" Dilemma Languages & Moral Codes "

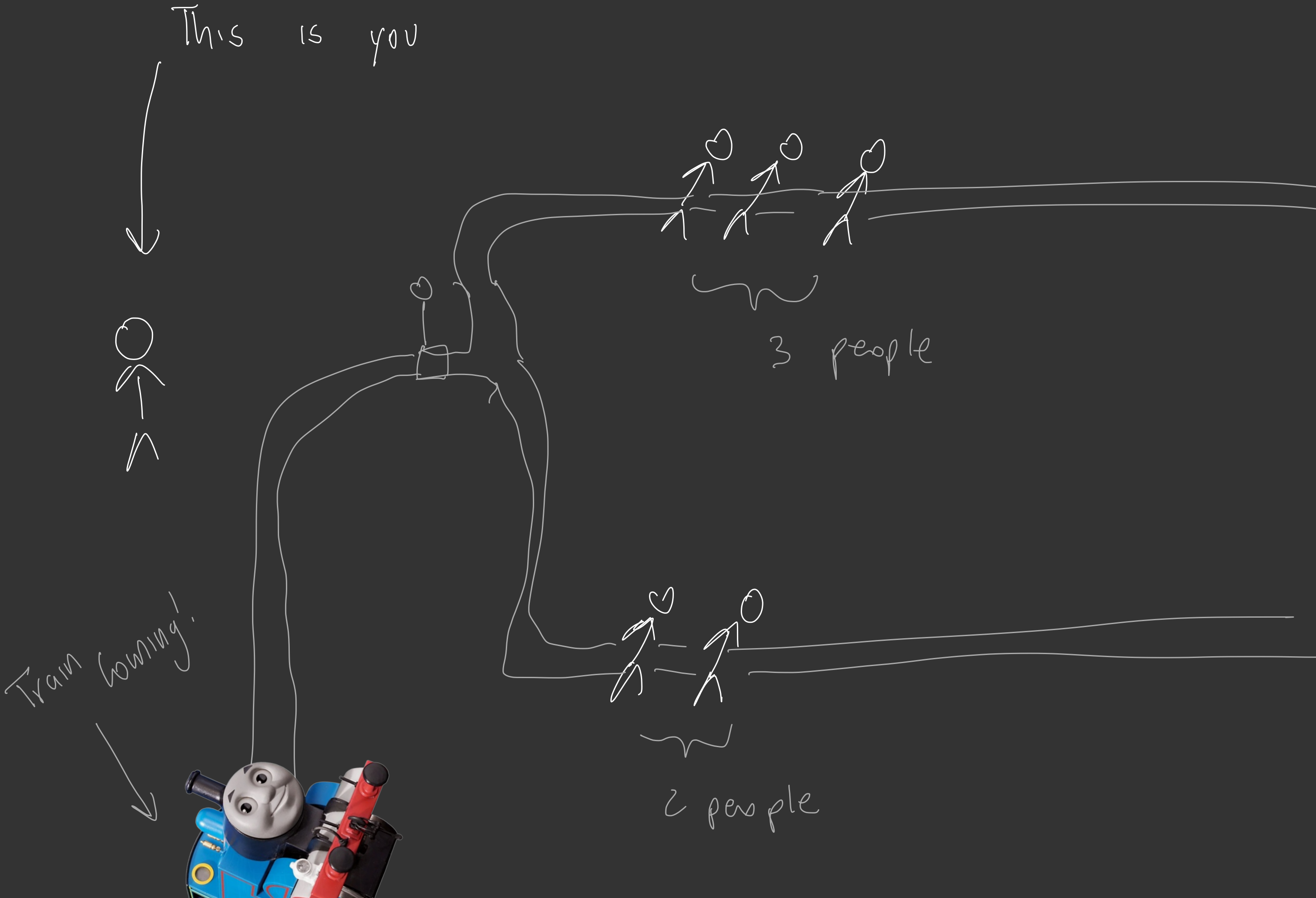
(Photo by Lucas Jackson, tasteless annotations my own)

