A Reasoning System for a First-Order Logic of Limited Belief

Christoph Schwering

UNSW Sydney
What is limited belief? And why?

**Task:** Given a KB and a query:

Does the KB *logically entail* the query?
What is limited belief? And why?

**Task:** Given a KB and a query:

Does the KB *logically entail* the query?

Which logic?
What is limited belief? And why?

**Task:** Given a KB and a query:

Does the KB *logically entail* the query?

Which logic?

Classical logic makes the agent **omniscient**:

- Unrealistic
- Undecidable (*first-order*) / intractable (*propositional*)
What is limited belief? And why?

**Task:** Given a KB and a query:

Does the KB *logically entail* the query?

Which logic?

**Limited belief:**

- **Belief level 0:** explicitly written down in the KB
- **Belief level $k > 0$:** derivable from KB with effort $k$
**Language**

**FOL** with equality + functions + sorts +

- Knowledge:  \( K_0 \alpha, K_1 \alpha, K_2 \alpha, \ldots \)
- Contingency:  \( M_0 \alpha, M_1 \alpha, M_2 \alpha, \ldots \)

**Example:**

- \( K_1 (R(\text{Frank}) \lor R(\text{Fred})) \)  
  We know Frank or Fred is rich
- \( \forall x \, M_1 f(S) \neq x \)  
  We don’t know who Sally’s father is
- \( K_1 \exists x (f(S) = x \land R(x) \land M_1 f(S) \neq x) \)  
  We know Sally’s father is rich, but we don’t who he is
Semantics

**Model:** set of clauses closed under unit propagation

- **Belief level 0:** subsumption
- **Belief level** $k > 0$: $k$ case splits

**Example:**

If all we know is (1) $f(S) = \text{Frank} \lor f(S) = \text{Fred}$

and (2) $\forall x (f(S) \neq x \lor R(x))$,

then $K_1 (R(\text{Frank}) \lor R(\text{Fred}))$?

**Yes!** Branch on $f(S)$:

- $\{ (1), (2), f(S) = \text{Frank} \} \ni R(\text{Frank})$ by UP with (2)
- $\{ (1), (2), f(S) = \text{Fred} \} \ni R(\text{Fred})$ by UP with (2)
- $\{ (1), (2), f(S) = n \} \ni \bot$ by UP with (1)

for $n \neq \text{Frank, Fred}$
KB entails $\sigma$ at some belief level $\implies$ KB classically entails $\sigma$

if $\sigma$ contains no $\neg K, \neg M$
KB entails $\sigma$ at some belief level $\iff$ KB classically entails $\sigma$

if $\sigma$ contains no $\neg K$, $\neg M$ and $KB$, $\sigma$ contain no $\exists$, $\forall$
KB entails $\sigma$ at some belief level is decidable
KB entails $\sigma$ at some belief level is tractable
for fixed belief level if KB, $\sigma$ contain no $\exists, \forall$
Hypothesis: human-level results at small belief levels
Hypothesis: human-level results at small belief levels ✓

Experiments: Sudoku

- Easy
- Medium
- Hard
- Top

Average # of cells solved at...

- clues
- level 0
- level 1
- level 2
- level 3
- level 4
- level 5
Hypothesis: human-level results at small belief levels ✔
**Limbo = Limited Belief**

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

www.github.com/schwering/limbo

Next: 1. actions  2. belief change  3. multiple agents
Appendix
Language in detail

**Terms:**
- First-order variables
- Functions $f(t_1, \ldots, t_m)$ where each $t_i$ is a name or variable
- Standard names infinitely many and sorted

**Formulas:**
- FOL: \( t_1 = t_2 \quad \neg \alpha \quad \alpha \lor \beta \quad \exists x \alpha \)
- Knowledge: \( K_0 \alpha \quad K_1 \alpha \quad K_2 \alpha \quad \ldots \)
- Contingency: \( M_0 \alpha \quad M_1 \alpha \quad M_2 \alpha \quad \ldots \)
- Knowledge base: \( O \alpha \) where \( \alpha \) is in universal CNF

\(\alpha \land \beta \quad \alpha \supset \beta \quad \alpha \equiv \beta \quad \forall x \alpha \) are abbreviations

