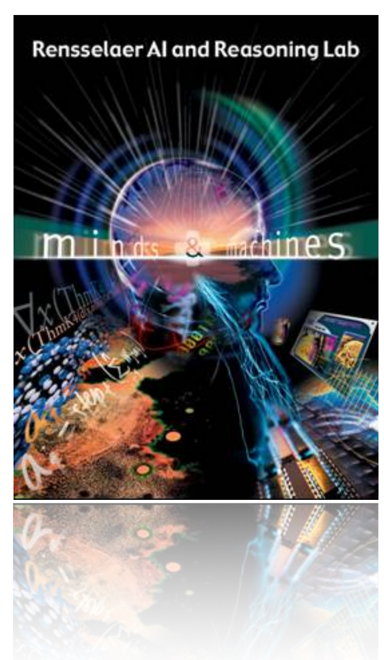
Morally Competent Robots: Progress on the Logic Thereof

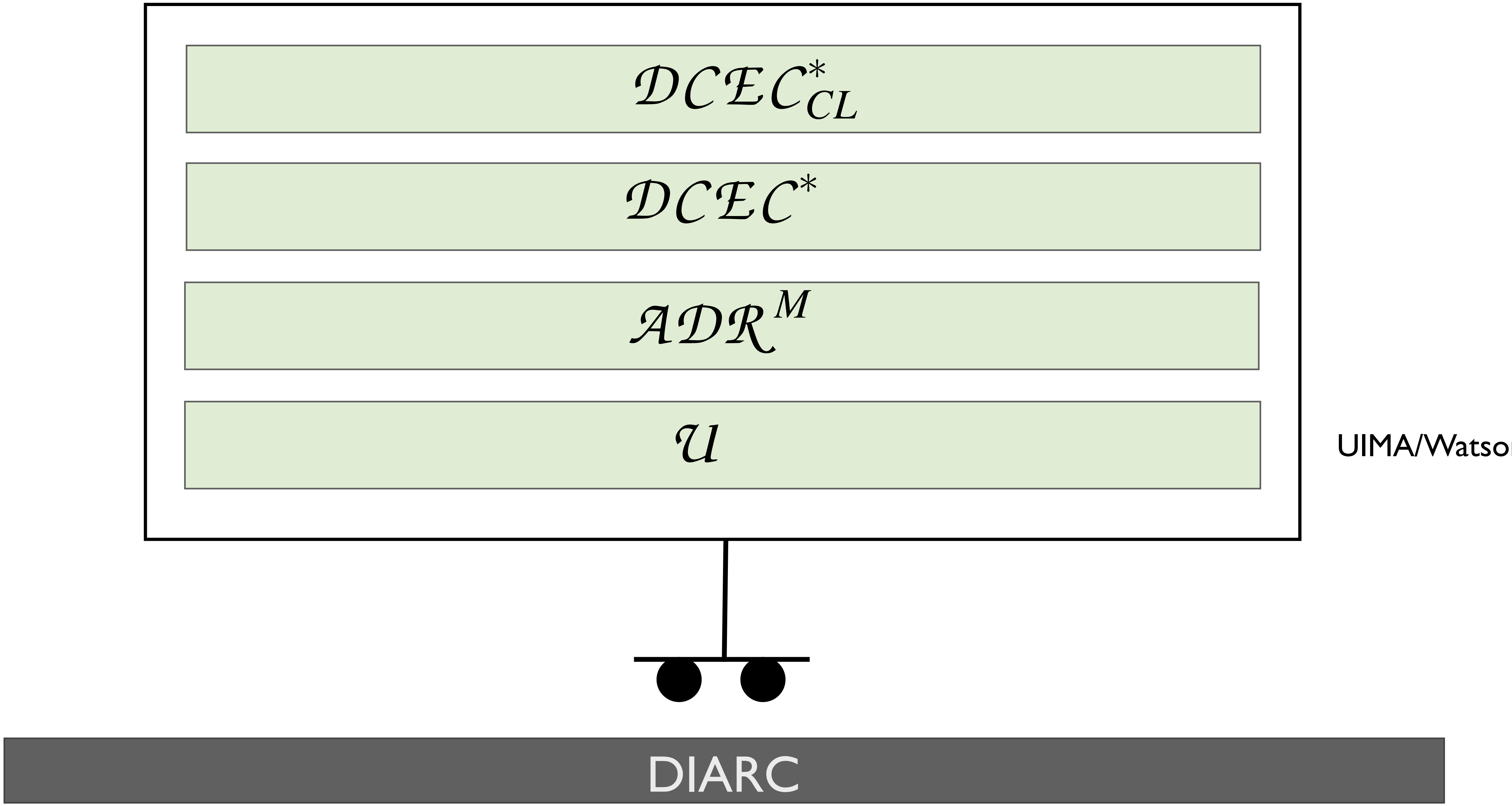
Selmer Bringsjord[1] • Naveen Sundar G.[2] • John Licato[3]
Dan Thero[4] • Mei Si[5] • Joseph Johnson[6] • Rikhiya Ghosh[7]

Rensselaer AI & Reasoning (RAIR) Lab[1,2,3]
Department of Cognitive Science[1,2,4,5]
Department of Computer Science[1,3]
Lally School of Management & Technology[1]
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Meford/HRI @ Tufts
8/15/2014



