Human-AI Communication for Deontic Reasoning Devops (CODORD)

Benjamin Grosof, PhD

Program Manager, Defense Sciences Office

https://www.darpa.mil/staff/dr-benjamin-grosof

https://www.darpa.mil/about-us/offices/dso

Exploratory: Automatically converting natural language to complex logic – is it possible?

Information Session

- Note: The solicitation takes precedence over everything said in this presentation. When in doubt, refer to the solicitation!!!

October 8, 2024

10:30 AM - 11:00 AM Log-in to Webcast

11:00 AM -11:05 AM Welcome Dr. Benjamin Grosof, DARPA PM

11:05 AM - 11:20 AM Defense Sciences Office (DSO) Overview Dr. Bart Russell, Deputy Director, DSO

11:20 AM - 12:00PM CODORD Disruption Opportunity Overview Dr. Benjamin Grosof, DARPA PM

Elevator Pitch

1. CODORD aims to make
   AI reasoning with high assurance about obligation and permission
   become widely practical in cyber systems for the first time,
   by combining advanced logic with machine learning.

2. Such reasoning is important in defense for complying with
   commander’s intents, regulations, ethics, laws, operational policies,
   directives, supply chain contracts, and international agreements.

10:30 AM - 11:00 AM Log-in to Webcast

11:00 AM -11:05 AM Welcome Dr. Benjamin Grosof, DARPA PM

11:05 AM - 11:20 AM Defense Sciences Office (DSO) Overview Dr. Bart Russell, Deputy Director, DSO

11:20 AM - 12:00PM CODORD Disruption Opportunity Overview Dr. Benjamin Grosof, DARPA PM

Program Solicitation Has the Final Say

If there is any discrepancy between what is presented today and the program solicitation, the program solicitation takes precedence.

Logic and Deontics

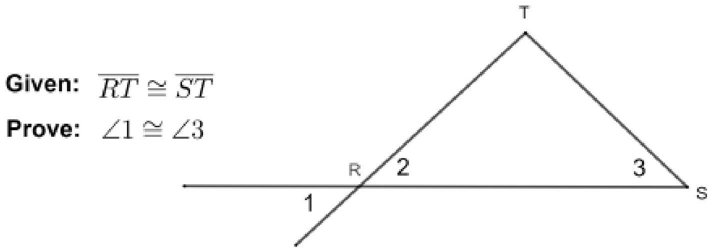
Logical reasoning infers conclusions from assertions, based on principles

- Provides ***assurance*** in reasoning
  - Inherent **verifiability**, extreme **correctness**
  - Unachievable with Machine Learning techniques (e.g., [Zhang+ ’22])

Human domain requires ***deontic*** *reasoning*

Increasing **expressive complexity** of logic & reasoning

Classical mathematical logic



Strict, simple notion of truth

Multi-agent, multi-*modal*

“Poland *believes* that USA *intends* to resist Russia’s *objective* to conquer Ukraine.”

Belief, Objective, Intention

[Genesereth+ ’87] [Chen+ ’93]

Defeasible

“H if P, *unless* there’s a stronger counterargument.”

[Wan+ ’09] [Wan ’15]

Deontic

“A is *obliged* to obey orders from B, if B is A’s superior officer.”

