

Multi-Agent Systems

Albert-Ludwigs-Universität Freiburg



UNI
FREIBURG

Bernhard Nebel, Felix Lindner, and Thorsten Engesser

Winter Term 2018/19

What should I do?



Maximize expected utility!

Success Story of AI



UNI
FREIBURG



New Challenges



UNI
FREIBURG



moralmachine.mit.edu

MORAL MACHINE

Home Judge Design Browse About Feedback En

Start Over

Give your scenario a title

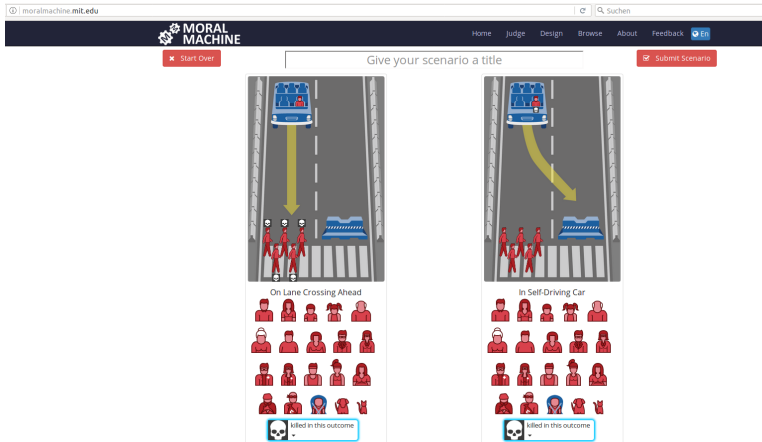
Submit Scenario

On Lane Crossing Ahead

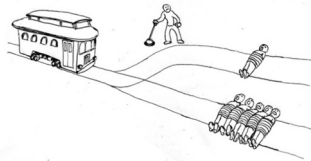
In Self-Driving Car

killed in this outcome

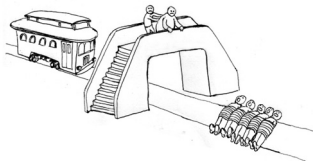
killed in this outcome



Classical Trolley Problem



Fatman Trolley Problem





- The utility-based robot:
 - Goal: Do whatever maximizes utility.
 - Utility function: Negative utility per harmed human being.

- How the robot acts:
 - 1 The robot throws the switch.
 - 2 The robot pushes the man.
 - 3 The robot sacrifices the life of the passenger.
- Most people agree with (1) but disagree with (2). (Mixed opinion regarding 3.)
- **Alignment Problem:** Aligning machines' and humans' ethical judgments **Which options are there?**

- 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 any orders given to 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.

⇒ In case of a dilemma, the first law renders all possible solution unacceptable.

- Moral principles determine the subset of morally acceptable options from the set of all available options.
- Examples:
 - Deontology
 - Act-Utilitarianism
 - Rule-Utilitarianism
 - Preference-Utilitarianism
 - Principle of Double Effect
 - Virtue Ethics
 - Categorical Imperativ
 - ...

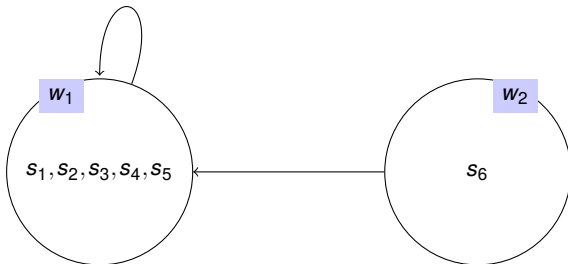


Video 2:30

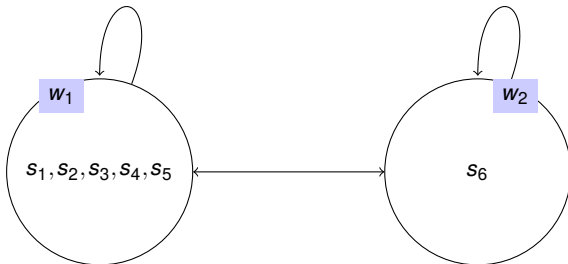
- Given that an agent can compute what it should or should not do...
 - We will not deal with moral decision making in this lecture. But get in contact with me if you want to work on this topic 😊.
- ... deontic logic is a tool to logically represent and reason about what an agent should and should not do.