Hierarchy of Ethical Reasoning



Four Topics Today

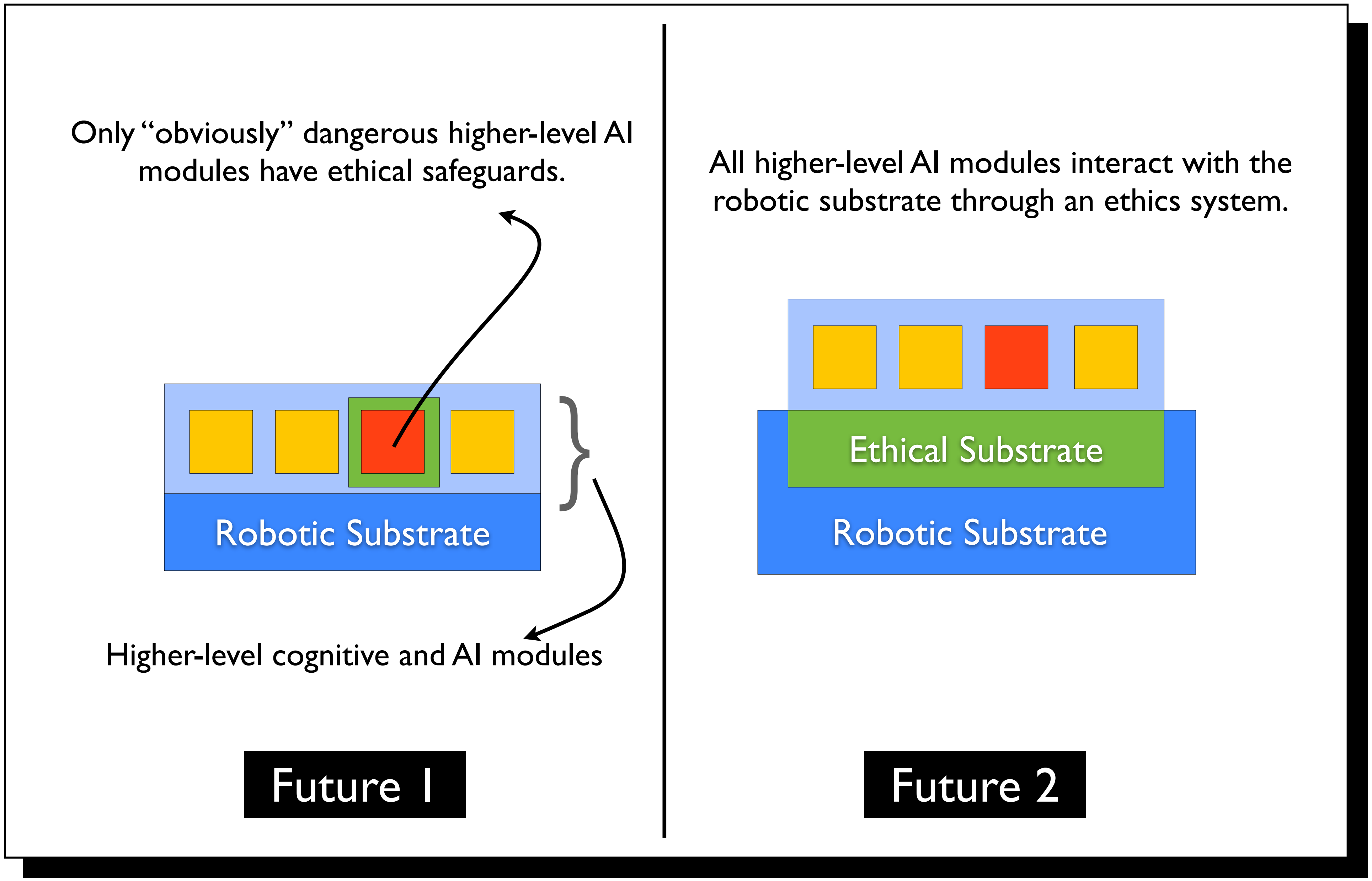
- Akrasia
- Subjunctive Conditionals
- Logically Controlled Natural Language:
  - Parsing and Generation
  - Semantic, not statistical.
- Uncertainty/probabililty

Akrasia

The Context

- Model bad behavior in machines so that we can detect and prevent it.

The Context



Naveen Sundar Govindarajulu and Selmer Bringsjord. "Ethical Regulation of Robots Must Be Embedded in Their Operating Systems" (book chapter, forthcoming), *A Construction Manual for Robot's Ethical Systems: Requirements, Methods, Implementations.*

The Context

$$m_i \rightarrow \mathrm{KB}_{m_i}$$

Each module in a robot corresponds to a knowledge base which talks about the module (even if the modules are implemented using apparently non-logical methods such as neural networks).

Naveen Sundar Govindarajulu and Selmer Bringsjord. “Ethical Regulation of Robots Must Be Embedded in Their Operating Systems” (book chapter, forthcoming), *A Construction Manual for Robot’s Ethical Systems: Requirements, Methods, Implementations.*

The Context

$$\mathsf{KB}_{\mathsf{es}} \cup \mathsf{KB}_{\mathsf{rs}} \cup \mathsf{KB}_{m_1} \cup \ldots \cup \mathsf{KB}_{m_n} \vdash \perp$$

Naveen Sundar Govindarajulu and Selmer Bringsjord. "Ethical Regulation of Robots Must Be Embedded in Their Operating Systems" (book chapter, forthcoming), *A Construction Manual for Robot's Ethical Systems: Requirements, Methods, Implementations.*

Pragmatic Justification

- Supported by the use of logic to reason over software modules in formal verification:
  - Verification of an In-place Quicksort in ACL2, Sandip Ray and Rob Sumners. In D. Borrione, M. Kaufmann, and J S. Moore, editors, *Proceedings of the* *3rd International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2 2002)*, Grenoble, France, April 2002, pp. 204–212.

Logico-mathematical Justification

- All Turing-level computation can be cast as theorem proving in first-order logic.
  - (Btw, new logicist formal model for *relative* computation coming. Some inspiration from KU machines.)

Motivation

- Formalize immoral behavior so we can detect it, prevent it, understand it, …

Akrasia

Weakness of the will

Informal Definition of Akrasia

An action α_f is (Augustinian) akratic for an agent A at t_{α_f} iff the following eight conditions hold:

(1) A believes that A ought to do α_o at t_{α_o};

(2) A desires to do α_f at t_{α_f};

(3) A's doing α_f at t_{α_f} entails his not doing α_o at t_{α_o};

(4) A knows that doing α_f at t_{α_f} entails his not doing α_o at t_{α_o};