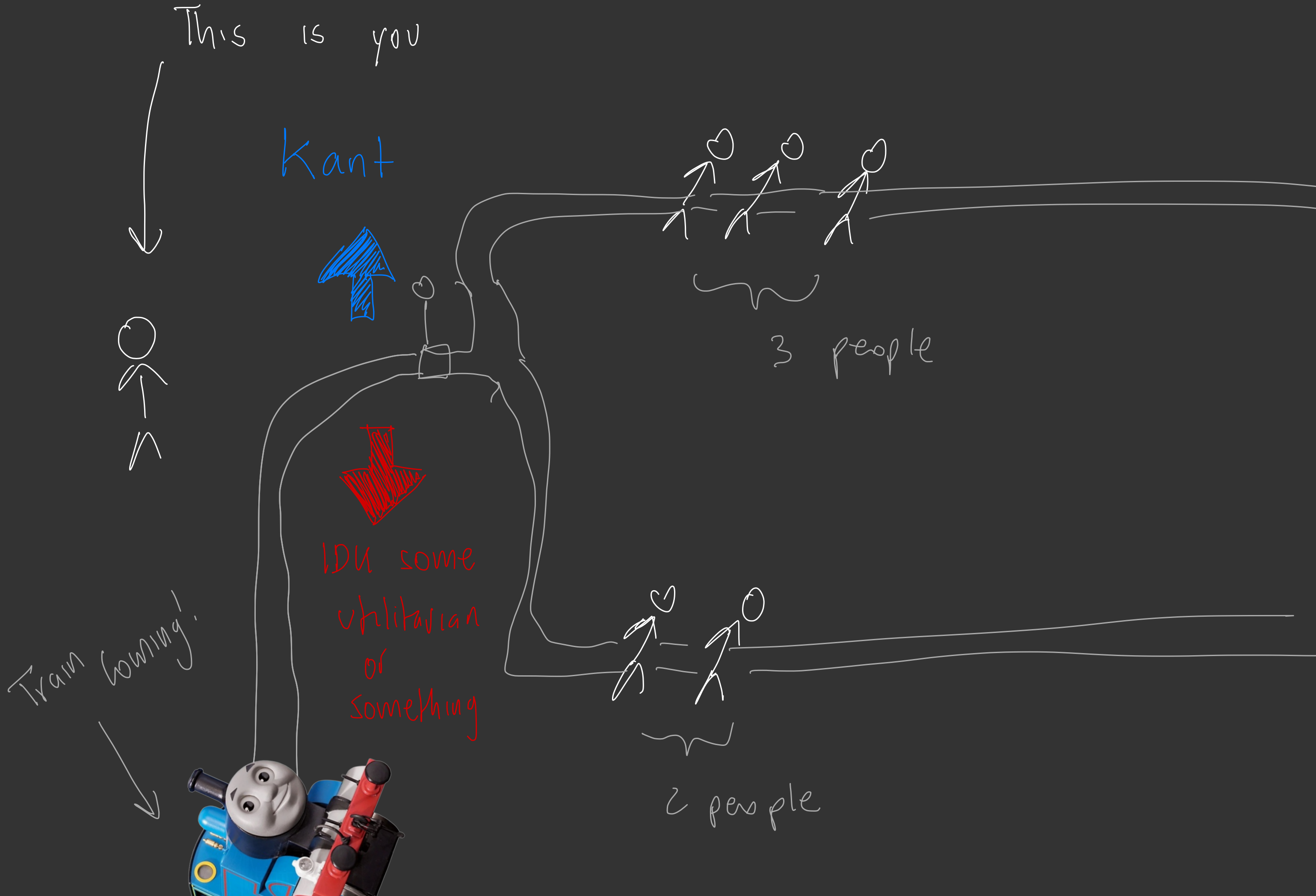
a half-baked philosophical idea by

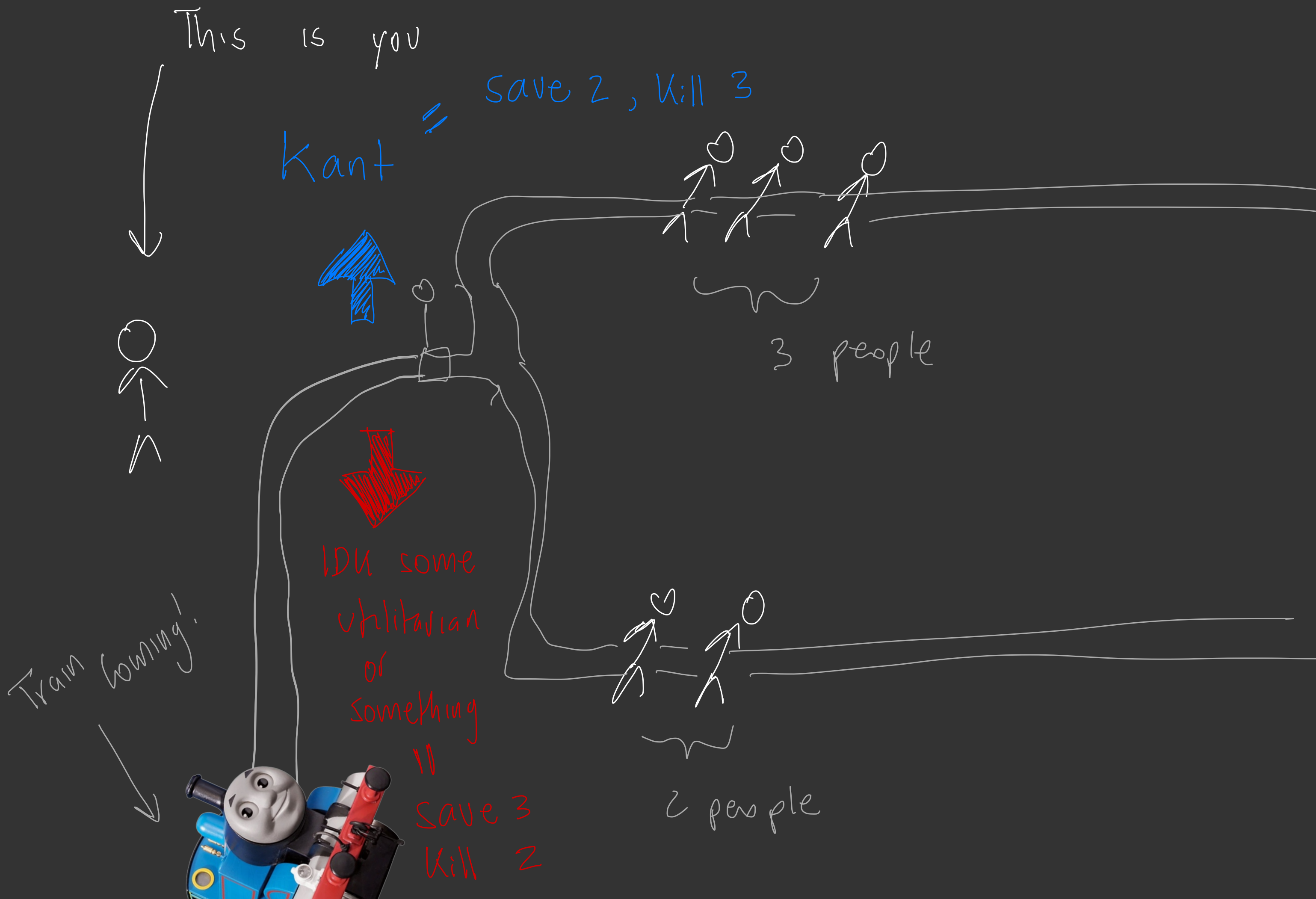
Max von Hippel

// Full of likely errors and mistakes, and not very well researched

This is you

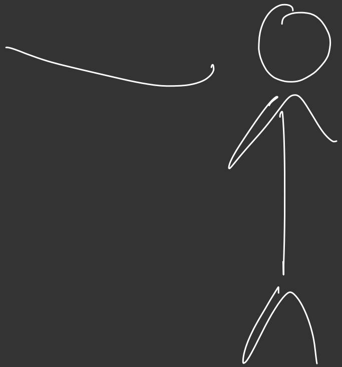

This is you
Train coming!
3 people
2 people


This is you
Kant
IDK some utilitarian or something
3 people
2 people
Train Coming!


