Tackling Benchmark Problems of Commonsense Reasoning

Ulrich Furbach[1], Andrew S. Gordon[2], and Claudia Schon[1]*

[1] Universität Koblenz-Landau, {uli,schon}@uni-koblenz.de
[2] University of Southern California, gordon@ict.usc.edu

Abstract. There is increasing interest in the field of automated commonsense reasoning to find real world benchmarks to challenge and to further develop reasoning systems. One interesting example is the Triangle Choice of Plausible Alternatives (Triangle-COPA), which is a set of problems presented in first-order logic. The setting of these problems stems from the famous Heider-Simmel film used in early experiments in social psychology. This paper illustrates with two logical approaches—abductive logic programming and deonitc logic—how these problems can be solved. Furthermore, we propose an idea of how to use background knowledge to support the reasoning process.

1 Introduction

In his influential 1958 paper, entitled "Programs with Common Sense" [14], John McCarthy set in motion his research agenda for Artificial Intelligence. He proposed the use of logic and deduction to overcome the difficult challenges of commonsense reasoning. His own pursuits led him later introduce the logic of circumscription [15], to handle the non-monotonic nature of human inference. In the intervening decades, numerous other approaches have been proposed by different researchers, e.g. based on probability theory or on argumentation frameworks. Progress on varied approaches was recently demonstrated, in dramatic fashion, in the success of IBM's Watson system in the Jeopardy challenge [3]. Subsequently, there has been considerable effort to investigate the varied techniques of the Watson system as a new programming paradigm, *cognitive computing*, and apply these techniques to diverse research and commercial problems, including eHealth, cancer research, and even supporting culinary chefs.

Although the Jeopardy challenge served to demonstrate the potential of new technologies, it does not provide the most appropriate benchmark problems for testing and evaluating individual research methods and approaches. Watson's success required a large engineering team, integrating technologies across many different fields of computer science. Logic-based approaches to commonsense reasoning may increasingly play a role in future cognitive programming applications, but the Jeopardy challenge is too ambitious as a tool for benchmarking
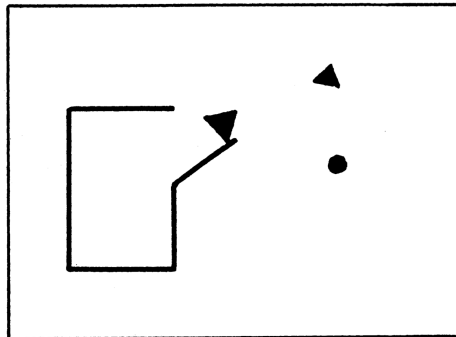
Fig. 1: One frame of the film used by Heider and Simmel in their study.

progress in this area. Over the years, logic-based approaches have been slow to move beyond the ubiquitous Tweety and Emu example problems to demonstrate their usefulness, although specialized benchmarking suites are increasingly being used in sub-disciplines of automated reasoning, e.g. in first-order theorem proving, answer set programming, and SAT solving. Recently, new sets of benchmark problems have been proposed for commonsense reasoning, such as the Winograd Schema Challenge [12] and the Choice Of Plausible Alternatives challenge [21]. Both of these challenges, however, require substantial capabilities for handling natural language (English), which complicates their use by researchers hoping to focus specifically on logic-based reasoning approaches.

The Triangle-COPA challenge[1] [13] provides a suite of one hundred logic-based commonsense reasoning problems, and was developed specifically for the purpose of advancing new logical reasoning approaches. Based on an influential psychology experiment from the 1940's, Triangle-COPA serves as a useful tool for studying the differences between human and logical reasoning. In the sections that follow, we describe the Triangle-COPA challenge problems and demonstrate that they can be solved using very different approaches to automated logical reasoning—first using a probabilistic form of logical abduction, and second using deontic logic—and discuss the challenges of authoring or acquiring the necessary background knowledge.