- Predicates are simulated with functions
- Existentials in KBs are simulated with Skolem functions
- Functions on the right-hand side and within functions are flattened:
  \[
  f(\cdot) = g(\cdot) \quad \iff \quad \forall x (g(\cdot) = x \supset f(\cdot) = x)
  \]
  \[
  f(g(\cdot)) = t \quad \iff \quad \forall x (g(\cdot) = x \supset f(x) = t)
  \]
Literal encoding

- Functions cannot be nested
  \[ f(\cdot) = g(\cdot) \quad \Rightarrow \quad \forall x (g(\cdot) = x \supset f(\cdot) = x) \]

- Functions cannot appear on rhs
  \[ f(g(\cdot)) = t \quad \Rightarrow \quad \forall x (g(\cdot) = x \supset f(x) = t) \]

- Term is 30-bit pointer (interning)

- Literal is 64-bit number
  - 60 bits for lhs and rhs
  - 1 bit to indicate whether = or ≠
  - 2 bits to indicate if lhs / rhs is name

- Literal subsumption and complement test:
  - \( \ell \) subsumes \( \ell' \)
  - \( t = n_1 \) subsumes \( t \neq n_2 \)
  - \( t = t' \) and \( t \neq t' \) are complementary
  - \( t = n_1 \) and \( t = n_2 \) are complementary

- Bitwise op’s on 64-bit numbers suffice

- Fast clause subsumption and unit propagation
Example: “I don’t know Sally’s father, but I know he’s rich”

\[ c_1 = f(S) = \text{Frank} \lor f(S) = \text{Fred} \]
\[ c_2 = \forall x (f(S) \neq x \lor r(x) = \top) \]
\[ O(c_1 \land c_2) \models K \exists x (f(S) = x \land r(x) = \top \land M f(S) \neq x) \]
Example: “I don’t know Sally’s father, but I know he’s rich”

\[ e = \{ w \mid w \models f(S) = \text{Frank} \lor f(S) = \text{Fred} \land \\
\quad \forall x (f(S) \neq x \lor r(x) = \top) \} \]

\[ e \models K \exists x (f(S) = x \land r(x) = \top \land M f(S) \neq x) \]
Example: “I don’t know Sally’s father, but I know he’s rich”

\[ e = \{ w \mid w \models f(S) = \text{Frank} \lor f(S) = \text{Fred} \land \forall x (f(S) \neq x \lor r(x) = \top) \} \]

\[ e \models K\exists x (f(S) = x \land r(x) = \top \land Mf(S) \neq x) \]

- For every \( w \in e \), for some \( n \), \( w \models f(S) = n \land R(n) \)
- For some \( w' \in e \), \( w \models f(S) \neq n \)
Example: “I don’t know Sally’s father, but I know he’s rich”

- $c_1 = f(S) = \text{Frank} \lor f(S) = \text{Fred}$
- $c_2 = \forall x (f(S) \neq x \lor r(x) = \top)$
- $O(c_1 \land c_2) \models K_1 \exists x (f(S) = x \land r(x) = \top \land M_1 f(S) \neq x)$
Example: “I don’t know Sally’s father, but I know he’s rich”

\[
\begin{align*}
\text{s} &= \{ f(S) = \text{Frank} \lor f(S) = \text{Fred}, \\
&\quad \quad \quad \quad \quad \quad \quad \quad f(S) \neq n \lor r(n) = \top \mid n \text{ is a name} \} \\
\text{s} &\vdash K_1 \exists x \left( f(S) = x \land r(x) = \top \land M_1 f(S) \neq x \right)
\end{align*}
\]
Example: “I don’t know Sally’s father, but I know he’s rich”

- $s = \{f(S) = \text{Frank} \lor f(S) = \text{Fred}, \quad f(S) \neq n \lor r(n) = \top \mid n \text{ is a name}\}$
- $s \models K_1 \exists x (f(S) = x \land r(x) = \top \land M_1 f(S) \neq x)$

$\iff$

for some $t_1$, for all $n_1$, for some $n$,

$s \cup \{t_1 = n_1\} \models f(S) = n \land r(n) = \top \land M_1 f(S) \neq n$
Example: “I don’t know Sally’s father, but I know he’s rich”

- $s = \{ f(S) = \text{Frank} \lor f(S) = \text{Fred}, \quad f(S) \neq n \lor r(n) = \top \mid n \text{ is a name} \}$
- $s \models K_1 \exists x (f(S) = x \land r(x) = \top \land M_1 f(S) \neq x)$

$\iff$

for some $t_1$, for all $n_1$, for some $n$, $s \cup \{ t_1 = n_1 \} \models f(S) = n \land r(n) = \top \land M_1 f(S) \neq n$