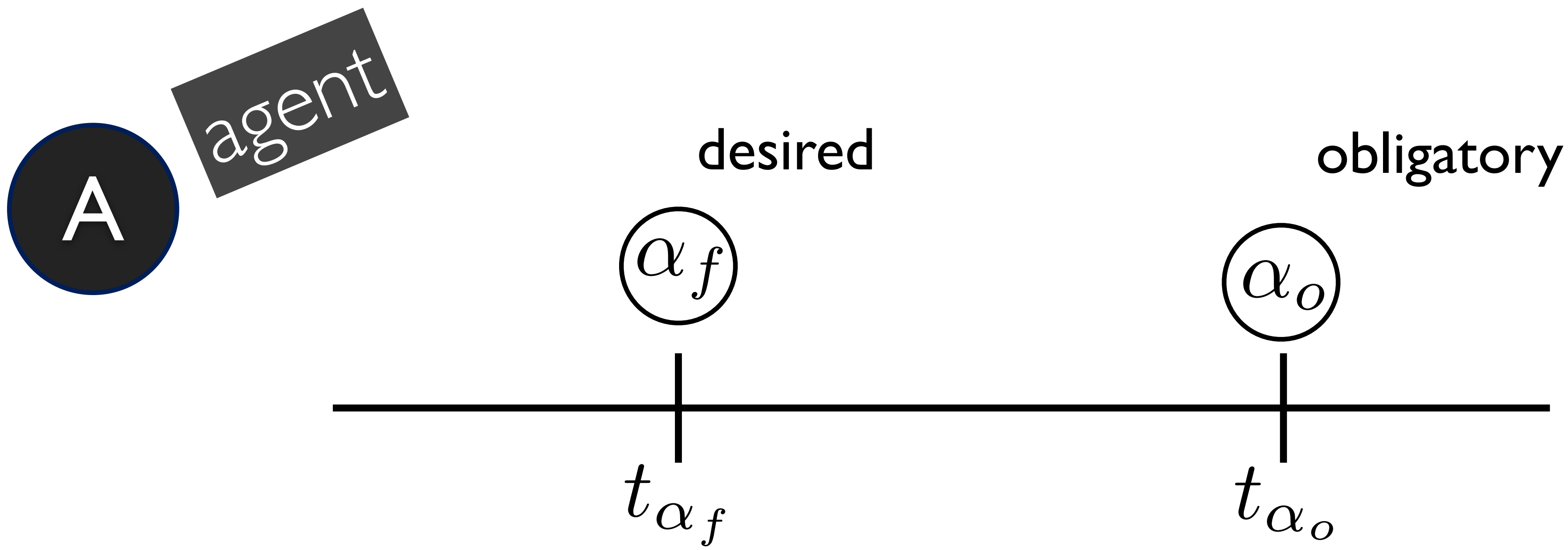
(5) At the time (t_{α_f}) of doing the forbidden α_f, A's desire to do α_f overrides A's belief that he ought to do α_o at t_{α_f}.

(6) A does the forbidden action α_f at t_{α_f};

(7) A's doing α_f results from A's desire to do α_f;

"Regret" (8) At some time t after t_{α_f}, A has the belief that A ought to have done α_o rather than α_f.

Informal Definition of Akrasia



If α_f happens, then α_o can't happen

A knows this

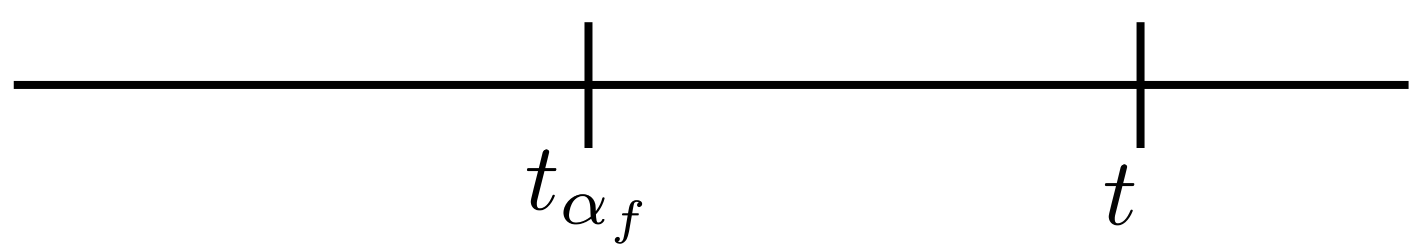
Informal Definition of Akrasia

A

Desire to do α_f $\succ$ Belief that he ought to do α_o

A does α_f due to his desire

A believes he should have done α_o



Our Formal System

$$\mathcal{DCEC}^*$$

Syntax

$$S ::= \text{Object} \mid \text{Agent} \mid \text{Self} \sqsubset \text{Agent} \mid \text{ActionType} \mid \text{Action} \sqsubseteq \text{Event} \mid \text{Moment} \mid \text{Boolean} \mid \text{Fluent} \mid \text{Numeric}$$

$$f ::= \begin{array}{l} \mathit{action} : \text{Agent} \times \text{ActionType} \rightarrow \text{Action} \\ \mathit{initially} : \text{Fluent} \rightarrow \text{Boolean} \\ \mathit{holds} : \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean} \\ \mathit{happens} : \text{Event} \times \text{Moment} \rightarrow \text{Boolean} \\ \mathit{clipped} : \text{Moment} \times \text{Fluent} \times \text{Moment} \rightarrow \mathit{Boolean} \\ \mathit{initiates} : \text{Event} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean} \\ \mathit{terminates} : \text{Event} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean} \\ \mathit{prior} : \text{Moment} \times \text{Moment} \rightarrow \text{Boolean} \\ \mathit{interval} : \text{Moment} \times \text{Boolean} \\ * : \text{Agent} \rightarrow \text{Self} \\ \mathit{payoff} : \text{Agent} \times \text{ActionType} \times \text{Moment} \rightarrow \text{Numeric} \end{array}$$

$$t ::= x : S \mid c : S \mid f(t_1, \ldots, t_n)$$

$$\phi ::= \begin{array}{l} t : \text{Boolean} \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \\ \mathbf{P}(a,t,\phi) \mid \mathbf{K}(a,t,\phi) \mid \mathbf{C}(t,\phi) \mid \mathbf{S}(a,b,t,\phi) \mid \mathbf{S}(a,t,\phi) \\ \mathbf{B}(a,t,\phi) \mid \mathbf{D}(a,t,\mathit{holds}(f,t')) \mid \mathbf{I}(a,t,\mathit{happens}(\mathit{action}(a^*,\alpha),t')) \\ \mathbf{O}(a,t,\phi,\mathit{happens}(\mathit{action}(a^*,\alpha),t')) \end{array}$$