2 The Triangle-COPA Benchmarks

In an early and influential study of human social perception, psychologists Fritz Heider and Marianne Simmel [7] presented subjects with a short animated film depicting the movements of two triangles and a circle in and around a box with a hinged opening (Figure 1). Asked what they saw in the film, subjects each responded with similar narratives that anthropomorphized the moving shapes as intentional characters with beliefs, goals, and emotions. The simplicity of the film

---

[1] Available at https://github.com/asgordon/TriangleCOPA/

was in sharp contrast with the richness of the subjects' narratives, highlighting the role of knowledge and personal experience in the process of interpretation. Heider [6] later argued that the interpretation of intentional behavior was driven by commonsense theories of psychology and sociology, and was the basis of human social interaction.

How could we build a software system that was capable of interpreting the Heider-Simmel film in the same manner as the study's subjects? Researchers in artificial intelligence and cognitive science have sought to construct such a system. Thibadeau [23] takes a symbolic approach, representing the coordinates of each object in each frame of original film, which are matched to defined action schemas, such as opening the door or going outside the box. Pautler et al. [18] follows a related approach, beginning with object trajectory information from an animated recreation of the Heider-Simmel film. An incremental chart parsing algorithm with a hand-authored action grammar is then applied to recognize character actions as well as their intentions.

These earlier attempts highlight several problems for the use of the original Heider-Simmel film as a challenge problem by automated reasoning researchers. First, any system must overcome the difficult challenge of recognizing actions in the visual scenes, e.g. by first extracting quantitative trajectory information from the image data. Contemporary gesture recognition methods may be suitable for this task, using models trained on copious amounts of annotated examples. However, the effort involved in apply these techniques shifts research attention away from the central automated reasoning task of interpretation. Second, the original Heider-Simmel film provides a compelling input as a challenge problem, but the correct output is unspecified. Precisely because the input is "open to interpretation" is it difficult to compare the relative performance of two competing approaches, or even of the same approach as it develops over time.

The Triangle Choice of Plausible Alternatives (Triangle-COPA) set of one hundred challenge problems is a recent attempt to overcome these two problems with the original Heider-Simmel movie [13]. Each of the one hundred questions in this problem set describes, in English and in first order logic, a short sequence of events involving the characters of the original Heider-Simmel film: two triangles and a circle moving around a box with a hinged opening. This description ends with a question that requires the interpretation of the action sequence, and provides a choice of two possible answers, also in both English and logical form. The task is to select which of the two options would be selected by a human, where the correctness of the choice has been validated by teams of human volunteers. Three examples of Triangle-COPA questions are as follows:

44: The triangle opened the door, stepped outside and started to shake. Why did the triangle start to shake?
   `(and (exit' E1 LT) (shake' E2 LT) (seq E1 E2))`
 a. The triangle is upset.
   `(unhappy' e3 LT)`
 b. The triangle is cold.
   `(cold' e4 LT)`

58: A circle and a triangle are in the house and are arguing. The circle punches the triangle. The triangle runs out of the house. Why does the triangle leave the house?
```
(and (argueWith' E1 C LT) (inside' E2 C) (inside' E3 LT)
(hit' E4 C LT) (exit' E5 LT) (seq E1 E4 E5))
```
a. The triangle leaves the house because it wants the circle to come fight it outside.
```
(and (attack' e6 C LT) (goal' e7 e6 LT))
```
b. The triangle leaves the house because it is afraid of being further assaulted by the circle.
```
(and (attack' e8 C LT) (fearThat' e9 LT e8))
```

83: A small triangle and big triangle are next to each other. A circle runs by and pushes the small triangle. The big triangle chases the circle. Why does the big triangle chase the circle?
```
(and (approach' E1 C LT) (push' E2 C LT) (chase' E3 BT C)
(seq E1 E2 E3))
```
a. The big triangle is angry that the circle pushed the small triangle, so it tries to catch the circle.
```
(angryAt' e4 BT C)
```
b. The big triangle and circle are friends. The big triangle wants to say hello to the circle.
```
(and (friend' e5 BT C) (goal' e6 e7 BT)
(greet' e7 BT C))
```

