

Limbo: A Reasoning System for Limited Belief (Demonstration)

Christoph Schwering

School of Computer Science and Engineering
The University of New South Wales
Sydney NSW 2052, Australia
c.schwering@unsw.edu.au

Abstract

We introduce LIMBO, a reasoning system for limited belief. The system features a highly expressive language with first-order quantification, functions and equality, sorts, and introspective belief modalities. Reasoning is based on clause subsumption, unit propagation, and case splits. Decidability and sometimes even tractability is achieved by limiting the number of case splits. This demo illustrates the practical utility of limited belief by way of toy examples as well as the games of Sudoku and Minesweeper.

1 Introduction

Reasoning about incomplete knowledge is part and parcel of intelligent behaviour. Unfortunately, it often requires highly expressive languages to accurately represent the incompleteness of knowledge, and this high degree of expressivity typically leads to poor computational complexity. This trade-off is what makes designing languages and systems for Knowledge Representation and Reasoning a notoriously difficult task.

One approach to deal with the complexity of very expressive languages is through *limited belief*. Originally motivated by philosophical considerations [Hintikka, 1975], the idea behind limited belief is to *weaken* the logical entailment relation in a way that preserves soundness but trades completeness for better computational complexity. Despite these appealing properties, limited belief has remained a tool for theoretical analysis without practical relevance until now.

This demo showcases LIMBO,¹ a system for reasoning about limited belief in first-order knowledge bases. LIMBO implements the theory of limited belief from [Schwering, 2017], which itself is based on earlier work by Liu, Lakemeyer, and Levesque [2004; 2013; 2014; 2016]. To my knowledge, LIMBO is the first implementation of any of these theories.

This demo aims to show that (i) limited belief is not just a theory but also has practical value and that (ii) LIMBO makes a useful framework for bringing to “life” otherwise predominantly theoretical work such as theories of actions, belief change, and multi-agent reasoning.

¹The following material is available online:

Source code: www.github.com/schwering/limbo

Examples: www.cse.unsw.edu.au/~cschwering/limbo

Video: www.cse.unsw.edu.au/~cschwering/limbo/video

```
1 Sort HUMAN, BOOL
2 Name Frank, Fred, Sally -> HUMAN
3 Name T -> BOOL
4 Fun fatherOf/1 -> HUMAN
5 Fun rich/1 -> BOOL
6 Var x -> HUMAN
7 // Sally's father is Frank or Fred, and he's rich.
8 KB: fatherOf(Sally) = Frank v
   fatherOf(Sally) = Fred
9 KB: rich(fatherOf(Sally)) = T
10 // Is Sally's father rich but unknown?
11 K<1> ex x (fatherOf(Sally) = x ^ rich(x) = T
   ^ M<1> fatherOf(Sally) /= x)
```

Listing 1: A simple reasoning problem specified in LIMBO’s textual interface. Lines 1–6 declare non-logical symbols; lines 8 and 9 define the knowledge base; line 11 specifies a query.

2 The Reasoning System

The representation language of LIMBO is a first-order modal dialect with functions, equality, sorts, and introspective belief operators [Schwering, 2017]. Semantically, belief is stratified in *levels*: level 0 comprises only the explicit beliefs; every following level draws additional inferences by doing another case split. Every belief operator specifies at which belief level it shall be evaluated, and thus controls how much effort should be spent on proving it.

Let us illustrate LIMBO’s expressivity with a brief example. Suppose that all we know is that either Frank or Fred is Sally’s father and that Sally’s father is rich. Intuitively this knowledge base should entail the query “we don’t know Sally’s father, but we do know he’s rich.” Formalising this query however is not trivial at all, as it requires a significant degree of expressivity: it involves an individual (Sally’s father) and both knowns (he’s rich) and unknowns (his identity) about him.

First-order modal logic provides a way to express the above knowledge base and query. A formalisation in the language of LIMBO’s textual user interface is given in Listing 1. In the following we describe what LIMBO by means of this example.