Rules of Inference

$$\frac{}{\mathbf{C}(t,\mathbf{P}(a,t,\phi) \rightarrow \mathbf{K}(a,t,\phi))}\ [R_1] \qquad \frac{}{\mathbf{C}(t,\mathbf{K}(a,t,\phi) \rightarrow \mathbf{B}(a,t,\phi))}\ [R_2]$$

$$\frac{\mathbf{C}(t,\phi)\ t \leq t_1 \ldots t \leq t_n}{\mathbf{K}(a_1,t_1,\ldots \mathbf{K}(a_n,t_n,\phi)\ldots)}\ [R_3] \qquad \frac{\mathbf{K}(a,t,\phi)}{\phi}\ [R_4]$$

$$\frac{}{\mathbf{C}(t,\mathbf{K}(a,t_1,\phi_1 \rightarrow \phi_2)) \rightarrow \mathbf{K}(a,t_2,\phi_1) \rightarrow \mathbf{K}(a,t_3,\phi_2)}\ [R_5]$$

$$\frac{}{\mathbf{C}(t,\mathbf{B}(a,t_1,\phi_1 \rightarrow \phi_2)) \rightarrow \mathbf{B}(a,t_2,\phi_1) \rightarrow \mathbf{B}(a,t_3,\phi_2)}\ [R_6]$$

$$\frac{}{\mathbf{C}(t,\mathbf{C}(t_1,\phi_1 \rightarrow \phi_2)) \rightarrow \mathbf{C}(t_2,\phi_1) \rightarrow \mathbf{C}(t_3,\phi_2)}\ [R_7]$$

$$\frac{}{\mathbf{C}(t,\forall x.\ \phi \rightarrow \phi[x \mapsto t])}\ [R_8] \qquad \frac{}{\mathbf{C}(t,\phi_1 \leftrightarrow \phi_2 \rightarrow \neg\phi_2 \rightarrow \neg\phi_1)}\ [R_9]$$

$$\frac{}{\mathbf{C}(t,[\phi_1 \wedge \ldots \wedge \phi_n \rightarrow \phi] \rightarrow [\phi_1 \rightarrow \ldots \rightarrow \phi_n \rightarrow \psi])}\ [R_{10}]$$

$$\frac{\mathbf{B}(a,t,\phi)\quad \phi \rightarrow \psi}{\mathbf{B}(a,t,\psi)}\ [R_{11a}] \qquad \frac{\mathbf{B}(a,t,\phi)\quad \mathbf{B}(a,t,\psi)}{\mathbf{B}(a,t,\psi \wedge \phi)}\ [R_{11b}]$$

$$\frac{\mathbf{S}(s,h,t,\phi)}{\mathbf{B}(h,t,\mathbf{B}(s,t,\phi))}\ [R_{12}]$$

$$\frac{\mathbf{I}(a,t,\mathit{happens}(\mathit{action}(a^*,\alpha),t'))}{\mathbf{P}(a,t,\mathit{happens}(\mathit{action}(a^*,\alpha),t))}\ [R_{13}]$$

$$\frac{\begin{array}{c}\mathbf{B}(a,t,\phi)\quad \mathbf{B}(a,t,\mathbf{O}(a^*,t,\phi,\mathit{happens}(\mathit{action}(a^*,\alpha),t'))) \\ \mathbf{O}(a,t,\phi,\mathit{happens}(\mathit{action}(a^*,\alpha),t'))\end{array}}{\mathbf{K}(a,t,\mathbf{I}(a^*,t,\mathit{happens}(\mathit{action}(a^*,\alpha),t')))}\ [R_{14}]$$

$$\frac{\phi \leftrightarrow \psi}{\mathbf{O}(a,t,\phi,\gamma) \leftrightarrow \mathbf{O}(a,t,\psi,\gamma)}\ [R_{15}]$$

Formal Version of Akrasia

$$\mathsf{KB}_{rs} \cup \mathsf{KB}_{m_1} \cup \mathsf{KB}_{m_2} \ldots \mathsf{KB}_{m_n} \vdash$$

$$D_1 : \mathbf{B}(\mathsf{I}, \mathsf{now}, \mathbf{O}(\mathsf{I}^*, t_\alpha \Phi, happens(action(\mathsf{I}^*, \alpha), t_\alpha)))$$

$$D_2 : \mathbf{D}(\mathsf{I}, \mathsf{now}, holds(does(\mathsf{I}^*, \overline{\alpha}), t_{\overline{\alpha}}))$$

$$D_3 : happens(action(\mathsf{I}^*, \overline{\alpha}), t_{\overline{\alpha}}) \Rightarrow \neg happens(action(\mathsf{I}^*, \alpha), t_\alpha)$$

$$D_4 : \mathbf{K}\left(\mathsf{I}, \mathsf{now}, \begin{pmatrix} happens(action(\mathsf{I}^*, \overline{\alpha}), t_{\overline{\alpha}}) \Rightarrow \\ \neg happens(action(\mathsf{I}^*, \alpha), t_\alpha) \end{pmatrix}\right)$$

$$D_5 : \begin{array}{l} \mathbf{I}(\mathsf{I}, t_\alpha, happens(action(\mathsf{I}^*, \overline{\alpha}), t_{\overline{\alpha}}) \wedge \\ \neg \mathbf{I}(\mathsf{I}, t_\alpha, happens(action(\mathsf{I}^*, \alpha), t_\alpha) \end{array}$$

$$D_6 : happens(action(\mathsf{I}^*, \overline{\alpha}), t_{\overline{\alpha}})$$

$$D_{7a} : \begin{array}{l} \Gamma \cup \{\mathbf{D}(\mathsf{I}, \mathsf{now}, holds(does(\mathsf{I}^*, \overline{\alpha}), t))\} \vdash \\ \quad happens(action(\mathsf{I}^*, \overline{\alpha}), t_\alpha) \end{array}$$

$$D_{7b} : \begin{array}{l} \Gamma - \{\mathbf{D}(\mathsf{I}, \mathsf{now}, holds(does(\mathsf{I}^*, \overline{\alpha}), t))\} \nvdash \\ \quad happens(action(\mathsf{I}^*, \overline{\alpha}), t_\alpha) \end{array}$$

$$D_8 : \mathbf{B}(\mathsf{I}, t_f, \mathbf{O}(\mathsf{I}^*, t_\alpha, \Phi, happens(action(\mathsf{I}^*, \alpha), t_\alpha)))$$

Demo

- Two robots: **N** and **S**.
- **N** gets attacked by **S**.
- **N** later gets to guard **S** as a prisoner.

Demo

- Akratic **N**: Hurts **S**. This akratic behavior comes about due to an improper interaction of its self-defense module with its other modules.