As a benchmark set of challenge problems for automated reasoning systems, Triangle-COPA has a number of attractive characteristics. By providing first-order logic representations as inputs and outputs, Triangle-COPA focuses the efforts of competitors specifically on the central interpretation problem. At the same time, it places no constraints on the particular reasoning methods that are actually used to select the correct answer, affording comparisons between systems that use radically different knowledge resources and reasoning algorithms. The relational vocabulary of Triangle-COPA literals are fixed [13], but the semantics of these predicates are not tied to any one ontology or theory. The correct answers of Triangle-COPA are randomly sorted, so the quality of any given system can be gauged between that of random guessing (50%) and human performance (near 100%).

Thus far, only Maslan et al. [13] has demonstrated an approach to solving Triangle-COPA problems. Using five axioms and an implementation of weighted abduction [10], the authors demonstrated that the least-cost proof of the observables in Question 83 (above) entailed answer "a", that the big triangle (BT) was angry at the circle (C).

In the following two sections, we show two alternative approaches to solving the scenario described in Question 83. Our aim is to demonstrate that this benchmark set of questions can serve as a grounds for comparison of different logical formalisms, algorithms, and knowledge bases, and help the larger automated

reasoning community make progress on the difficult challenges of automated commonsense reasoning.

3 Probabilistic Abductive Reasoning

Triangle-COPA problems can be viewed as a choice between two alternative interpretations of a sequence of observable actions. Hobbs et al. [8] describes how interpretation of natural language can be cast as a problem of logical abduction, and solved using automated abductive reasoning technologies. Abduction, as distinct from logical deduction or induction, is a form of logical reasoning that identifies a hypothesis that, if it were true, would logically entail the given input. In classical logic, abduction is not a sound inference mechanism; asserting the truth of an antecedent given an observable consequent is a logical fallacy, "affirming the consequent." Still, automated abductive reasoning is a natural fit for many commonsense reasoning problems in artificial intelligence, including the interpretation problems in Triangle-COPA.

Automated abductive reasoning requires two mechanisms: a means of generating sets of hypotheses that entail the input, and a scoring function for preferential ordering these hypotheses. Hobbs et al. [8] described "Weighted Abduction," where hypotheses are generated by backchaining from the given input using the implicature form of knowledge base axioms, unifying literals across different antecedents wherever possible. The process generates an and-or proof graph similar to that created when searching for first-order proofs by backchaining, but where every solution in the and-or graph identifies a set of assumptions that, if true, would logically entail the given observables. Weighted Abduction orders these hypotheses by computing the combined cost of all assumed literals (those without justification), through a mechanism of propagating initial costs to antecedents during backchaining. Maslan et al. [13] demonstrated how Weighted Abduction can be used to solve Triangle-COPA problems by searching for the least-cost set of assumptions that entailed the literals in one of the two alternatives.

Several researchers have pursued probabilistic reformulations of Weighted Abduction, eschewing the use of ad-hoc weights for probabilities that might be learned from empirical data. Ovchinnikova et al. [17] and Blythe et al. [2] describe two recent probabilistic reformulations, each casting the and-or proof graph as a Bayesian network whose posterior probabilities can be calculated using belief propagation algorithms for graphical models. These efforts help to position abductive reasoning among current approaches to uncertain inference, and to take advantage of recent advances and tools for reasoning with Markov Logic Networks [20]. However, a simpler formulation of probabilistic abduction may be more appropriate when the task is only to rank possible hypotheses.

As in other probabilistic reasoning tasks, the calculation of the joint probability of a set of events is trivially easy if we assume that they are all conditionally independent: the joint probability of the conjunction is the product of their prior probabilities. If we know the prior probabilities of all assumed literals in an abductive proof (those without justification), then the naive estimate of their joint

probability is simply their product [19]. This calculation can be applied to any solution in an and-or graph created by backchaining from the given input, giving us a convenient means of ranking hypotheses.