Names in LIMBO represent distinct objects. For instance, Frank, Fred, and Sally represent three different individuals of the same sort HUMAN. LIMBO assumes for every sort the existence of an infinite number of such objects. In logical terminology, **Names** are constants that satisfy the unique-name

	NYT easy	NYT medium	NYT hard	Top 1465
clues	38.0	24.4	24.0	18.0
level 0	42.8	49.5	44.2	45.1
level 1	0.3	6.6	11.2	9.5
level 2	–	0.5	1.8	4.6
level 3	–	–	–	3.1
level 4	–	–	–	0.5
level 5	–	–	–	0.01
time	0.1 s	0.8 s	4.1 s	49.5 m

Table 1: Sudoku experiments over eight puzzles of each category from The New York Times website as well as the first 125 of the “Top 1465” list. The rows show how many cells on average per game were preset or solved at belief level 1, 2, 3, 4, or 5. The last row shows the average time per puzzle.

assumption and an infinitary version of domain closure.

Aside from these special constants, LIMBO also supports classical functions, such as the unary function `fatherOf` that returns an object of sort `HUMAN`. Predicates are not part of the language at the moment; they can however easily be simulated without overhead with functions of a special sort and a distinguished name that represents truth, such as `Bool` and `T`.

Knowledge bases in LIMBO are subject to two syntactic restrictions: they must be in clausal form and contain no existential quantifiers. The former however is no effective restriction as existentials can be simulated with Skolem functions. The semantic concept underlying a knowledge base is Levesque’s only-knowing [1984], which implicitly specifies that everything that is not entailed by the knowledge base is unknown.

The query in Listing 1 intuitively says that “we don’t know who Sally’s father is, but we do know he’s rich.” The $\mathbf{K}\langle 1 \rangle$ and $\mathbf{M}\langle 1 \rangle$ modal operators express what is known and what is considered possible, respectively, at belief level 1. Observe that the variable x is existentially quantified into the modal context of $\mathbf{M}\langle 1 \rangle$, which captures the difference between knowing *who* and knowing *that*. LIMBO proves the query by splitting the cases for `fatherOf(Sally)`. As for $\mathbf{K}\langle 1 \rangle$, if `Frank` is the father, then he’s rich, and if `Fred` is the father, then he is rich, and every other potential father is ruled out by the knowledge base. As for $\mathbf{M}\langle 1 \rangle$, both `Frank` and `Fred` are consistent options for `fatherOf(Sally)`, so the father is unknown.

LIMBO is implemented in C++ and closely follows its logical specification [Schwering, 2017]. A noteworthy point about the implementation is that terms and literals have a very lightweight representation. Using a technique called *interning*, each literal is represented by a single 64-bit number, and whether two ground literals are complementary or one subsumes the other can be computed by bitwise operations on these numbers. This lightweight representation turned out to be 24× faster than the naive representation and helps us to adopt SAT solving technology, where literals by nature are very lightweight. Some syntactic restrictions were useful to facilitate this representation. For instance, we disallow nested functions, so `rich(fatherOf(Sally)) = T` is flattened to `fa x (fatherOf(Sally) /= x v rich(x) = T)`. Similarly, functions are only allowed on the left-hand side of literals. LIMBO automatically rewrites formulas to adhere to this syntactic format.

level		8×8–10	16×16–40	16×30–99	32×64–320
0	win	62.0%	46.0%	1.4%	28.3%
	time	0.01 s	0.06 s	0.24 s	5.08 s
1	win	87.3%	84.9%	37.7%	69.8%
	time	0.01 s	0.08 s	0.43 s	5.46 s
2	win	87.8%	85.0%	39.1%	70.0%
	time	0.02 s	0.10 s	0.64 s	5.60 s
3	win	87.8%	85.0%	39.1%	70.0%
	time	0.07 s	0.25 s	4.94 s	5.90 s

Table 2: Minesweeper experiments over 1000 randomised runs of different configurations, where $W \times H - M$ means M mines on a $W \times H$ grid. The rows contain results for different maximum belief levels used by the reasoner to figure out whether cells are safe or not. Numbers are the chance of winning and execution time per game.