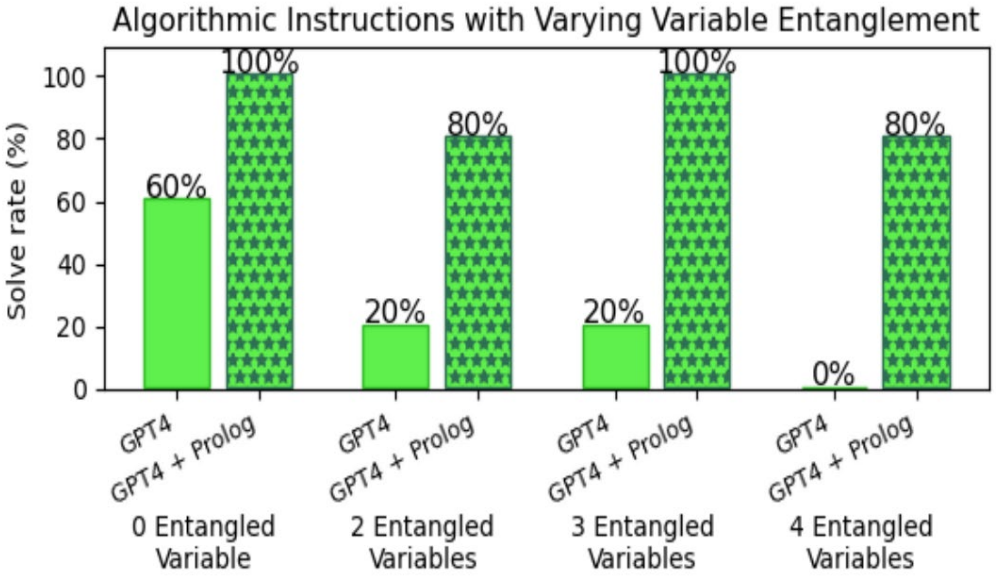
Obligation, Permission, Prohibition

[Genesereth+ ’87] Genesereth, Michael, et al, *Logical Foundations of Artificial Intelligence.*
[Chen+ ’93] Chen, Weidong, et al, HiLog: A Foundation for Higher-Order Logic Programming, *J. Logic Programming* 15(3):187-230.
[Zhang+ ’22] Zhang, Honghua, et al, On the Paradox of Learning to Reason from Data, *arxiv.*
[Wan+ ’09] Wan, Hui, et al, Logic Programming with Defaults and Argumentation Theories, *Proc. 28th Intl. Conf. on Logic Programming.*
[Wan+ ’15] Wan, Hui, et al, Defeasibility in Answer Set Programs via Argumentation Theories, *Semantic Web* 6(1): 81-98.

CODORD Objective: Deontic Reasoning that's Both Highly Assured and Highly Cost-Efficient

	Automated reasoning (e.g., recently available [1])	**LLM** (& ML)	**CODORD**
Extremely high correctness (>>99%)	✔	✘ [2]	✔
Explicit, verifiable logical explanation	✔	✘	✔
Rapid development	✘	✔	✔ MVP
No specialized expertise required to use	✘	✔	✔ MVP

- Key MVP step towards cost-efficient:
  - Automatic-from-NL logic generation
  - … with high assurance on focal implications



[2] [Borazjanizadeh+ '24] Borazjanizadeh, Nasim, et al, Reliable Reasoning Beyond Natural Language, arxiv.

[1] [Kifer+ '23] Kifer, Michael, et al, ErgoAI software, manuals & tutorials, https://github.com/ErgoAI ; also see on later slide (slide ~13) references related to ErgoAI, Rulelog, and explanation, including: [Grosof+ '10] [Andersen+ '13] [Grosof+ '13] [Swift '14] [Swift+ '22] [Swift '23] [Grosof+ '23].

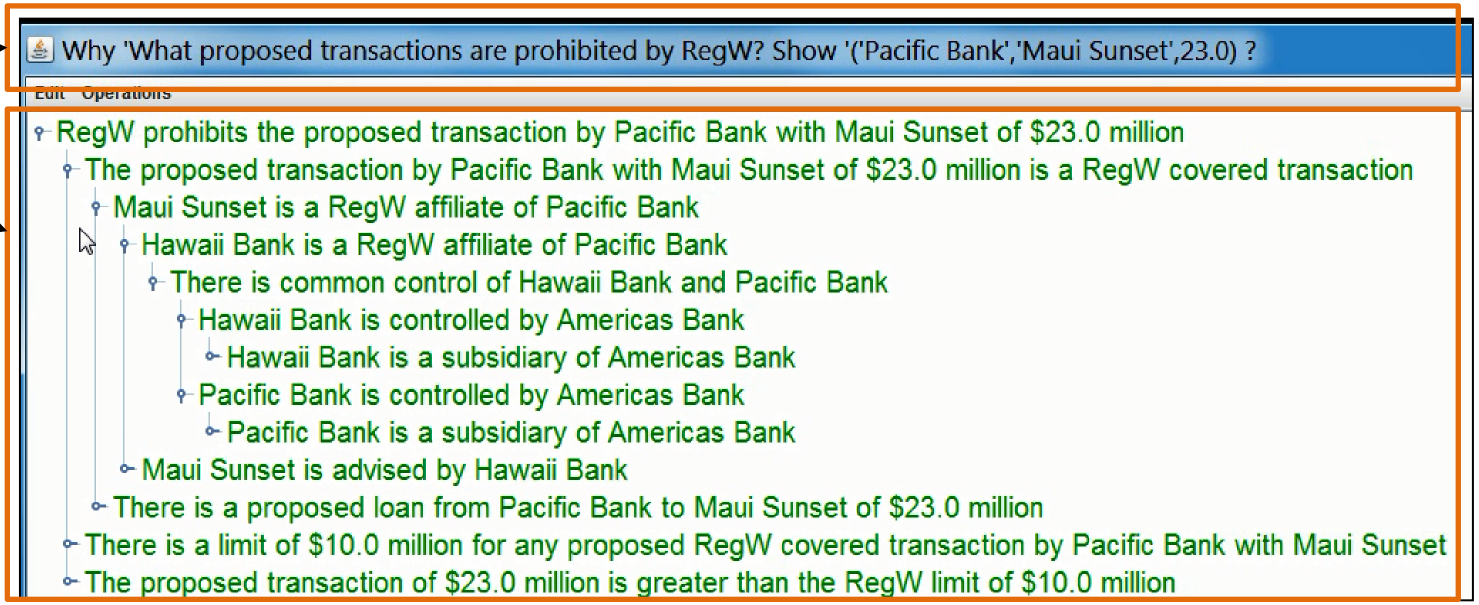
MVP: DARPA terminology for "Minimal Viable Program"
LLM: Large Language Model, e.g., GPT-4
ML: Machine Learning, e.g., Neural Networks (NN) which are the basis for LLM's
NL: Natural Language, e.g., English
Prolog = a logic programming language, ancestor of Rulelog, roughly a subset of Rulelog and of ErgoAI

