For example consider the framework given by:

$$\pi \models H((a, b, c), W) \wedge H((a, b, d), W) \wedge H((e), W)$$

This creates a non-linear ordering we can visualise it by drawing the hierarchies as directed acyclic graphs, an example of this can be seen in Figure 4.6. Future work could explore the translation between the logical form of this constraint and the DAG's created by them.

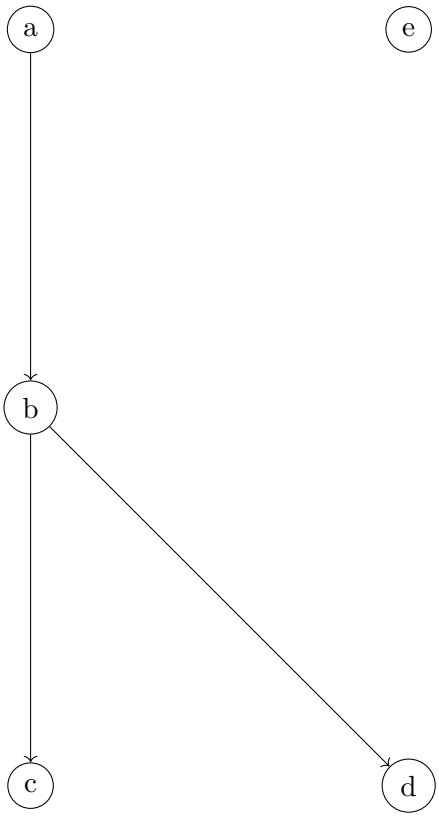

Figure 4.6: A directed acyclic graph representing the complex partial order on the norms a, b, c, d, e given by $\pi \models H((a, b, c), W) \wedge H((a, b, d), W) \wedge H((e), W)$.

Chapter 5

Concluding Remarks

5.1 Conclusion

The goal of this project is to create an operator which allows us to specify a hierarchy of norms to constrain a plan to an ethical framework. In this project we construct the operator $\lhd(a, b, W)$ which takes two norms and constructs a first order constraint for plans to maintain the hierarchy $a \succ b$. Then we construct a more powerful but still equivalent form of this operator $H(A, W)$. This takes a sequence of norms $A = (a_i)_{i=1}^N$ and then implements this as a constraint over some plan π by restricting the world of plans W to only those which satisfy the hierarchy. The first order form of the hierarchy we first came across in this project was not particularly useful in the final work but did serve as the ground work for the sequence definition later. The work in this project established the hierarchy function as useful in constructing an ethical framework as demonstrated through Asimov's Laws in the trolley problem. The body of this work proved that this hierarchy is robust to the removal of duplicates. It proved that we are able to take prefixes and concatenate sequences and maintain the framework. Finally we use the models function in conjunction with the hierarchy operator to construct a kind of partial order on the set of norms. These theorems serve to illustrate the richness of the operator.

5.2 Future Work

The full use of all of the properties proven was not included in this project and for future work on this operator these should be explored. The main work that still needs to be done is:

1. Exploring more complex hierarchies as mentioned at the end of the last section in Figure 4.6

2. Proving the Time and Space complexity of this operator
3. Implementing formally in a planner with differing ethical frameworks.

Because this operator uses many quantifiers which require checking the whole space of plans, this suggests that both the time and space complexity of using this operator may be unreasonable. Ideally future work would theoretically prove a bound on this complexity, and perhaps provide an implementation in some planning domain to demonstrate it in use. This operator offers a richness in properties and it would be rewarding to implement it using different ethical frameworks such as utilitarianism and act-based deontology to explore the full ability of it. It may at least provide an avenue of search in the pursuit of ethical machines.

Bibliography

Asimov, I., 1942. *Runaround*. Street & Smith. [Cited on pages 1 and 9.]

Awad, E.; Dsouza, S.; Kim, R.; Schulz, J.; Henrich, J.; Shariff, A.; Bonnefon, J.-F.; and Rahwan, I., 2018. The moral machine experiment. *Nature*, 563, 7729 (2018), 59–64. [Cited on page 1.]

Bäckström, C. and Nebel, B., 1995. Complexity results for planning. *Computational Intelligence*, 11 (1995), 625–656. doi:10.1111/j.1467-8640.1995.tb00052.x. https://doi.org/10.1111/j.1467-8640.1995.tb00052.x. [Cited on pages 3 and 4.]

Govindarajulu, N. S.; Bringsjord, S.; Ghosh, R.; and Sarathy, V., 2019. Toward the engineering of virtuous machines. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 29–35. [Cited on page 8.]

Grastien, A.; Benn, C.; and Thiébaux, S., 2021. Computing plans that signal normative compliance. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 509–518. [Cited on pages 7 and 8.]

Lindner, F.; Mattmüller, R.; and Nebel, B., 2020. Evaluation of the moral permissibility of action plans. *Artif. Intell.*, 287 (2020), 103350. doi:10.1016/j.artint.2020.103350. https://doi.org/10.1016/j.artint.2020.103350. [Cited on page 7.]