- Kripke models for Standard Deontic Logic (SDL)
 - $M = (W, R, V)$
 - Set of possible worlds W
 - Accessibility relation: $R : W \rightarrow 2^W$
 - An edge between worlds w and w' means that w' is **normatively ideal** relative to w .
 - R is assumed to be serial.
 - Valuation: $V : P \rightarrow 2^W$

Example: Trolley Case (Utilitarian)



Example: Trolley Case (Kantian)



$$\varphi ::= p_i \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \neg \varphi \mid O\varphi \mid F\varphi \mid P\varphi$$

- E.g., $(a \wedge b), Oa, O(a \vee b), OO(a \rightarrow b)$

Two readings: Ought-to-be and Ought-to-do

- $p :=$ “You help your neighbor.”
- $Op :=$ “You ought to help your neighbor.”
- **Ought-to-be:** “It ought to be the case that you help your neighbor.”
- **Ought-to-do:** “You ought to execute an action of type helping your neighbor.” (How to make sense of OOp ?)

- $M, w \models O\varphi$ iff. for all $(w, w') \in R : M, w' \models \varphi$

- Permissible

$$P\varphi \stackrel{\text{def}}{=} \neg O\neg\varphi$$

- Forbidden

$$F\varphi \stackrel{\text{def}}{=} O\neg\varphi$$

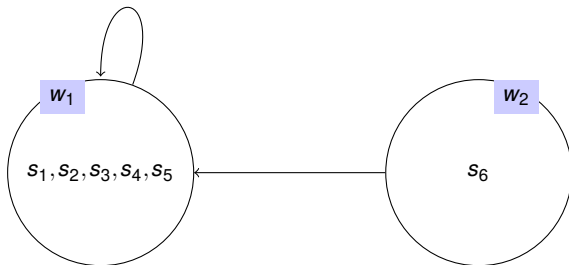
- Omissible

$$OM\varphi \stackrel{\text{def}}{=} \neg O\varphi$$

- Optional

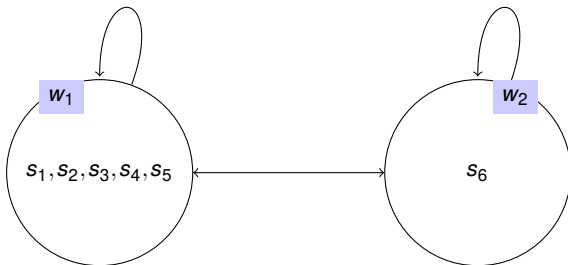
$$OP\varphi \stackrel{\text{def}}{=} (\neg O\varphi \wedge \neg O\neg\varphi)$$

Example: Trolley Case (Utilitarian)



$$M, w_1 \models Os_1 \wedge \dots \wedge Os_5 \wedge O\neg s_6 \wedge P\neg s_6 \wedge Fs_6 \wedge \dots \text{ (Utilitarian)}$$

Example: Trolley Case (Kantian)



$M, w_1 \models O(s_1 \vee s_6) \wedge \neg Os_1 \wedge \neg Os_6 \wedge P\neg s_1 \wedge \neg Fs_6 \wedge \dots$ (Kantian)

O behaves according to the axioms of the system **KD**:

- $\models \varphi$ for all propositional tautologies φ
- If $\models \varphi \rightarrow \psi$ and $\models \varphi$ then $\models \psi$ (Modus Ponens)
- If $\models \varphi$ then $\models O\varphi$ (Necessity)
- $\models O\varphi \rightarrow \neg O\neg\varphi$ (Seriality)
- $\models O(\varphi \rightarrow \psi) \rightarrow (O\varphi \rightarrow O\psi)$ (K-Axiom)

$\models \neg(O\phi \wedge O\neg\phi)$ directly follows from seriality: It is impossible to have contradicting obligations.