The Power of Deontic Reasoning: Banking and COA

Federal Reserve Regulation W
compliance pilot
[RuleML-2015 industry track]

Query

Explanation



- Assurance & audit trail
- ~1,000x faster, ~1,000,000x cheaper, vs. manual (in marginal costs)

Similar needs for DoD→

"Order 91748115 (revocable) is that Col. Smith grants permission to Maj. Kinjay to air drop equipment packages, if operation Blue Falcon is greenlighted."

2015 example is still SOA, due to lack of progress on knowledge authoring (KA)

[RuleML-2015 industry track] demonstration and paper: Grosof, Benjamin, et al, Automated Decision Support for Financial Regulatory/Policy Compliance, using Textual Rulelog, *9th Intl. Web Rule Symposium (RuleML-2015).*

KA: Knowledge Authoring
COA: Course of Action planning

Decision Support Demands Assurance and Deontics

COA planning

"How to fulfill orders from higher, joint, coalition?"

Autonomy in weapon systems

"Which system actions are ethical?"

Navigating international agreements

"What transportation logistic plans comply?"

DoD Need

- Assurance is a growing need, across many crucial scenarios, in decision support
- Decisions (about actions) must be based on beliefs, objectives, and *deontics*
- Highly expressive logical reasoning is required
  - Multiple agents with differing levels & scopes of authority; exceptions exist for most rules

SOA

- Existing logical languages & toolsets (e.g., Rulelog/ErgoAI) enable deontic logical reasoning
  - Highly expressive, yet computationally scalable
- Has resisted practical wide deployment for decades, because NL's logical semantics is far from solved (not addressed by LLMs)

Time cost of developing logic spec that captures domain knowledge ("Knowledge Authoring (KA) bottleneck") limits scalability

COA: Course Of Action planning
NL: Natural Language, e.g., English
LLM: Large Language Model, e.g., GPT-4
KA: Knowledge Authoring
spec = specification, as set of statements

SOA: State Of the Art
DoD: (US) Department of Defense

Image credits:
https://nara.getarchive.net/media/medium-close-shot-lto-r-flight-crew-cpt-joseph-speight-nav-1lt-dave-jesurun-0cebc7
https://www.flickr.com/photos/joncutrer/43252568250
https://nara.getarchive.net/media/three-unmanned-aerial-systems-uas-soar-in-the-sky-99dd32
https://commons.wikimedia.org/wiki/File:Main_maritime_shipping_routes.png
https://www.pexels.com/photo/birds-eye-view-photo-of-freight-containers-2226458/

CODORD: Alleviate Bottleneck in KA Process for Deontic Reasoning

SOA

Existing:
- *Logic*
- *Engine*

Often $1000s per sentence

AI performs logical reasoning

e.g., Rulelog

[Andersen+ '13] [Kifer+ '23] [Grosof+ '23]

KE authors: generates, revises

KE learns domain from SME

CODORD

Enable KEs to start from NL, and SMEs to contribute directly

KE revises

AI performs logical reasoning, **generates**

KE or SME inputs NL

LONG-TERM VISION

AI performs logical reasoning, **generates, revises**

SME inputs NL

KE: Knowledge Engineer, a logician & programmer – with years of training – expert in KA
SME: Subject Matter Expert, typically lacking logic expertise and KE skill
KA: Knowledge Authoring. NL: Natural Language
Generate = generate logic code starting from Natural Language
Revise = iteratively reach shared understanding via test+edit, and debug
Rulelog = a highly expressive logic – SOA for deontic reasoning

[Andersen+ '13] Andersen, Carl, et al, Advanced Knowledge Base Debugging for Rulelog, *Proc. 7th Intl. Web Rule Symposium (RuleML-2013).*
[Kifer+ '23] Kifer, Michael, et al, ErgoAI software, manuals, & tutorials, https://github.com/ErgoAI .
[Grosof+ '23] Grosof, Benjamin, et al, Ergo: A Quest for Declarativity in Logic Programming, in: Warren, David S., et al (eds.), *Prolog: The Next 50 Years.*

Technical Challenges and Hypothesis

Hypothesis: We can enable humans to do much of the Knowledge Authoring while speaking only in Natural Language (NL), and using automation

Cannot rely on LLMs for reasoning. Rather, use LLMs for their strengths – in language translation.