This approach allows us to use standard first-order logic and familiar technologies of lifted backchaining instead of belief propagation in graphical models. However, by using logical inference (rather than uncertain inference) we require that the consequent of an implication is always true when the antecedent holds, i.e. the probability of the consequent given the antecedent is always one. Hobbs et al. [8], building on McCarthy's [15] formulation of circumscription, describes how defeasible first-order axioms can be authored by the inclusion of a special etcetera literal (etc) as a conjunct in the antecedent. These literals are constructed with a unique predicate name that appears nowhere else in the knowledge base, and therefore can only be assumed (via abduction), never proved. The arguments of this predicate are all of the other variables that appear in the axiom, restricting its unification with other etcetera literals of the same predication that may be assumed in the proof.

The probabilities of etcetera literals can be quantified if we interpret them as being an unspecified conjunction of all of the unknown factors of the world that must also be true for the antecedent to imply the consequent. Etcetera literals are true in exactly the cases where the remaining antecedent literals and the consequent are all true. As such, their prior probabilities are equal to the conditional probability of the consequent given the remaining conjuncts in the antecedent.

Using etcetera literals, we can author defeasible versions of the five axioms used by Maslan et al. [13] to correctly solve Triangle-COPA Question 83, above. Here the prior probabilities of the etcetera literals are encoded directly in the literal as its first argument, appearing a numerical constant.

- Push: Maybe you are attacking
  ```
  (implies (and (attack_ e1 x y)
                (goal_ e2 e1 x)
                (etc1_push 0.9 e1 e2 x y))
           (push_ e3 x y))
  ```
- Approach: Maybe you want to attack
  ```
  (implies (and (goal_ e1 e2 x)
                (attack_ e2 x y)
                (etc1_approach 0.9 e1 e2 x y))
           (approach_ e4 x y))
  ```
- AngryAt: Maybe they attacked someone you like
  ```
  (implies (and (attack_ e1 y z)
                (like_ e2 x z)
                (etc1_angryAt 0.9 e1 e2 x y z))
           (angryAt_ e x y))
  ```

- Attack: Maybe you are angry at them
  ```
  (implies (and (angryAt_ e1 x y)
                (etc1_attack 0.9 e1 x y))
           (attack_ e x y))
  ```
- Chase: Maybe you want to attack
  ```
  (implies (and (attack_ e1 x y)
                (goal_ e2 e1 x)
                (etc1_chase 0.9 e1 e2 x y))
           (chase_ e3 x y))
  ```

Etcetera literals also afford a means of encoding the prior probabilities of other literals directly in the knowledge base. Below we provide eight additional axioms, one for each predicate used in either the Triangle-COPA question or in the axioms above, where an etcetera literal is the antecedent of each predicate form. By adding these axioms to the knowledge base, we can conduct our search for unique sets of assumptions by backtracking on all axioms to construct an and-or graph that terminates with etcetera literals. The probability of any solution in this and-or graph (assuming conditional independence) is simply the product of the priors of each etcetera literal.

```
(implies (etc0_push 0.01 e x y) (push_ e x y))
(implies (etc0_approach 0.01 e x y) (approach_ e x y))
(implies (etc0_angryAt 0.01 e x y) (angryAt_ e x y))
(implies (etc0_attack 0.01 e x y) (attack_ e x y))
(implies (etc0_chase 0.01 e x y) (chase_ e x y))
(implies (etc0_goal 0.9 e x y) (goal_ e x y))
(implies (etc0_like 0.9 e x y) (like_ e x y))
(implies (etc0_seq3 1.0 x y z) (seq x y z))
```