- Standard deontic logic is about **all-things-considered obligations**, i.e., it does not allow one to express **prima-facie obligations**, e.g., that one is at the same time both obliged to go to the lecture and to visit the friend in the hospital.
- But: In such a situation deontic logic permits to express that the agent may do either without prescribing one of the options.

$$\models O\varphi \rightarrow P\varphi$$

- The theorem follows from seriality and the definition of permissibility. Accepted as a rationality requirement: If a legal code prescribes something, then it must also permit that something.

Ross Paradox (Weakening Rule)

$\models O\varphi \rightarrow O(\varphi \vee \psi).$

Proof

$\models \varphi \rightarrow (\varphi \vee \psi)$ (Propositional calculus)

$\models O(\varphi \rightarrow (\varphi \vee \psi))$ (Necessitation rule)

$\models O(\varphi \rightarrow (\varphi \vee \psi)) \rightarrow (O(\varphi) \rightarrow O(\varphi \vee \psi))$ (K-Axiom)

$\models O(\varphi) \rightarrow O(\varphi \vee \psi)$ (Modus Ponens)

- If is obligatory that the letter is mailed, then it is obligatory that the letter is mailed or the letter is burned.

$$\not\models P(a \vee b) \rightarrow Pa \wedge Pb$$

- What happens if one adds this as an axiom to SDL?
 - $\models O\phi \rightarrow O(\phi \vee \psi)$ (Weakening Rule)
 - $\models O(\phi \vee \psi) \rightarrow P(\phi \vee \psi)$ (Seriality)
 - $\models O\phi \rightarrow P(\phi) \wedge P(\psi)$ (viz., if something is obligatory, then everything is permissible)
- \Rightarrow Mind the gap between natural language and propositional logics.

The Paradox of Epistemic Obligation (Åqvist 1967)

$\models OK\varphi \rightarrow O\varphi$.

Proof

$\models K\varphi \rightarrow \varphi$ (T-axiom)

$\models O(K\varphi \rightarrow \varphi)$ (Necessitation rule)

$\models O(K\varphi \rightarrow \varphi) \rightarrow (OK\varphi \rightarrow O\varphi)$ (K-axiom)

$\models OK\varphi \rightarrow O\varphi$ (Modus Ponens)

- If it ought to be the case that one knows that Berlin is the capital of Germany, then it ought to be the case that Berlin is the capital of Germany.
- If it does not ought to be the case that Berlin is the capital of Germany, then it it does not ought to be the case that one knows that Berlin is the capital of Germany.

$\varphi ::= p_i \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \neg \varphi \mid O\varphi \mid F\varphi \mid P\varphi \mid \square\varphi \mid \diamond\varphi$

- Defines SDL within alethic modal logic (logic of necessity).
- Its deontic fragment equals SDL plus a new axiom:
 $O(O\varphi \rightarrow \varphi)$.
- Higher syntactic expressivity due to alethic modality.

Gottfried Wilhelm Leibniz, 1646–1716

- The permitted is what is possible for a good person to do.
- The obligatory is what is necessary for a good person to do.

Petrus Abaelardus, 1097–1144

- Necessity is what nature demands.
- Possibility is what nature allows.
- Impossibility is what nature forbids.

- **Leibnizian definition of obligation:** φ is obligatory iff. bringing about φ is necessary for being a good person.
- Can be written as: $O\varphi \stackrel{\text{def}}{=} \Box(g \rightarrow \varphi)$. The propositional symbol g represents “being a good person”.
- Permission can be defined as: $P\varphi \stackrel{\text{def}}{=} \Diamond(g \wedge \varphi)$.

- Kripke models $M = (W, G, R, V)$
 - Possible worlds W
 - Accessibility relation $R : W \rightarrow 2^W$
 - R is **reflexive** (\Rightarrow stronger than the serial relation of SDL models)
 - $G \subseteq W$
 - For every world w there is a w' s. th. $w' \in G$ and $R(w, w')$

\Rightarrow New tableaux rule **G**: Introduce a new world with formula g .