Technical Challenges in KA limiting SOA practicality

1. Lack Automation: Even KEs must *manually* author into logic
   - Logical semantics of NL is far from solved in general
2. Lack Accessibility: SMEs *cannot* directly author into logic
   - Logic expertise is needed to encode expressive K
3. Iterativity: Even KEs cannot accomplish authoring *in one round*
   - *Revising:* Contextualization is complex, requires test+edit
     - To reach shared understanding: Agreement on focal implications

Example logical formulation (in Rulelog [Kifer+ '23]):

```
@!{order_91748115} @{default}
  Permission[grantor->'Col. Smith', actor->'Maj. Kinjay',
  action->${'air drop'[obj->?ep, location_descr->?place]}]
  :-
  'Blue Falcon':'assault operation'[proceed->green],
  ?ep:'equipment package', in(?place,`map sector'(578306)).
```

CODORD: Automate *Generating* (NL to logic)

CODORD: Re-run *Generating* within *Revising*

SOA: ~ $\$10^2$ – $\$10^4$ (est.) per assertion, and days – months elapsed, for debugged knowledge; depending largely on expressive complexity of the assertion

Industry experience
Little/no academic research

NL: Natural Language
KA: Knowledge Authoring
LLM: Large Language Model
KE: Knowledge Engineer
K: Knowledge. edit = add/modify/delete

[Kifer+ '23] Kifer, Michael, et al, ErgoAI software, manuals & tutorials, https://github.com/ErgoAI

Arrow Key: = addresses Challenge

Technical Insight: Automatic *Generating* (NL to logic) – How It's Possible

- Open hard problem: Generate high-expressiveness **logic** code (e.g., Rulelog or other extended logic programs)

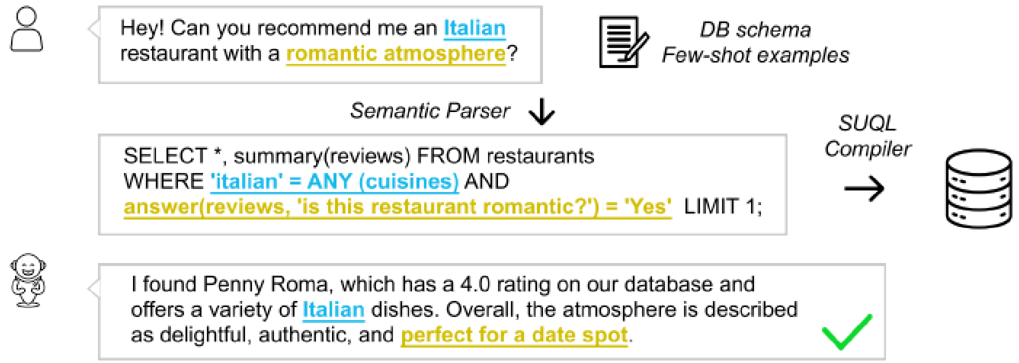
- Can leverage recent Machine Learning (ML) advances in techniques for code generation from NL
  - Via LLM into imperative programming code (e.g., [Agarwal+ '24]) ; **1.5x - 2.0x productivity gains** (McKinsey study '23)
  - Via semantic parsing, combined with LLM/NN, into formulas incl. logic
    - Query formulation for databases – subset of high-expressiveness logic programs (e.g., [Liu+ '24] – 90.3% accuracy on a Yelp task)
    - Can enhance semantic parsing for higher expressiveness (e.g., [Bao+ '24] – 90.2 - 95.3% accuracy on a logic task suite)
  - Can create synthetic training data via: Logic-to-NL generation, + diverse rephrasings (e.g., [Maini+ '24] – **5x less real data required**)

- Can choose coding style conventions in: logic; semantic parsing

Comparison: logic KA, vs. Imperative coding assistant

harder in logic KA	**opportunities**
complex contextualization	no procedural state
far less existing training data	create synthetic
SMEs don't speak coding language	SMEs are familiar with reasoning

Example of query formulation for database [Liu+ '24]