This is you
Kant
=
save 2, kill 3
3 people
2 people
Train Coming!
IDK some utilitarian or something
=
Save 3
Kill 2

Oh no!

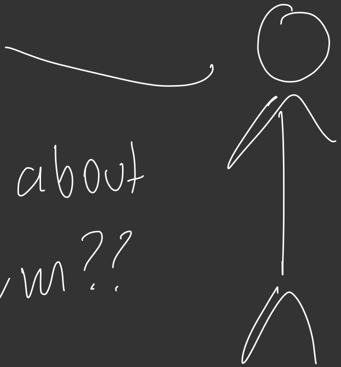
<u>What</u> should I do?



Oh no!

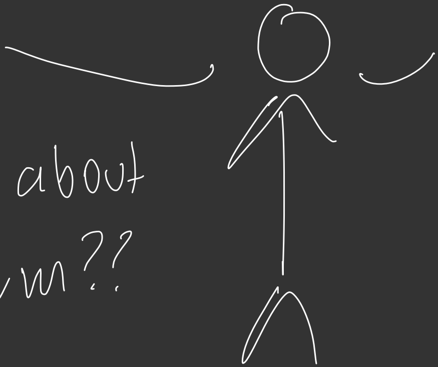
What should I do?

How should I even think about the problem??



Oh no!
What should I do?

How should I even think about the problem??

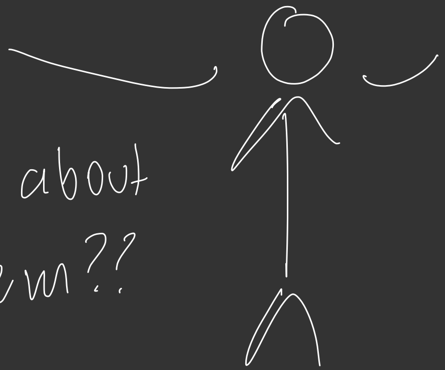


CLEARLY, I need a really fancy (Turing-complete?) language to express the problem....

Oh no!

What should I do?

How should I even think about the problem??

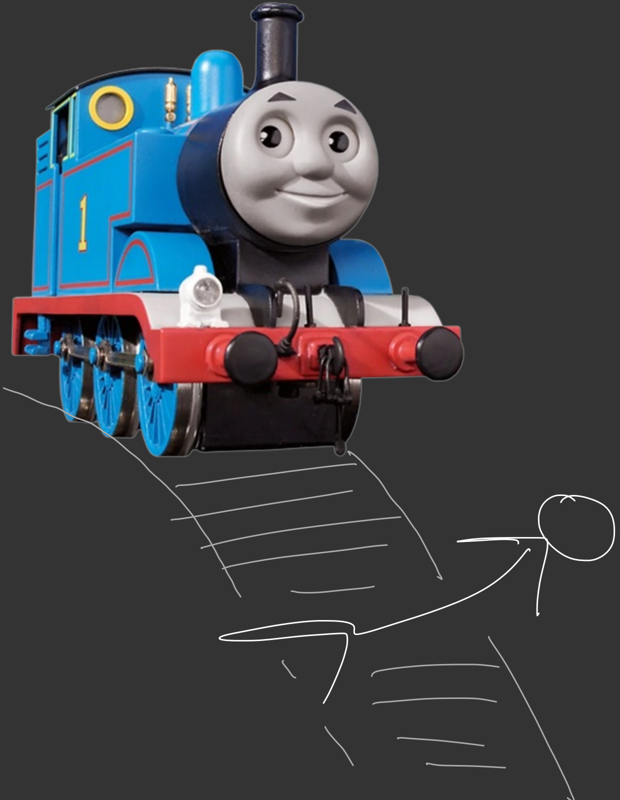


CLEARLY, I need a really fancy (Turing-complete?) language to express the problem....