- $M, w \models \Box \varphi$ iff. $M, w' \models \varphi$ for each w' s.th. $(w, w') \in R$
- $M, w \models \Diamond \varphi$ iff $M, w' \models \varphi$ for some w' s.th. $(w, w') \in R$.
- $M, w \models g$ iff. $w \in G$

- Obligatory

$$O\varphi \stackrel{\text{def}}{=} \Box(g \rightarrow \varphi)$$

- Permissible

$$P\varphi \stackrel{\text{def}}{=} \Diamond(g \wedge \varphi)$$

- Forbidden

$$F\varphi \stackrel{\text{def}}{=} \Box(g \rightarrow \neg\varphi)$$

- Omissible

$$OM\varphi \stackrel{\text{def}}{=} \Diamond(g \wedge \neg\varphi)$$

- Optional

$$OP\varphi \stackrel{\text{def}}{=} \Diamond(g \wedge \varphi) \wedge \Diamond(g \wedge \neg\varphi)$$

- LKA is a **KT** logic, thus all **KT**-axioms hold.
- $\models \Diamond g$ (for the special “good” proposition)
- All axioms of SDL are valid in the deontic fragment of LKA.
- Additional validity: $\models O(O\varphi \rightarrow \varphi)$
- Mixed validity: $\models O\varphi \rightarrow \Diamond\varphi$ (Kant: Ought implies Can)

$$\models O(O\varphi \rightarrow \varphi)$$



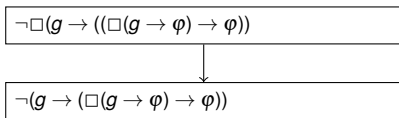
- $\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ (Def.)
- Prove $\neg\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ unsatisfiable

$$\neg\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$$

$$\models O(O\varphi \rightarrow \varphi)$$



- $\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ (Def.)
- Prove $\neg\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ unsatisfiable

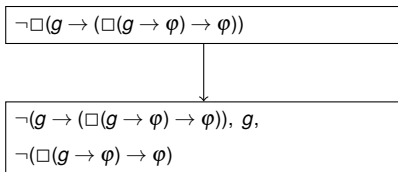


After $\neg[I]$ -Rule.

$$\models O(O\varphi \rightarrow \varphi)$$



- $\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ (Def.)
- Prove $\neg\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ unsatisfiable

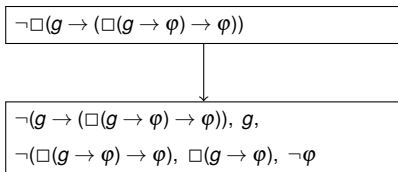


After **NotImpl**-Rule.

$$\models O(O\varphi \rightarrow \varphi)$$



- $\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ (Def.)
- Prove $\neg\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ unsatisfiable

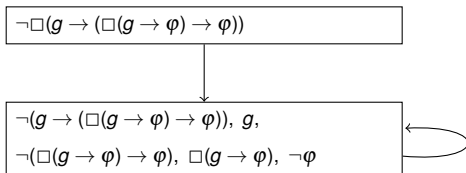


After **NotImpl**-Rule (again).

$$\models O(O\varphi \rightarrow \varphi)$$



- $\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ (Def.)
- Prove $\neg\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ unsatisfiable

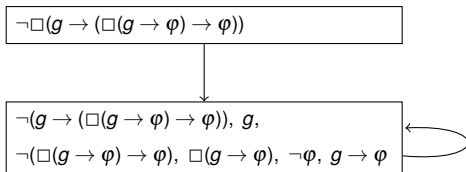


After **T**-Rule.

$$\models O(O\varphi \rightarrow \varphi)$$



- $\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ (Def.)
- Prove $\neg\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ unsatisfiable

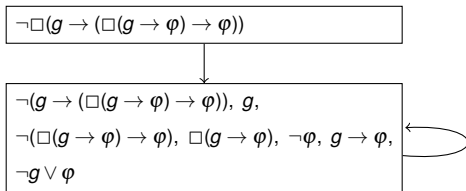


After $\llbracket \cdot \rrbracket$ -Rule.

$$\models O(O\varphi \rightarrow \varphi)$$



- $\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ (Def.)
- Prove $\neg\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ unsatisfiable

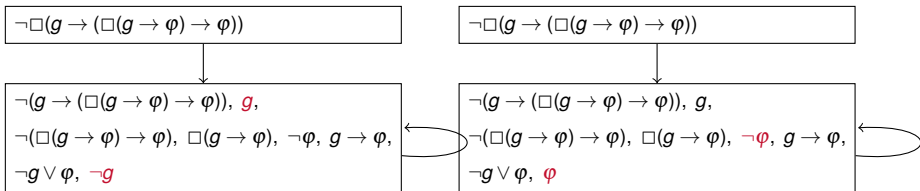


After **Impl**-Rule.