[Agarwal+ '24] Agarwal, Anisha, et al, Copilot Evaluation Harness: Evaluating LLM-Guided Software Programming, *arxiv.*
[Maini+ '24] Maini, Pratyush, et al, Rephrasing the Web: A Recipe for Compute and Data-Efficient Language Modeling, *arxiv.*

ML: Machine Learning
NL: Natural Language
LLM: Large Language Model
NN: Neural Networks
DB: database
Imperative programming – language examples: Python, C, Javascript, C++, C#, Java, Fortran, Visual Basic

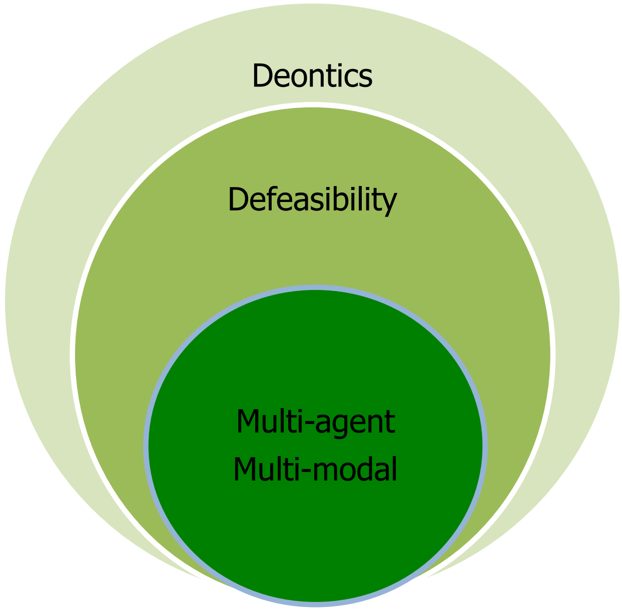
[Liu+ '24] Liu, Shicheng, et al, SUQL: Conversational Search over Structured and Unstructured Data with Large Language Models, *arxiv.*
[Bao+ '24] Bao, Qiming, et al, Abstract Meaning Representation-Based Logic-Driven Data Augmentation for Logical Reasoning, *arxiv.*
(McKinsey study '23) https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai#/

Why Do We (CODORD) Think We'll Succeed, When Others Have Failed?

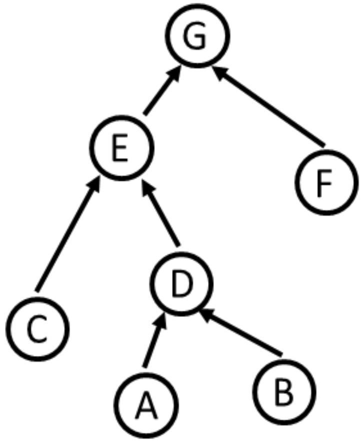
1. Recent advances in NN-based NLP, incl. LLMs: competence, incl. integration & breadth (e.g., [Yang+ '24])

2. Focus on target logic that meets 4 practical requirements
   a. Sufficient <u>expressiveness</u>; concisely [Kifer+ '23]
   b. <u>Scalable</u> computationally [Grosof+ '13] [Swift '14]
   c. Strong <u>explainability</u> [Andersen+ '13]
   d. Commercial-quality open-source <u>toolset</u> recently available [Kifer+ '23] [Swift+ '23]

3. Focus on deontics – for decision support
   - In contrast to general semantic NL Understanding, which has hugely broad diffuse scope



Expressiveness tiers



<u>Assertions</u>
E if C and D
D if A and B
G if E and F

Recent tooling exposes logical dependency structure in K

[Yang+ '24] Yang, Jingfeng, et al, Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond, *ACM Transactions on Knowledge Discovery from Data* 18.6: 1-32
[Kifer+ '23] Kifer, Michael, et al, ErgoAI software, manuals & tutorials, https://github.com/ErgoAI ; also see:
[Swift+ '22] Swift, Theresa, et al, XSB software & manuals, https://xsb.sourceforget.net .
[Grosof+ '23] Grosof, Benjamin, et al, Ergo: A Quest for Declarativity in Logic Programming, in: Warren, David S., et al (eds.), *Prolog: The Next 50 Years.*
[Grosof+ '13] Grosof, Benjamin, et al, Radial Restraint: A Semantically Clean Approach to Bounded Rationality, *Proc. AAAI Conference on Artificial Intelligence,* Vol. 27, No. 1.
[Swift '14] Swift, Theresa, Incremental Tabling in Support of Knowledge Representation and Reasoning, *Theory and Practice of Logic Programming* 14(4-5):553-567.
[Andersen+ '13] Andersen, Carl, et al, Advanced Knowledge Base Debugging for Rulelog, *Proc. 7th Intl. Web Rule Symposium (RuleML-2013)* ; also see:
[Grosof+ '10] Grosof, Benjamin, et al, A SILK Graphical UI for Defeasible Reasoning, with a Biology Causal Process Example, *9th Intl. Semantic Web Conference.*
[Swift+ '23] Swift, Theresa, et al, The Janus System: Multi-Paradigm Programming in Prolog and Python, *arxiv.*