... and a compiler to solve (decide) it.

Table of Contents

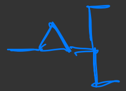
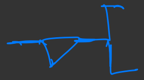
1. ~~Introduction~~
2. Dilemma Languages
3. Moral Codes
4. Crazy edge cases!



A "Dilemma Language" expresses possible ethical dilemmas.

A "Dilemma Language" expresses possible ethical dilemmas.

example: Rail Car Problems (in BNF)

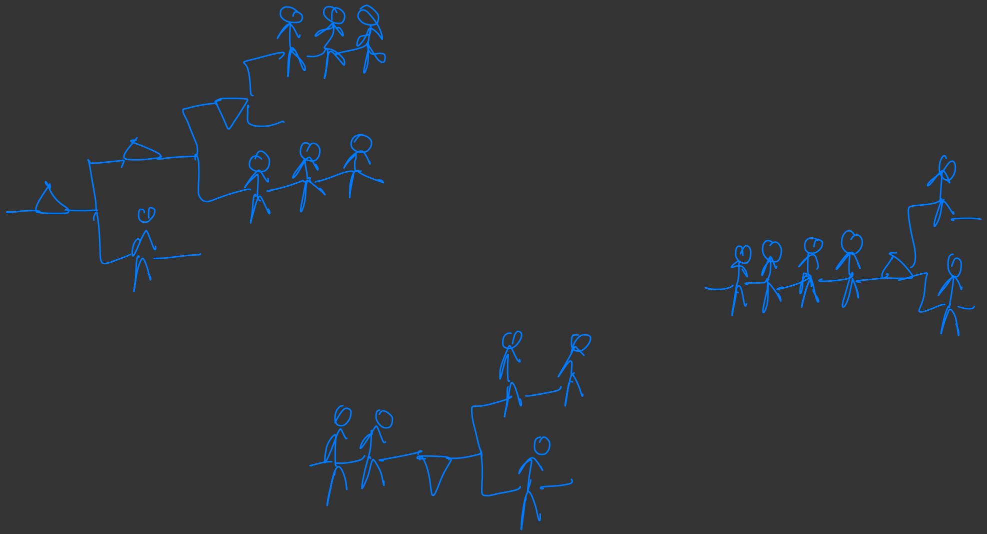
decide ::=  | 

consequence ::= | consequence consequence

RCP ::= $\text{decide}^{\text{RCP}}_{\text{RCP}}$ | consequence | Consequence RCP

A "Dilemma Language" expresses possible ethical dilemmas.

Some examples of Rail Car Problems:



A "moral code" is a compiler from the dilemma language, to the dilema language.

A "moral code" is a compiler from the dilemma language, to the dilema language.

examples: $\text{Kant}[\![\text{RCP}]\!] = \emptyset$ // do nothing

A "moral code" is a compiler from the dilemma language, to the dilema language.

examples: $\text{Kant}[\![\text{RCP}]\!] = \emptyset$ // do nothing

IDK random utilitarian dude $[\![\text{RCP}]\!] =$ a little more complex....

IDK random
utilitarian
dude $[\![\text{RCP}]\!] =$

$$C = \text{decide}^{[\cdot]}_{[\cdot]} \mid \text{consequence } C$$ // context grammar with which to make the decision

IDK random utilitarian dude $[\![\text{RCP}]\!] =$

$$C = \text{decide} \begin{matrix} [\circ] \\ {[\circ]} \end{matrix} \mid \text{consequence } C$$

// context grammar with which to make the decision

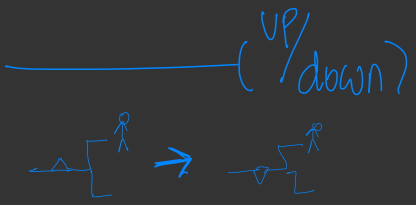
decide $[\circ]$ $[\circ]$ $\equiv$ $\text{decide} \begin{matrix} [\circ] \\ {[\circ]} \end{matrix}$