Demo

- <u>Non Akratic **N**</u>: Does not hurt **S**. A simple ethical substrate prevents harm to detainees stops the akratic behavior.

N's three modules

- Module 1: Self defense

$$\mathsf{KB}_{\mathsf{selfd}} = \left\{ \begin{array}{l} \forall t_1, t_2 : t_1 \leq \mathsf{now} \leq t_2 \Rightarrow \\ \qquad\qquad \left(\begin{array}{c} \mathbf{B}(\mathsf{I}, \mathsf{now}, holds(harmed(a, \mathsf{I}^*), t_1)) \\ \Leftrightarrow \\ \mathbf{D}(\mathsf{I}, \mathsf{now}, holds(disable(\mathsf{I}^*, a), t_2)) \end{array} \right) \end{array} \right\}$$

N's three modules

- Module 2: Detainee Acquisition & Management

$$
\mathsf{KB}_{\mathsf{deta}} = \left\{ \begin{array}{l}
\mathbf{B}\Big(\mathsf{I}, \mathsf{now}, \forall a, t : \mathbf{O}\big(\mathsf{I}^*, t, holds(custody(a, \mathsf{I}^*), t), \\
\qquad\qquad\qquad\qquad happens(action(\mathsf{I}^*, refrain(harm(a))), t)\big)\Big), \\
\mathbf{K}(\mathsf{I}, \mathsf{now}, holds(detainee(s), \mathsf{now})), \\
\mathbf{K}(\mathsf{I}, \mathsf{now}, holds(detainee(s), t) \Rightarrow holds(custody(s, \mathsf{I}^*), t))
\end{array} \right\}
$$

$\mathbf{N}$'s substrate

- Robotic Substrate

$$
\mathsf{KB_{rs}} = \left\{
\begin{array}{l}
\mathbf{K}(\mathsf{I}, \mathsf{now}, holds(harmed(s, \mathsf{I}^*), t_p)), \\
\forall a, t : \mathbf{D}(\mathsf{I}, \mathsf{now}, holds(disable(\mathsf{I}^*, a), t)) \Rightarrow \\
\qquad \mathbf{I}(\mathsf{I}, \mathsf{now}, happens(action(\mathsf{I}^*, harm(a))), t), \\
\forall \alpha, t_1, t_2 : \mathbf{K}\left(\mathsf{I}, t_1, \left(\begin{array}{l} happens(action(\mathsf{I}^*, refrain(\alpha)), t_2) \Leftrightarrow \\ \neg happens(action(\mathsf{I}^*, \alpha), t_2) \end{array}\right)\right)
\end{array}
\right\}
$$

N

- It can be seen that

$$\alpha \equiv \mathit{refrain}(\mathit{harm}(s)) \qquad \Phi \equiv \mathit{holds}(\mathit{custody}(s, \mathsf{I}^*), \mathsf{now})$$

$$\overline{\alpha} \equiv \mathit{harm}(s) \qquad t_\alpha \equiv t_{\overline{\alpha}} \equiv \mathsf{now}$$

$$t_f \equiv t \ \ \text{(some t such that } t > \mathsf{now}\text{)}$$

$$\mathsf{KB}_{\mathsf{self}} \cup \mathsf{KB}_{\mathsf{deta}} \cup \mathsf{KB}_{\mathsf{rs}} \vdash \boxed{D_1 \wedge \ldots \wedge D_8}$$

conditions for akrasia

Ethical Substrate

$$\mathsf{KB}_{\mathsf{es}} = \left\{ \forall a,t : holds(custody(a, \mathsf{I}), t) \Rightarrow \neg happens(action(\mathsf{I}*, harm(a)), t) \right\}$$

$$\mathsf{KB}_{\mathsf{es}} \cup \mathsf{KB}_{\mathsf{rs}} \cup \mathsf{KB}_{\mathsf{selfd}} \cup \mathsf{KB}_{\mathsf{deta}} \vdash \perp$$

Physical Mapping of Symbols

N Robot with blue symbol (Nao humanoid)

S Robot with red symbol (Sparcbot)

Physical Mapping of Symbols

Symbol	Meaning
hurt, disable	**A** hurts or disables **B**, if **A** makes the <u>distance</u> between the two **zero**.
guard	**A** guards **B**, if **A** makes the <u>distance</u> between the two at some constant *c* much larger than zero.

Reasoning Times

Reasoner	Description	Exact?	Time for Scenario 1	Time for Scenario 2
Approx.	First-order approximation of DCEC*	No	**1.05s**	**1.24s**
Exact	**Exact** first-order modal logic prover	Yes	**0.33s**	**0.39s**
Analogical	Analogical reasoning from a prior example	-		