NN: Neural Networks
NLP: Natural Language Processing
LLM: Large Language Model
NL: Natural Language
K: Knowledge. Modal – examples: Belief, Intention, Obligation, Permission

Program Design

DEFENSE SCIENCES OFFICE

Objective: Alleviate KA Bottleneck for deontic reasoning

Hypothesis: We can enable humans to do much of the Knowledge Authoring while speaking only in **Natural Language** (NL), and using **automation**, with high assurance

- **MVP Program Scoping:** Show ***feasibility***, for deterministic (defer probabilistic)
  - **Automation from NL**: manual → automated – *fully* for Generating (*partially* for Revising/Overall)

    spec in logic language only → spec in NL

Performer task:

- Automate generating from NL: into Logic
  - & the reverse direction (easier)

Create and deliver a novel KA technique:
- <u>Software</u>
- <u>ML models</u>
- <u>Methodology guidance</u>
- <u>Insights/rationale for design of approach</u>

T&E tasks:

- Create programmatic <u>use/test cases</u>
  - Multiple test problems per use case
- Create <u>T&E protocol for KA</u>
- Execute KA for <u>evaluations</u>
  - Operate KE teams
  - Measure SOA too – required
- Develop & support <u>test framework</u> software for all performers
  - Common <u>manual KA</u>, data gathering

- Collaboration among Performers and T&E to develop:
  - <u>Training examples for ML</u>
  - Style conventions in logic, NL
  - <u>Test cases</u>
  - Best practices
  - <u>Novel measures & insights</u>
    - Exploring the space

<u>Underlining</u> indicates program outputs

KA: Knowledge Authoring
NL: Natural Language
from/in NL = NL plus graphical interaction
ML: Machine Learning. spec: specification

Metrics and Evaluation: Comparing Performer New KA Approach to SOA

Evaluation Protocol

Performer tasks:
1. Train T&E in their new KA approach
   - Software + recommended methodology

T&E Compares head-to-head, for each use case:
1. SOA KA (manual *Generating*)
2. each performer's KA approach (automated *Generating*)

T&E explores additional measures

Metric for performer KA approach	Phase 1 (12 months)	Phase 2 (12 months)
• Assurance*: on focal implications		
• Correctness of answers to focal queries, compared to SOA KA for each use case	(1 – Correctness) is within 3x of SOA KA	Correctness ≥ SOA KA
• Cost-Efficient: on *Generating*		
• Automatic-from-NL logic generation, as percentage of: logic sentences after *Revising*	40%** of: logic sentences after *Revising*	80% of: logic sentences after *Revising*
• Total KA labor time (including *Revising*), compared to SOA for each use case	Total KA Labor Time ≤ 4x*** SOA KA	Total KA Labor Time ≤ 2x SOA KA

* Observation: The automated reasoning is fully verifiable
** Observation: SOA is 0%
*** Goal of program is feasibility, not yet optimality

KE: Knowledge Engineer (expert in logic, reasoning, and KA)
KA: Knowledge Authoring
NL: Natural Language

Program Schedule & Milestones

DARPA · DEFENSE SCIENCES OFFICE

	Phase 1				Phase 2			
Program Year	1				2			
Quarter	1	2	3	4	1	2	3	4

Prep initial:
- Use cases
- Test framework

T&E develops and supports: Use/test cases, test framework

T&E develops T&E protocol for KA, and executes KA for evaluations

Performers develop approaches to automate *Generating*

△ = Progress milestone (report, demo)

▲ = Evaluation by T&E, across 3+ use cases

* non-fact
- Additionally, there are:
  - fact sentences
  - background non-fact sentences

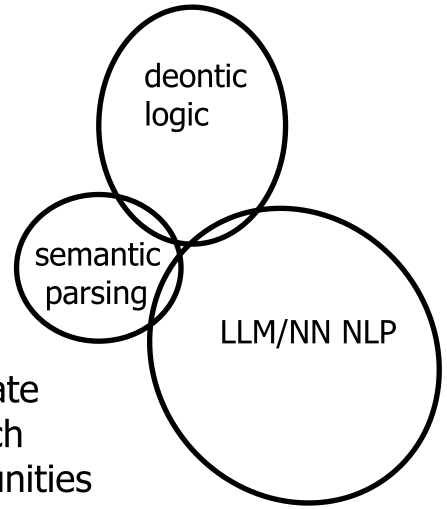
Month 9	Phase 1 end Month 12	Month 18	Phase 2 final eval Month 21
3-10* NL foreground sentences	10-30* NL foreground sentences	20-60* NL foreground sentences	30-100* NL foreground sentences
≤ 5x SOA incorrectness	≤ 3x SOA incorrectness	≤ 2x SOA incorrectness	≥ SOA correctness
≥ 20% automatic from NL	≥ 40% automatic from NL	≥ 60% automatic from NL	≥ 80% automatic from NL
≤ 5x SOA total KA labor time	≤ 4x SOA total KA labor time	≤ 3x SOA total KA labor time	≤ 2x SOA total KA labor time