// shitty pseudo-BNF equivalence relation on decisions

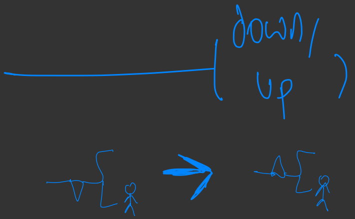
IDK random utilitarian dude $[\![RCP]\!] =$

$C = \text{decide}^{[\circ]}_{[\circ]} \mid \text{consequence } C$ // context grammar with which to make the decision

$\text{decide}^{🧍[\circ]}_{🧍[\circ]} \equiv \text{decide}^{[\circ]}_{[\circ]}$ // shitty pseudo-BNF equivalence relation on decisions

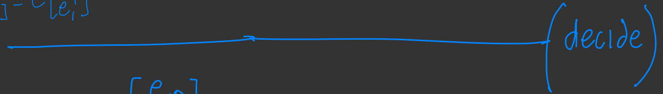


// switch down to save a life



// switch up to save a life

$C^{[e_0]}_{[e_1]} \equiv C^{[e_0']}_{[e_1']}$ $\quad |e_0| < |e_0'|$ $\quad |e_1| < |e_1'|$ $\quad C^{[e_0']}_{[e_1']} \rightarrow x$



$$C^{[e_0]}_{[e_1]} \longrightarrow x$$

// simplify the equation by subtracting from each side until you can solve it in 1-🧍 normal form

I guess that's cool or whatever

....

but how do I decide which moral code to use?

For moral code $\mathbb{C}$ over dilemma language $\mathcal{L}$:

$$\text{expected \# deaths} = \operatorname{avg}\{\,\text{num-deaths}(\mathbb{C}[\omega]) \mid \omega \in \mathcal{L}\,\}$$

$$\text{expected lives saved} = \operatorname{avg}\{\,\text{num-deaths}(\mathbb{C}^c[\omega]) \mid \omega \in \mathcal{L}\,\}$$

// where $\mathbb{C}^c$ just makes any decision except for $\mathbb{C}$

expected people killed who would not have been killed had you done nothing = //... not sure how to calculate this in a logically reasonable manner...

that seems reasonable but...

that seems reasonable but...

(

... what if the dilemma language were

INFINITE?

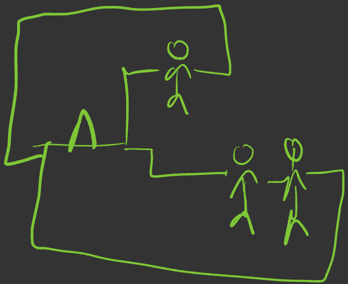
(| (

h

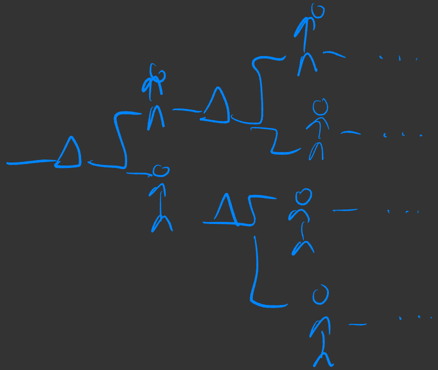
(

mua ha ha ha ha

Recursion...

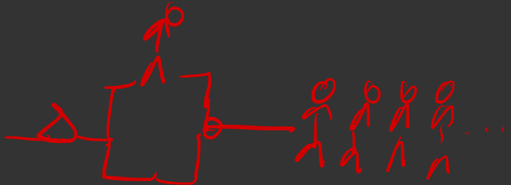


Infinite trees



Infinite Consequence

we could extend $\mathcal{L}$...



CONCLUSION

Dilemma languages → cool new way to express moral challenges

Moral Codes → formalism with which to compare (informal) moral codes over given challenges

How can we even compare codes over ∞-$\mathcal{L}$'s?

Are there languages having measure > 0?

Maybe cool applications to self-driving, etc.?