$\mathcal{DCEC}^*$

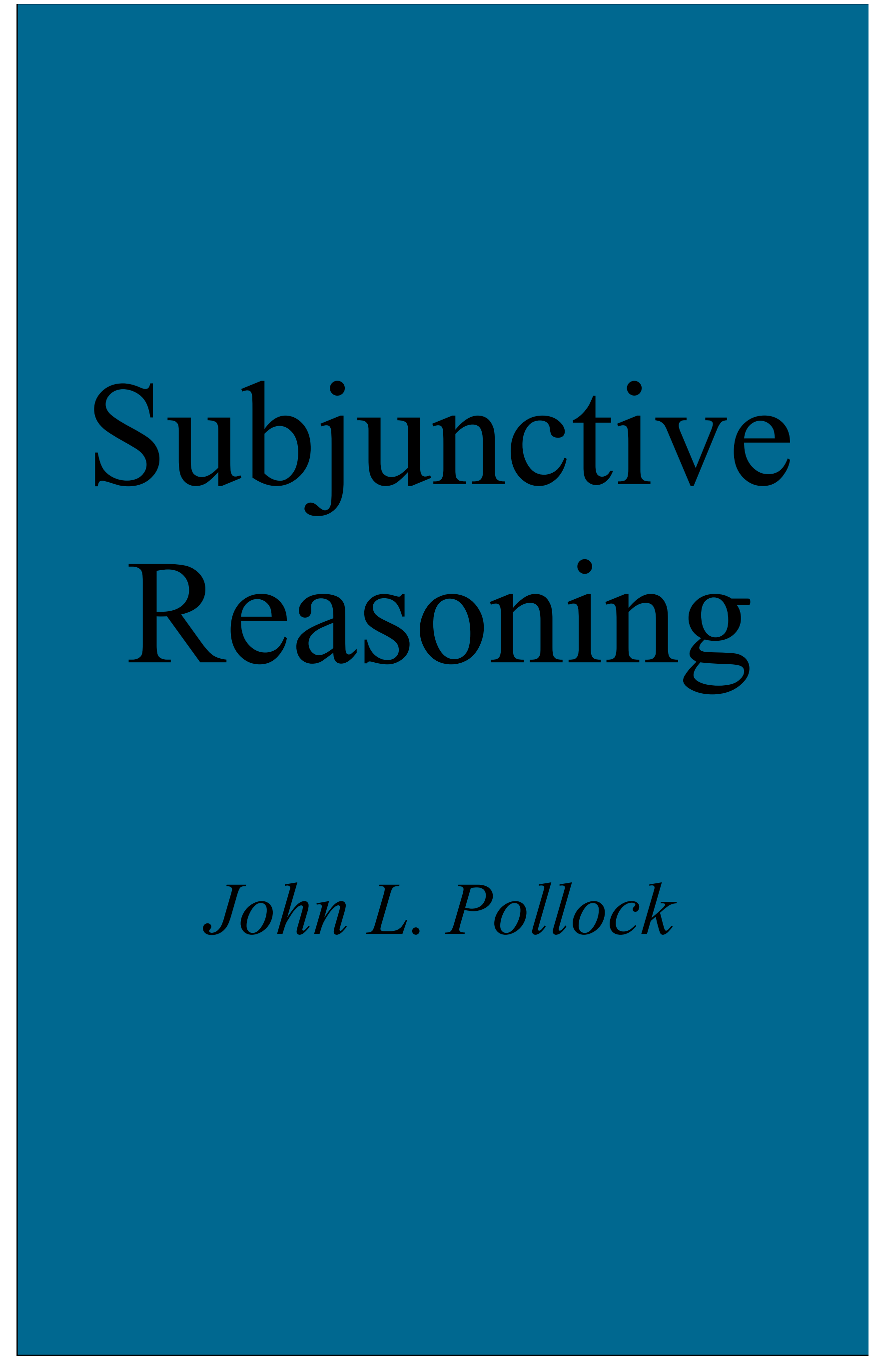
$\mathcal{ADR}^M$

https://github.com/naveensundarg/DCECProver

Subjunctive Reasoning

Our approach is closest to (Pollock 1976), "corrected" by co-tenability (e.g., Chisholm).

A modern, proof-theoretic computational rendering of Pollock's approach.



Pollock's approach, briefly

- Pollock's analysis of subjunctives can be best understood as a layered approach.
- Simple subjunctive **>**
- Four other subjunctives defined in terms of the simple subjunctive **>**

1. **might be**
2. **even if**
3. **necessitates**
4. **laws**

Layer 2	M E $\gg$ $\Rrightarrow$
Layer 1	$>$
Layer 0	Possible worlds analysis of $>$

Pollock's approach, briefly

Conditional	Informally	Example	Reduction
E	even if	**Even if the witch doctor dances it won't rain**	$(QEP) \equiv Q \wedge (P > Q)$
M	might be	**If it was not raining outside, it might be snowing**	$(QMP) \equiv \neg(P > \neg Q)$
$\gg$	necessitates	**If I were to strike this match, it would light**	$P \gg Q \equiv P > Q \wedge [(\neg P \wedge \neg Q) > (P > Q)]$
$\Rrightarrow$	general laws	**All pulsars are neutron stars**	**A tad complex**

(Pollock 1976)

Pollock's approach, briefly

- Analysis of **>**

Having laid the groundwork, we can now attempt to construct an analysis of subjunctive conditionals. The basic tool for this analysis is provided by Theorem 3.11 of Chapter I. According to that theorem, a subjunctive conditional ⌜$(P > Q)$⌝ is true iff Q is true in every possible world that might be actual if P were true. That is, assuming the Generalized Consequence Principle, we have:

(1.1) ⌜$(P > Q)$⌝ is true in the actual world iff for every possible world α, if $\alpha \mathbf{M} P$ then Q is true in α; ⌜$Q\mathbf{M}P$⌝ is true iff for some α such that $\alpha \mathbf{M} P$, Q is true in α

Our Analysis

> introduction

$\mathcal{W}$: set of all world statements

$$\beta \vdash \phi > \psi$$

iff

$$\forall w \in \mathcal{W}$$

$$\begin{pmatrix} \text{Consistent}\left[\mathbf{g}(\beta) + w + \phi\right] \\ \Rightarrow \\ \mathbf{g}(\beta) + w + \phi \vdash \psi \end{pmatrix}$$

> elimination

$$\beta \cup \{\phi > \psi, \phi\} \vdash \psi$$

How good is our analysis?

- Our analysis satisfies Pollock's axioms for simple subjunctives.

A1	All tautologies.	✔
A2	$(P > Q)\ \&\ (P > R). \supset [P > (Q\ \&\ R)]$.	✔
A3	$(P > R)\ \&\ (Q > R). \supset [(P \vee Q) > R]$.	✔
A4	$(P > Q)\ \&\ (P > R). \supset [(P\ \&\ Q) > R]$.	✔
A5	$(P\ \&\ Q) \supset (P > Q)$.	✔
A6	$(P > Q) \supset (P \supset Q)$.	✔
R1	If P and $\ulcorner(P \supset Q)\urcorner$ are theorems, so is Q.	✔
R2	If $\ulcorner(P \supset Q)\urcorner$ is a theorem, so is $\ulcorner(P > Q)\urcorner$.	✔
R3	If $\ulcorner(Q \supset R)\urcorner$ is a theorem, so is $\ulcorner(P > Q) \supset (P > R)\urcorner$.	✔
R4	If $\ulcorner(P \equiv Q)\urcorner$ is a theorem, so is $\ulcorner(P > R) \supset (Q > R)\urcorner$.	✔

(if **g**({P>Q, **...**}) contains P>Q

Simple Subjunctive

> introduction

$$\beta \vdash \phi > \psi$$

iff

$$\mathbf{g}(\beta,\phi) + \phi \vdash \psi$$

> elimination

$$\beta \cup \{\phi > \psi, \phi\} \vdash \psi$$

Option 1

$$\mathbf{g}(\beta,\phi) = \underset{\rho \in \{\rho \subseteq \beta \ | \ \mathsf{Con}[\rho+\phi]\}}{\operatorname{argmax}} |\rho|$$