Figure 2 shows a visual representation of the most probable proof ($Pr = 0.0043$) of the given observables of Triangle-COPA Question 83, found amongst a set of 6038 possible proofs found by backchaining on the axioms listed above. The approach happens because the circle (C) had the goal to attack the little triangle (LT). The push happens for this same reason, and these explanations are unified. The chase happens because the big triangle (BT) had the goal to attack the circle, because it was angry at the circle, because the circle's attack on someone that the big triangle likes. The attacks are unified, and we infer that the big triangle likes the little triangle. Left unexplained are why the circle had the goal of attacking the little triangle, why the big triangle likes the circle, why attacking was the goal chosen by the big triangle, and why these eventualities happened in this sequence. The correct alternative appears in the most-probable proof, namely that the big triangle is angry at the circle.
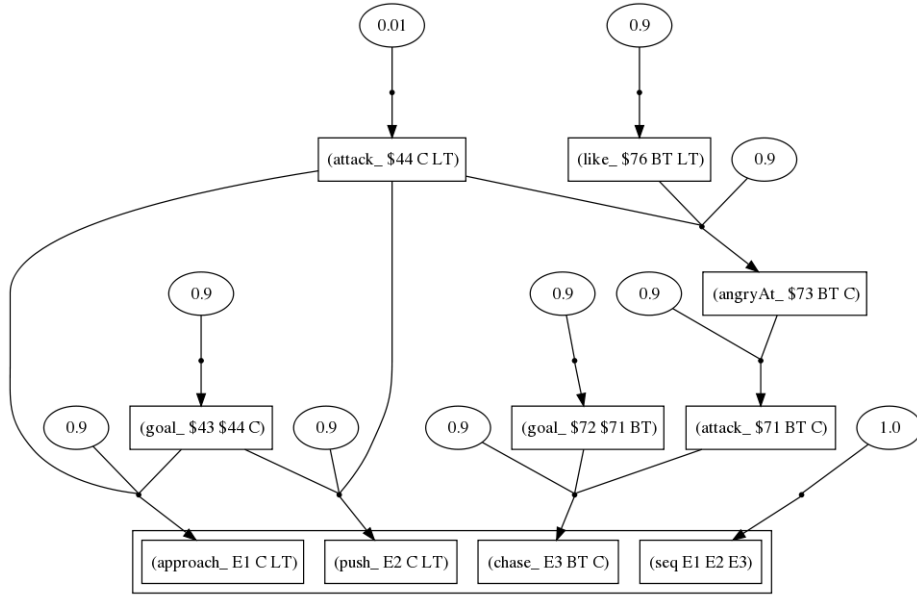
Fig. 2: The most-probable proof of Triangle-COPA Question 83.

4 Standard Deontic Logic

Now we consider a different approach, namely deontic logic, to tackle the Triangle-COPA benchmarks. This approach appears to be promising, since in [4] it was demonstrated that deontic logic is very well-suited for modelling different kinds of human reasoning. There are interesting examples from cognitive psychology, e.g. the Wason-selection task, or the suppression task, which can be formalized in a way that they are accessible for automated reasoning systems. We are using the tableau prover Hyper [1], a first order refutational theorem prover, which is able to decide standard deontic logic as well.

Standard deontic logic (SDL) is obtained from the well-known modal logic K by adding the seriality axiom D:

$$\mathsf{D}: \quad \Box P \to \Diamond P$$

In this logic, the $\Box$-operator is interpreted as "it is obligatory that" and the $\Diamond$ as "it is permitted that". The $\Diamond$-operator can be defined by the following equivalence:

$$\Diamond P \equiv \neg \Box \neg P$$

The additional axiom D: $\Box P \to \Diamond P$ in SDL states that if a formula has to hold in all reachable worlds, then there exists such a world. With the deontic reading of $\Box$ and $\Diamond$ this means: Whenever the formula P ought to be, then there exists a world where it holds. In consequence, there is always a world, which is

ideal in the sense that all the norms formulated by the 'ought to be'-operator hold.