3 Experimental Results

We put to test the concept of limited belief using the games of Sudoku and Minesweeper.² While both games do not require much expressivity to be modelled, they are nevertheless interesting applications of limited belief because they are known to be computationally hard – Sudoku on $N \times N$ grids is NP-complete, Minesweeper is co-NP-complete – yet often easily solved by humans.

According to the motivating hypothesis behind limited belief, a small belief level should often suffice to reach human-level performance. Indeed we find this hypothesis confirmed for both games. The results for Sudoku in Table 1 show that most ‘easy’ instances are solved just by unit propagation and that the number of necessary case splits moderately increases for ‘medium’ and ‘hard’ games. Significantly higher belief levels are needed to solve games from the “Top 1465” list, a repository of extremely difficult Sudokus. For Minesweeper, Table 2 shows that strong results are achieved at level 1 already, and while belief level 2 increases the chance of winning by 0.5%, there is no further improvement at level 3.

4 Discussion and Perspectives

The experimental results show that limited belief is an effective method for bringing first-order epistemic languages to practice. One goal of LIMBO is to encourage authors of related theories in the field of epistemic reasoning to bring their theories to “life” as well. For instance, a theory of (limited) conditional belief [Schwering and Lakemeyer, 2016] has been integrated into LIMBO already. Theories of action [Reiter, 2001; Lakemeyer and Levesque, 2014], belief change [Schwering *et al.*, 2015], and/or multiple agents [Belle and Lakemeyer, 2015] are further candidates to be incorporated.

While the system at the current stage is still a long way from real-world applications, LIMBO’s runtime performance appears promising enough to expect useful results from such a highly expressive system; especially considering that SAT solving techniques like clause learning or backjumping are not used by the system yet. With added expressivity, we see potential applications of LIMBO in domains like epistemic planning or high-level control in cognitive robotics.

²For details on these games, see en.wikipedia.org/wiki/Sudoku and [en.wikipedia.org/wiki/Minesweeper_\(video_game\)](http://en.wikipedia.org/wiki/Minesweeper_(video_game)).

References

- [Belle and Lakemeyer, 2015] Vaishak Belle and Gerhard Lakemeyer. Semantical considerations on multiagent only knowing. *Artif. Intell.*, 223:1–26, 2015.
- [Hintikka, 1975] Jaakko Hintikka. Impossible possible worlds vindicated. *J. Philos. Logic*, 1975.
- [Lakemeyer and Levesque, 2013] Gerhard Lakemeyer and Hector J. Levesque. Decidable reasoning in a logic of limited belief with introspection and unknown individuals. In *Proc. IJCAI*, 2013.
- [Lakemeyer and Levesque, 2014] Gerhard Lakemeyer and Hector J. Levesque. Decidable reasoning in a fragment of the epistemic situation calculus. In *Proc. KR*, 2014.
- [Lakemeyer and Levesque, 2016] Gerhard Lakemeyer and Hector J. Levesque. Decidable reasoning in a logic of limited belief with function symbols. In *Proc. KR*, 2016.
- [Levesque, 1984] Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artif. Intell.*, 1984.
- [Liu *et al.*, 2004] Yongmei Liu, Gerhard Lakemeyer, and Hector J. Levesque. A logic of limited belief for reasoning with disjunctive information. In *Proc. KR*, 2004.
- [Reiter, 2001] Ray Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press, 2001.
- [Schwering and Lakemeyer, 2016] Christoph Schwering and Gerhard Lakemeyer. Decidable reasoning in a first-order logic of limited conditional belief. In *Proc. ECAI*, 2016.
- [Schwering *et al.*, 2015] Christoph Schwering, Gerhard Lakemeyer, and Maurice Pagnucco. Belief revision and progression of knowledge bases in the epistemic situation calculus. In *Proc. IJCAI*, 2015.
- [Schwering, 2017] Christoph Schwering. A reasoning system for a first-order logic of limited belief. In *Proc. IJCAI*, 2017. To appear.