Fact = relatively simple form of logic sentence, lacking an if-then connective
Incorrectness = 1 – Correctness
Foreground = in the explicit focus of domain/use-case/test-problem description and of SME spec in NL

Non-fact = a more complex form of logic sentence, containing an if-then connective.
Background = non-foreground; often identified as needed, then authored, during *Revising*

- Paradigm shift, bend AI trajectory for complex K back towards logic too (“*reasoning and* learning”)
  - High assurance
  - Huge realm of applications value for deontic and similar high-expressive reasoning

- Expertise in deontic reasoning and semantic parsing is dispersed and disparate



Disparate research communities

K: Knowledge
NL: Natural Language
NLP: Natural Language Processing
LLM: Large Language Model
NN: Neural Nets

CODORD contracting and proposing: mechanics/info (I)

- CODORD is a Disruption Opportunity (DO): https://www.darpa.mil/work-with-us/disruptioneering

- DO's are solicited under a common program announcement: https://sam.gov/opp/cb7a935d59bb4ceeb62b9515f7d9f9b0/view

- DO awards are Other Transactions (OT's)
  - What are OT's?: https://acquisitioninnovation.darpa.mil/what-are-ots

- Expect the CODORD DO **solicitation** within the next few days or weeks (if it's not there already): on SAM.gov
  - https://sam.gov/content/home

- Note: The solicitation takes precedence over everything said in this presentation. When in doubt, refer to the solicitation!!!

CODORD contracting and proposing: mechanics/info (II)

- DARPAConnect: how to work with DARPA, e.g., doing business, process, and resources
  - https://www.darpa.mil/work-with-us/darpaconnect

- CODORD Resource (web) page: find it on the DARPA.mil website

- Explainer videos, e.g., on deontics: find on the DARPA.mil website, e.g., in news –ish
  - We aim to link that from the Resource page

- CODORD FAQ
  - We aim to link that from the Resource page
  - Q&A protocol: potential-proposer Qs are answered here (perhaps after aggregating or restating Q's), with A's being available to all (for maximum fairness), NOT answered only to the individual questioner

- CODORD email address: CODORD@darpa.mil
  - Appropriate for submitting questions (Q's) about the proposal process, clarifying the solicitation, etc.
    - But check the above resources first, please (including on the last slide)
  - NOT appropriate for submitting questions about your technical approach!!
  - Please use this rather than emailing individuals; this is a mailbox shared by the DARPA CODORD team

CODORD proposal process schedule: key dates

- Day 0: CODORD solicitation: Program Announcement (analogous to a BAA) released on SAM.gov

- Day ~11: Teaming profiles due by 1pm EDT
  - The Special Notice about CODORD (find it on SAM.gov) gives info on teaming profiles
    - Also look for that info on the CODORD Information Session's associated Resource Page for CODORD, on the DARPA.mil website
  - (Day ~11: DARPA sends the batch of teaming profiles to those who submitted one)

- Day ~25: proposal Abstracts due
  - Submission of an Abstract is optional, not required, but is recommended

- Day ~32: DARPA provides feedback on abstracts incl. to encourage/discourage submission of full proposal, and often also technical questions/requests for points to cover in a full proposal
  - (If you receive discouraging feedback, it might not be worth your while to prepare & submit a full proposal)

- Day 60: Full proposals due

- For more precise dates, and other details, refer to the solicitation!!

Phase 1 Proposal Abstract* Preparation – Expected Information

Section Headings	Required Content
Page limit -- total	**4 pages**
Cover Sheet (Counted towards the 4-page limit.)	Proposer Name, Title, Date, E-Mail Addresses, Phone Numbers, and Addresses for Technical Point of Contact and Administrative Point of Contact. [See the solicitation ("Program Announcement") for organizational conflict of interest information]
Technical Content on your proposed approach (No more than 2 pages, and is counted towards the total page limit)	Provide a summary of the following: • Your technical vision to achieve the goals of this program • Overall technical approach to meet the goals and milestones of Phases 1 and 2 • Technical expertise of your performer team, described briefly, including links to bios/CVs • Less than half a page, on why the proposer believes their team can be successful at achieving program goals if selected to participate in CODORD. The proposer may include past experience, organizational capabilities, team members' qualifications, or anything else that demonstrates competence in logical reasoning, knowledge authoring, machine learning, and natural language processing, e.g., large language models.
References (No longer than 1 page, and is counted towards the 4-page limit, although is excluded from the 2-page limit for the Technical Content)	Provide a list of citations, references, or end notes.