SDL can be used in a natural way to describe knowledge about norms or licenses. The use of conditionals for expressing rules which should be considered as norms seems likely, but holds some subtle difficulties. If we want to express that *if P then Q* is a norm, an obvious solution would be to use

$$\Box(P \to Q)$$

which reads *it is obligatory that Q holds if P holds*. An alternative would be

$$P \to \Box Q$$

meaning *if P holds, it is obligatory that Q holds*. In [24] there is a careful discussion which of these two possibilities should be used for conditional norms. The first one has severe disadvantages. The most obvious disadvantage is that P together with $\Box(P \to Q)$ does not imply $\Box Q$. This is why we prefer the latter method, where the $\Box$-operator is in the conclusion of the conditional. For a more detailed discussion of such aspects we refer to [5].

For the examples in Triangle-COPA we argue that one can understand norms as expectation—many emotions in everyday life can be explained with unmet expectations. The husband not bringing flowers on the wedding anniversary and the friend arriving delayed to a date are only two examples, where unmet expectations cause negative feelings. On the other hand, expectations met can cause positive feelings. The husband helping with the dishes causes the wife to be content. We consider the scenario described in Question 83 from Triangle-COPA corresponding to the following set of facts:

$$approach(e1, c, lt). \tag{1}$$
$$push(e2, c, lt). \tag{2}$$
$$chase(e3, bt, c). \tag{3}$$
$$seq(e1, e2, e3). \tag{4}$$

The last fact states that the eventualities $e1$, $e2$ and $e3$ constitute a sequence of events.

The question we are asking is "How does the little triangle feel?". The two alternatives provided are as follows:

a. The little triangle feels relieved:
   $relief(e4, lt, e3)$
b. The little triangle is angry at the big triangle:
   $angryAt(e5, lt, bt)$

The notion of fulfilled expectations can be helpful to answer this question. The big triangle observes the circle attacking the little triangle. The little triangle expects the big triangle to defend it. The big triangle chases the circle away from the little triangle which corresponds to defending it. The little triangle is relieved that the big triangle hurried to its defense.

We need some background knowledge in this example:

- Pushing someone means attacking someone:

$$push(E, X, Y) \rightarrow attack(E, X, Y). \tag{5}$$

- Chasing an attacker means defending the person under attack:

$$attack(E, X, Y) \wedge chase(E', Z, X) \wedge after(E, E') \rightarrow defend(E', Z, Y). \tag{6}$$

Where *after* is a transitive predicate, stating that one eventuality occurs after another. $after(e1, e2)$ means that event $e2$ occurs after $e1$.

It is possible to model expectations with the help of deontic logic. Normative statements are used to model expected behavior. In our example, we use deontic logic to model the fact that one should defend someone who is attacked by someone else. This set of deontic formulae is the set of ground instances of the following formula:

$$attack(E, Z, X) \rightarrow \Box defend(E, Y, X) \vee Z = Y. \tag{7}$$

Formula (7) is not a SDL formula. However, we use it as an abbreviation for its set of ground instances. The ground instance interesting for our example is:

$$attack(e2, c, lt) \rightarrow \Box defend(e2, bt, lt) \vee c = bt. \tag{8}$$

With the help of formula (8), it is possible to derive that the big triangle ought to defend the little triangle in event $e2$.

Formula 8 states that in the ideal world following eventuality $e2$, the big triangle defends the little triangle. Another possibility to express this, would be to use the eventuality, which is part of every atom. We could state $defend(e9, bt, lt)$ for some new eventuality $e9$ and add some information stating that eventuality $e9$ is the ideal successor of $e2$. For this it would be necessary to introduce a new relation, connecting eventualities with its ideal successor. Since this is rather cumbersome, we use standard deontic logic instead.