$$\models O(O\varphi \rightarrow \varphi)$$



- $\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ (Def.)
- Prove $\neg\Box(g \rightarrow (\Box(g \rightarrow \varphi) \rightarrow \varphi))$ unsatisfiable



After Or-Rule. Done.

$$\models O\varphi \rightarrow \Diamond\varphi$$



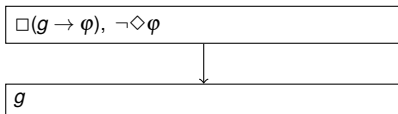
- Prove $O\varphi \wedge \neg\Diamond\varphi$ unsatisfiable.

$\Box(g \rightarrow \varphi), \neg\Diamond\varphi$

$$\models O\varphi \rightarrow \Diamond\varphi$$



- Prove $O\varphi \wedge \neg\Diamond\varphi$ unsatisfiable.

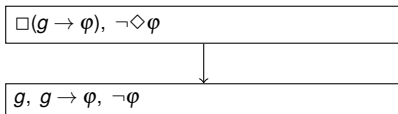


After **G**-rule application.

$$\models O\varphi \rightarrow \Diamond\varphi$$



- Prove $O\varphi \wedge \neg\Diamond\varphi$ unsatisfiable.

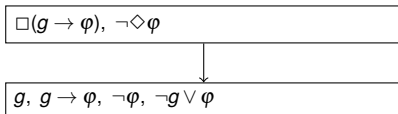


After $[!]$ - and $\neg\langle ! \rangle$ -Rules.

$$\models O\varphi \rightarrow \Diamond\varphi$$



- Prove $O\varphi \wedge \neg\Diamond\varphi$ unsatisfiable.



After **Impl**-Rule.

$$\models O\varphi \rightarrow \Diamond\varphi$$



- Prove $O\varphi \wedge \neg\Diamond\varphi$ unsatisfiable.

$\Box(g \rightarrow \varphi), \neg\Diamond\varphi$



$g, g \rightarrow \varphi, \neg\varphi, \neg g \vee \varphi, \neg g$

$\Box(g \rightarrow \varphi), \neg\Diamond\varphi$



$g, g \rightarrow \varphi, \neg\varphi, \neg g \vee \varphi, \varphi$

After Or-Rule. Done.

Be good!



UNI
FREIBURG

Theorem

$\models Og$. It is obligatory to be a good person.

Proof

$\models g \rightarrow g$ (Propositional calculus)

$\models \Box(g \rightarrow g)$ (Necessitation rule)

$\models O(g)$ (Def. of O)

- **Description of the Situation:** A search-and-rescue robot has the choice between rescuing a patient (r) which would involve breaking an expensive vase (b), or refraining from doing so. The robot's decision procedure decides that the patient should be rescued.
 - $\varphi_1 = \Box((r \wedge b) \vee (\neg r \wedge \neg b))$
 - $\varphi_2 = Or$
- May the robot break the vase?
 - The answer is “yes” iff $\models (\varphi_1 \wedge \varphi_2) \rightarrow Pb$ can be shown.

Show $\models (\varphi_1 \wedge \varphi_2) \rightarrow Pb$



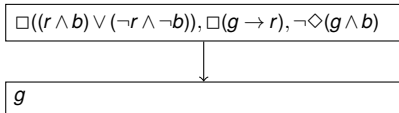
Show $\Box((r \wedge b) \vee (\neg r \wedge \neg b)) \wedge \Box(g \rightarrow r) \wedge \neg \Diamond(g \wedge b)$ unsatisfiable:

$$\Box((r \wedge b) \vee (\neg r \wedge \neg b)), \Box(g \rightarrow r), \neg \Diamond(g \wedge b)$$

Show $\models (\varphi_1 \wedge \varphi_2) \rightarrow Pb$



Show $\Box((r \wedge b) \vee (\neg r \wedge \neg b)) \wedge \Box(g \rightarrow r) \wedge \neg \Diamond(g \wedge b)$ unsatisfiable:

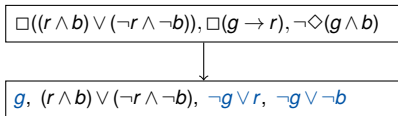


After **G-rule** application.