* Submitting an abstract is optional (see last slide)

Back-up slides

Back-up: Logic Style Convention: Example of Alternatives

Context:

Technical Challenges in KA limiting SOA practicality

- Logical semantics of NL is far from solved in general
- Many-many mapping of phrasing ←→ formulation

Example logical formulation in Rulelog; it uses "frame syntax":

```
@!{order_91748115} @{default}
  Permission[grantor->'Col. Smith',
             actor->'Maj. Kinjay',
             action->${'air drop'[obj->?ep,
                                  location_descr->?place]}]
  :-
  'Blue Falcon':'assault operation'[proceed->green],
  ?ep:'equipment package',
  in(?place, `map sector'(578306)).
```

Alternative formulation (one among many) in Rulelog; it uses "predicate syntax":

```
@!{order_91748115} @{`Col. allows Blue Falcon air drop'}
  Permits(action)(agent('Maj. Kinjay'),
                  by('Col. Smith'),
                  ${'air drop'(object, location)(?ep, ?p)}
  :-
  `assault operation'('Blue Falcon'),
  proceed('Blue Falcon', status(green)),
  'equipment package'(?ep),
  `in place'(`map sector', ?p, 578306).
```

KA: Knowledge Authoring
NL: Natural Language

Rulelog syntax is cf. ErgoAI [Kifer+ '23]

[Kifer+ '23] Kifer, Michael, et al, ErgoAI software, manuals & tutorials, https://github.com/ErgoAI

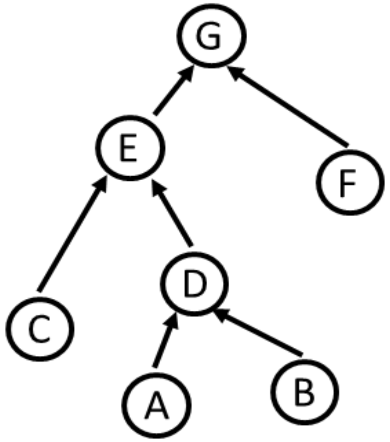
Back-up:
More about KA Process downstream of *Generating*: *Revising*

- **What**: Agree in implications, reach shared human-machine understanding
- **How**: KEs do iterative test+edit, in the Reasoner's IDE, via command line and visual interface
  - Plus: Re-run *Generating*
  - Inspect: Explanations, logical dependency structure in K,
    intermediate sub-queries & conclusions ("tables"), sizes & CPU times
  - Compose/organize the set of K, via: Hierarchical modularity; scaffolding of tests
  - Activities: Create tests. Find knowledge gaps, then specify additional/background K. E.g., via: run *Generating.*
  - **Overall: Heavily Manual**

Can expose to the human: the machine's understanding
via NL representation of: logical dependency structure in K (e.g., [Kifer+ '23])

Example

Assertions
E if C and D
D if A and B
G if E and F



KA: Knowledge Authoring
IDE: Integrated Development Environment (command-line and graphical)
K: Knowledge
NLP: Natural Language Processing

[Kifer+ '23] Kifer, Michael, et al, ErgoAI software, manuals & tutorials, https://github.com/ErgoAI ;
also see: on earlier slide (slide ~13) references related to ErgoAI, Rulelog, and explanation; as well as:
[Ullman '88] Ullman, Jeffrey, *Principles of Database and Knowledge-Based Systems.* (2 volumes.)
[Przymusinski '94] Well Founded and Stationary Models of Logic Programs,
Annals of Mathematics and Artificial Intelligence 12:141-187.

Back-up: More about Applications

<u>Examples</u> of deontic assertions knowledge:
orders, laws, regulations, directives, doctrines, ethics, treaties, agreements, contracts, operational policies

- Potential to open gate to huge realm of applications value in military & commercial
  - *Where deontic reasoning, and/or similar high-expressiveness reasoning, is crucial*
  - Operations planning, policies, & execution; confidentiality; ethical/legal compliance; M&S, systems integration, wargaming; autonomy
  - Supply chain & financial/contracting; health care treatment guidance & insurance
  - *Compliance* in the above

NL: Natural Language
HCI: Human-Computer Interaction
M&S: Modeling & Simulation

DARPA

www.darpa.mil