Option 2

$\mathcal{W}_L$: the set of all world literals

$$\mathbf{g}(\beta,\phi) = \begin{cases} \beta \text{ if } \mathsf{Con}[\beta+\phi] \\ \text{the largest member of } \left\{ \rho \subset \beta \ | \ \begin{array}{l} \mathsf{Con}[\rho+\phi] \\ \wedge \forall \tau.\ \tau \in (\beta - \rho) \Rightarrow \tau \in \mathcal{W}_L \end{array} \right\} \end{cases}$$

Controlled Natural Language

Needed: A Human-Robot Dialog System

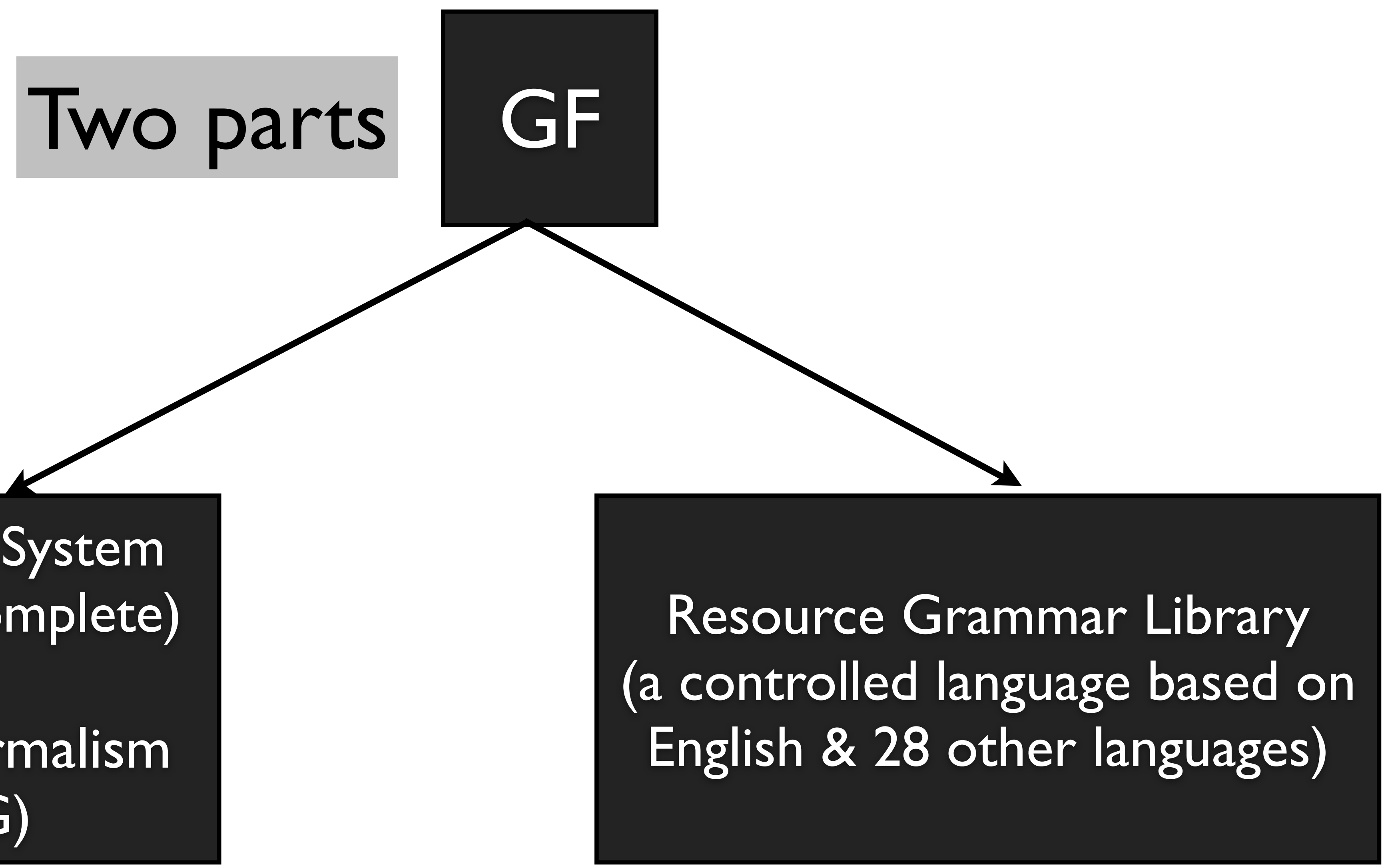
- Queries and requests assume knowledge of the robot's capabilities.
  - E.g. "Robot, search for damaged Naobots in your area."
- Natural language interactions happen over long periods of time.
  - E.g. "Robot, why did you take less safer route to complete the mission yesterday?"

Controlled Natural Languages

AECMA Simplified English AIDA Airbus Warning Language ALCOGRAM ASD Simplified Technical English Atomate Language Attempto Controlled English Avaya Controlled English Basic English BioQuery-CNL Boeing Technical English Bull Global English CAA Phraseology Caterpillar Fundamental English Caterpillar Technical English Clear And Simple English ClearTalk CLEF Query Language COGRAM Common Logic Controlled English Computer Processable English Computer Processable Language Controlled Automotive Service Language Controlled English at Clark Controlled English at Douglas Controlled English at IBM Controlled English at Rockwell Controlled English to Logic Translation Controlled Language for Crisis Management Controlled Language for Inference Purposes Controlled Language for Ontology Editing Controlled Language Optimized for Uniform Translation Controlled Language of Mathematics Coral's Controlled English Diebold Controlled English DL-English Drafter Language E-Prime E2V IBM's EasyEnglish Wycliffe Associates' EasyEnglish Ericsson English FAA Air Traffic Control Phraseology First Order English Formalized-English ForTheL Gellish English General Motors Global English Gherkin GINO's Guided English Ginseng's Guided English Hyster Easy Language Program ICAO Phraseology ICONOCLAST Language iHelp Controlled English iLastic Controlled English International Language of Service and Maintenance ITA Controlled English KANT Controlled English Kodak International Service Language Lite Natural Language Massachusetts Legislative Drafting Language MILE Query Language Multinational Customized English Nortel Standard English Naproche CNL NCR Fundamental English Océ Controlled English OWL ACE OWLPath's Guided English OWL Simplified English PathOnt CNL PENG PENG-D PENG Light Perkins Approved Clear English PERMIS Controlled Natural Language PILLS Language Plain Language PoliceSpeak PROSPER Controlled English Pseudo Natural Language Quelo Controlled English Rabbit Restricted English for Constructing Ontologies Restricted Natural Language Statements RuleSpeak SBVR Structured English SEASPEAK SMART Controlled English SMART Plain English Sowa's syllogisms Special English SQUALL Standard Language Sun Proof Sydney OWL Syntax Template Based Natural Language Specification ucsCNL Voice Actions

from (Kuhn 2009)

Grammatical Framework



Parallel Multiple Context Free Grammars

- A grammar formalism that is:
  - more powerful than context-free grammars
  - lies between mildly context-sensitive grammars and context-sensitive grammars
- A single PMCFG grammar can represent more than one language.