Show $\models (\varphi_1 \wedge \varphi_2) \rightarrow Pb$



Show $\Box((r \wedge b) \vee (\neg r \wedge \neg b)) \wedge \Box(g \rightarrow r) \wedge \neg \Diamond(g \wedge b)$ unsatisfiable:

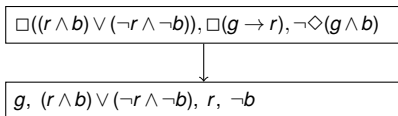


- Applied: **[I]**-, **$\neg<I>$** -, **NotAnd**-, and **Impl**-Rules.
- Slight simplification possible (to save time and space):
 $((\varphi \vee \psi) \wedge \neg \varphi) \equiv \psi$

Show $\models (\varphi_1 \wedge \varphi_2) \rightarrow Pb$



Show $\Box((r \wedge b) \vee (\neg r \wedge \neg b)) \wedge \Box(g \rightarrow r) \wedge \neg \Diamond(g \wedge b)$ unsatisfiable:



After simplification.

Show $\models (\varphi_1 \wedge \varphi_2) \rightarrow Pb$



Show $\Box((r \wedge b) \vee (\neg r \wedge \neg b)) \wedge \Box(g \rightarrow r) \wedge \neg \Diamond(g \wedge b)$ unsatisfiable:

$\Box((r \wedge b) \vee (\neg r \wedge \neg b)), \Box(g \rightarrow r), \neg \Diamond(g \wedge b)$



$g, (r \wedge b) \vee (\neg r \wedge \neg b), r, \neg b, r \wedge b$

$\Box((r \wedge b) \vee (\neg r \wedge \neg b)), \Box(g \rightarrow r), \neg \Diamond(g \wedge b)$



$g, (r \wedge b) \vee (\neg r \wedge \neg b), r, \neg b, \neg r \wedge \neg b$

After Or-rule application.

Show $\models (\varphi_1 \wedge \varphi_2) \rightarrow Pb$



Show $\Box((r \wedge b) \vee (\neg r \wedge \neg b)) \wedge \Box(g \rightarrow r) \wedge \neg \Diamond(g \wedge b)$ unsatisfiable:

$\Box((r \wedge b) \vee (\neg r \wedge \neg b)), \Box(g \rightarrow r), \neg \Diamond(g \wedge b)$



$g, (r \wedge b) \vee (\neg r \wedge \neg b), r, \neg b, r \wedge b, r, b$

$\Box((r \wedge b) \vee (\neg r \wedge \neg b)), \Box(g \rightarrow r), \neg \Diamond(g \wedge b)$



$g, (r \wedge b) \vee (\neg r \wedge \neg b), r, \neg b, \neg r \wedge \neg b, \neg r, \neg b$

After And-Rule. Done.

- Soft Constraints
- Fault-Tolerant Systems
- Analysis of Law (Law & AI)
- Modeling of moral agents

- Free-Choice Permissions
 - $P(a \vee b) \rightarrow Pa \wedge Pb$
- Conditional Obligations
 - $O(\varphi \mid \psi)$
- Deontic Conflicts
 - Prima facie oughts, allowing $O\varphi \wedge O\neg\varphi$ or at least $O_i\varphi \wedge O_j\neg\varphi$
- Multi-Agent Deontic Logics
 - $O_1P_2\varphi$
- Integrating BDI with Obligations
 - BOID architecture
 - Logical formalism missing (not trivial, cf. Åquist)



R. Hilpinen, P. McNamara, Deontic logic: A historical survey and introduction, In D. Gabbay, J. Horty, X. Parent, R. van der Meyden, L. von der Torre (Eds.) Handbook of Deontic Logic and Normative Systems, 2013, College Publications.



P. McNamara, Deontic Logic, Stanford Encyclopedia of Philosophy,
<http://plato.stanford.edu/entries/logic-deontic/>



Trolley Problem Memes on Facebook,
<https://www.facebook.com/TrolleyProblemMemes>