Ground instances of the following formula can be used to deduce that someone is relieved if someone ought to be defended by someone and is actually defended:

$$(\Box defend(E, X, Y) \wedge defend(E', X, Y) \wedge after(E, E'))) \rightarrow (\bigwedge_{\substack{\forall E'' \\ after(E', E'')}} relief(E'', Y, E'))$$

A ground instance interesting for our example is:

$$(\Box defend(e2, bt, lt) \wedge defend(e3, bt, lt) \wedge after(e2, e3) \rightarrow$$
$$(relief(e4, lt, e3) \wedge relief(e5, lt, e3)) \tag{9}$$

We want to use a theorem prover in order solve example 83 together with the above introduced question. To accomplish this, the following formulae are combined to one set of formulae S:

- Formulae (1) - (4) describing the scenario,
- the background knowledge given (5) and (6),
- some additional formulae formalizing the *after* predicate,
- the deontic logic formulae (8) and (9) stating the information about expectations and
- some formulae stating that bt, lt and c are pairwise different.

We use Hyper to solve example 83 with the question introduced above. It is possible to deduce that the little triangle is relieved in $e4$ by transforming this reasoning task into a satisfiability test. Hyper constructs a closed hyper tableau for $S \cup \{\neg relief(e4, lt, e3)\}$ which implies that $relief(e4, lt, e3)$ is entailed by S.

Referring to the question "How does the little triangle feel?" formulated before, we can use the derived $relief(e4, lt, e3)$ to show that the second alternative given is the correct one.

Of course, it is not desirable to formalize all rules manually. Rules like (9) can be generated automatically by formalizing a metarule stating that: whenever x and y are friends and y is obliged to do something for x and actually does it, x is relieved. This metarule can then be instantiated by the respective obligation.

5 Integration of Background Knowledge

In the previous section we used standard deontic logic to tackle one of the examples from Triangle-COPA. In addition to the formulae for normative statements, we used formulae (5) and (6) stating some essential background knowledge. In order to solve all Triangle-COPA benchmarks, an extensive background knowledge on psychology is essential. It is labor intensive and error-prone to state the whole background knowledge manually. Therefore it is desirable to use existing knowledge bases. There are several detailed ontologies like Yago [22], Cyc [11], and Sumo [16], stating knowledge about common sense. The very size of these ontologies however forbids to use these ontologies entirely. For example Yago contains more than 10 million entities (like persons, organizations, cities, etc.) and contains more than 120 million facts about these entities. ResearchCyc contains more than 500,000 concepts, forming an ontology in the domain of human consensus reality. Nearly 5,000,000 assertions (facts and rules) using more than 26,000 relations interrelate, constrain, and, in effect, (partially) define the concepts. And even the smallest version of Cyc, OpenCyc, still contains more than 3 million formulae.

Therefore it is necessary to extract relevant parts from these ontologies. However brute-force extraction by selecting for example all assertions from OpenCyc containing the word "attack" results in a set of 13,184 assertions. The vast majority of these assertions contains irrelevant information. For example assertions about the movie "Mars attacks" are selected. These irrelevant assertions potentially thwart the reasoning process, making it worthwhile to invest some effort into carefully selecting assertions suitable as background knowledge. Partitioning techniques used to handle large theories with theorem provers like the SInE (Sumo Inference Engine) [9] metaprover could be helpful to address this problem.

6 Discussion

Benchmark problems have helped to spur new ideas and compare technologies across many areas of computer science and beyond. For researchers interested in logical approaches to automated reasoning, as in other fields, the most useful benchmarks will be those that focus specifically on the core research challenge, but are not prejudice for or against any one technical approach. In this paper we have argued that the Triangle-COPA set of challenge problems is a useful tool for exploring the relationship between human and logical reasoning. We described two different logical approaches for solving Triangle-COPA questions, a probabilistic form of logical abduction and deontic logic. In so doing, we demonstrate that the questions are agnostic to the particular logic framework that is used to solve them. By tackling the same questions with different approaches, we gain new insights into both the similarities and differences afforded by different techniques. We encourage other research groups in our community to apply their unique approaches to the same questions, to consider the similarities and differences among approaches that go beyond the shallow characteristics of various logical notation, and to focus their efforts on overcoming the enormous challenges of humanlike commonsense reasoning.