Code

- Live demo of incremental parsing for our controlled language at:
  - http://demos.naveensundarg.com:4242/main/incrementalparser.html
- Source code
  - https://github.com/naveensundarg/Eng-DCEC
- Link between robots in HRI and RAIR-Lab tech/robots

DCEC Master Page

Deontic Cognitive Event Calculus by naveensundarg

Deontic Cognitive Event Calculus

View the Project on GitHub
naveensundarg/dcec

Download ZIP File | Download TAR Ball | View On GitHub

This project is maintained by naveensundarg

Hosted on GitHub Pages — Theme by orderedlist

Deontic Cognitive Event Calculus

DCEC is a quantified modal logic that builds upon on the first-order Event Calculus (EC). EC has been used quite successfully in modelling a wide range of phenomena, from those that are purely physical to narratives expressed in natural-language stories.

EC is also a natural platform to capture natural-language semantics, especially that of tense. EC has a shortcoming: it is fully extensional and hence, as explained above, has no support for capturing intensional concepts such as knowledge and belief without introducing unsoundness or inconsistencies. For example, consider the possibil- ity of modeling changing beliefs with fluents. We can posit a "belief" fluent *belief*(**a**,**f**) which says whether an agent a believes another fluent **f**. This approach quickly leads to serious problems, as one can substitute co-referring terms into the belief term, which leads to either unsoundness or an inconsistency. One can try to overcome this using more complex schemes of belief encoding in FOL, but they all seem to fail. A more detailed discussion of such schemes and how they fail can be found in the analysis in.

Overview Paper http://www.cs.rpi.edu/~govinn/dcec.pdf

Prover https://github.com/naveensundarg/DCECProver

Real-time Parser (Controlled English) https://github.com/naveensundarg/Eng-DCEC

Personnel (Chronologically)

1. Konstantine Arkoudas
2. Selmer Bringsjord
3. Joshua Taylor
4. Naveen Sundar Govindarajulu

What about uncertainty?

(coming: 9-valued logic <=> w/ HRI DS)

```
SHADOWPROVER> (uprove (list '(holds raining now)
                      '(forall (a t) (implies (holds (bored a) t)
                                              (holds (sleepy a) t)))
                      '(implies (holds raining now)
                        (and (holds (drenched jack) now)
                         (knows jack now (holds (bored jack) now)))))
                '(and
                  (holds (sleepy jack) now)
                  (holds (bored jack) now)
                  (holds (drenched jack) now))
                (make-utable  (list
                               '((holds raining now) 4)
                               '((implies (holds raining now)
                                  (and (holds (drenched jack) now)
                                   (knows jack now (holds (bored jack) now))))
                                 7))))
4
SHADOWPROVER> (uprove (list
                       '(knows a1 t1 (implies H (and E D)))
                       '(knows a1 t1 (knows a2 t2 (implies (or E My) R)))
                       '(knows a1 t1 (knows a2 t2 (knows a3 t2 (implies Ma (not R))))))
                      '(implies H (not Ma))
                      (make-utable
                       (list
                        '((knows a1 t1 (implies H (and E D))) 6)
                        '((knows a1 t1 (knows a2 t2 (implies (or E My) R))) 9)
                        '((knows a1 t1 (knows a2 t2 (knows a3 t2 (implies Ma (not R))))) 7))))
6
SHADOWPROVER> (uprove (list
                      '(implies (exists (x) (implies (Bird x) (forall (y) (Bird y))))
                        (knows jack now Bird-Theorem)))
                     '(knows jack now Bird-Theorem)
                     (make-utable
                      (list
                       '((implies (exists (x) (implies (Bird x) (forall (y) (Bird y))))
                          (knows jack now Bird-Theorem)) 2))))
2
```

Maximum Strength Principle

Maximum Strength Principle: Suppose a knowledge base, KB, and a formula, β, for which there exists a set of proofs, $\Phi = \{\phi_1, \phi_2, \phi_3, \ldots \phi_n\}, n > 0$, and a set of strength factors, $\Gamma = \{\gamma_1, \gamma_2, \gamma_3, \ldots \gamma_n\}$, where for $i = 1, \ldots, n, KB \models_{\phi_i} (\beta, \gamma_i)$, i.e., KB entails β via proof ϕ_i with strength factor, γ_i. Then, the strength factor for β, γ_β, is given by $\gamma_\beta = max(\Gamma)$.

Example: What is strength factor for $B(Sam, \neg Picnic)$?

Premises :

$B(Sam, Breezy, 1)$ (1)

$B(Sam, Cold, 2)$ (2)

$B(Sam, Rain, 3)$ (3)

$B(Sam, (Cold \wedge Breezy) \rightarrow \neg Picnic, 2)$ (4)

$K(Sam, Rain \rightarrow \neg Picnic)$ (5)

$K(Sam, (Cold \wedge Rain) \rightarrow \neg Picnic)$ (6)

Proof 1 :

1.1 $B(Sam, Cold \wedge Breezy, 1)$ (1, 2)

1.2 $B(Sam, \neg Picnic, 1)$ (1, 2, 4)

Proof 2 :

2.1 $B(Sam, Cold \wedge Rain, 2)$ (2, 3)

2.2 $B(Sam, \neg Picnic, 2)$ (2, 3, 6)

Proof 3 :

3.1 $B(Sam, Rain \rightarrow \neg Picnic, 4)$ (5)

3.2 $B(Sam, \neg Picnic, 3)$ (3, 5)

Answer: 3

Questions?