References

1. Peter Baumgartner, Ulrich Furbach, and Björn Pelzer. Hyper tableaux with equality. In Frank Pfennig, editor, *CADE 21*, volume 4603 of *LNCS*, 2007.
2. James Blythe, Jerry R. Hobbs, Pedro Domingos, Rohit J. Kate, and Raymond J. Mooney. Implementing weighted abduction in markov logic. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pages 55–64, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
3. David Ferrucci, Anthony Levas, Sugato Bagchi, David Gondek, and Erik T. Mueller. Watson: Beyond jeopardy! *Artificial Intelligence*, 199–200(0):93 – 105, 2013.
4. Ulrich Furbach and Claudia Schon. Deontic logic for human reasoning. In Thomas Eiter, Hannes Strass, Miroslaw Truszczynski, and Stefan Woltran, editors, *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation - Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, volume 9060 of *Lecture Notes in Computer Science*, pages 63–80. Springer, 2014.
5. D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. van der Torre, editors. *Handbook of Deontic Logic and Normative Systems*. College Publications, 2013.
6. Fritz Heider. *The Psychology of Interpersonal Relations*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1958.
7. Fritz Heider and Marianne Simmel. An experimental study of apparent behavior. *American Journal of Psychology*, 57(2):243–259, 1944.
8. Jerry R. Hobbs, Mark E. Stickel, Douglas E. Appelt, and Paul Martin. Interpretation as abduction. *Artif. Intell.*, 63(1-2):69–142, October 1993.
9. Krystof Hoder and Andrei Voronkov. Sine qua non for large theory reasoning. In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *Automated Deduction - CADE-23 - 23rd International Conference on Automated Deduction, Wroclaw,*

Poland, July 31 - August 5, 2011. Proceedings, volume 6803 of *Lecture Notes in Computer Science*, pages 299–314. Springer, 2011.

10. Naoya Inoue and Kentaro Inui. Ilp-based inference for cost-based abduction on first-order predicate logic. *Journal of Natural Language Processing*, 20(5):629–656, 2013.

11. Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

12. Hector J. Levesque. The Winograd Schema Challenge. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI, 2011.

13. Nicole Maslan, Melissa Roemmele, and Andrew S. Gordon. One hundred challenge problems for logical formalizations of commonsense psychology. In *Twelfth International Symposium on Logical Formalizations of Commonsense Reasoning, Stanford, CA*, 2015.

14. John McCarthy. Programs with common sense. In *Semantic Information Processing*, pages 403–418. MIT Press, 1968.

15. John McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

16. Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9. ACM, 2001.

17. Ekaterina Ovchinnikova, Andrew S. Gordon, and Jerry. R. Hobbs. Abduction for discourse interpretation: A probabilistic framework. In *Joint Symposium on Semantic Processing*, pages 42–50, 2013.

18. David Pautler, Bryan L. Koenig, Boon-Kiat Quek, and Andrew Ortony. Using modified incremental chart parsing to ascribe intentions to animated geometric figures. *Behavior Research Methods*, 43(3):643–665, 2011.

19. David Poole. Representing bayesian networks within probabilistic horn abduction. In Bruce D'Ambrosio and Philippe Smets, editors, *UAI '91: Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence, University of California at Los Angeles, Los Angeles, CA, USA, July 13-15, 1991*, pages 271–278. Morgan Kaufmann, 1991.

20. Matthew Richardson and Pedro M. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

21. Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.

22. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A large ontology from wikipedia and wordnet. *Web Semant.*, 6(3):203–217, September 2008.

23. Robert H. Thibadeau. Artificial perception of actions. *Cognitive Science*, 10(2):117–149, 1986.

24. Frank von Kutschera. *Einführung in die Logik der Normen, Werte und Entscheidungen*. Alber, 1973.