$\iff$

for some $t_1$, for all $n_1$, for some $n$, $s \cup \{ t_1 = n_1 \} \models f(S) = n \land r(n) = \top$

for some $t_2$ and $n_2$, $s \cup \{ t_2 = n_2 \} \models f(S) \neq n$
Example: “I don’t know Sally’s father, but I know he’s rich”

- \( s = \{ \text{f(S) = Frank} \lor \text{f(S) = Fred}, \) \]
  \[ \text{f(S) \neq n} \lor \text{r(n) = } \top \mid n \text{ is a name} \} \]

- \( s \models \mathbf{K}_1 \exists x (\text{f(S) = x} \land \text{r(x) = } \top \land \mathbf{M}_1 \text{f(S) \neq x}) \)

\[ \iff \]

(a) for some \( t_1 \), for all \( n_1 \), for some \( n \), \( s \cup \{ t_1 = n_1 \} \models \text{f(S) = n} \land \text{r(n) = } \top \)

(b) for some \( t_2 \) and \( n_2 \), \( s \cup \{ t_2 = n_2 \} \models \text{f(S) \neq n} \)

\[ \iff \]

(a) choose \( t_1 = \text{f(S)} \):

  if \( n_1 = \text{Frank} \), choose \( n = \text{Frank} \):

  \( s \cup \{ \text{f(S) = Frank} \} \) contains \( \text{f(S) = Frank, r(Frank) = } \top \)

(b) choose \( t_2 = \text{f(S)} \) and \( n_2 = \text{Fred} \):

  \( s \cup \{ \text{f(S) = Fred} \} \) contains \( \text{f(S) \neq Frank} \)

  if \( n_1 = \text{Fred} \), analogous

  if \( n_1 \neq \text{Frank, Fred} \), \( s \cup \{ \text{f(S) = n_1} \} \) is obv. inconsistent
Theorems in detail

- $\models$ is classical entailment
- $\models \approx$ is limited entailment
- $\sigma$ contains no $O, \neg K_k, \neg M_k$
- $\sigma^*$ removes belief levels
- $\sigma_k$ sets belief levels to $k$

### Soundness & Eventual Completeness

| $O \alpha \models \approx \sigma$ | $\implies$ | $O \alpha \models \models \sigma^*$ | if $\sigma$ without $\neg K_k, \neg M_k$
| $O \alpha \models \approx \sigma_k$ for some $k$ | $\iff$ | $O \alpha \models \models \sigma^*$ | if $\alpha, \sigma$ quantifier-free

### Complexity

| $O \alpha \models \approx \sigma$ is decidable |  |  |
| $O \alpha \models \approx \sigma_k$ is tractable in $O(2^k(|\alpha| + |\sigma|)^{k+3})$ | if $\alpha, \sigma$ quantifier-free |
Semantics in detail

- $(\neg) t = n$
- $(\alpha \lor \beta)$
- $\neg(\alpha \lor \beta)$
- $\exists x \alpha$
- $\neg \exists x \alpha$
- $\neg \neg \alpha$

- $K_0 \alpha$
- $K_{k+1} \alpha$
- $M_0 \alpha$
- $M_{k+1} \alpha$
- $O \alpha$
Semantics in detail

- $s \models (\neg) t = n$ iff $(\neg) t = n \in s$
- $s \models (\alpha \lor \beta)$ iff $(\alpha \lor \beta) \in s$ or $s \models \alpha$ or $s \models \beta$
- $s \models \neg(\alpha \lor \beta)$ iff $s \models \neg \alpha$ and $s \models \neg \beta$
- $s \models \exists x \alpha$ iff $s \models \alpha^x_n$ for some name $n$
- $s \models \neg \exists x \alpha$ iff $s \models \neg \alpha^x_n$ for every name $n$
- $s \models \neg \neg \alpha$ iff $s \models \alpha$

- $K_0 \alpha$
- $K_{k+1} \alpha$
- $M_0 \alpha$
- $M_{k+1} \alpha$
- $O \alpha$

\[ \hat{=} \text{ contains the empty clause} \]

\[ \hat{=} \text{ not potentially inconsistent} \]

\[ \hat{=} \text{ potentially inconsistent} \]

\[ \hat{=} (A) \text{ obviously consistent} \]

\[ \hat{=} (B) \text{ two unsubsumed clauses mention two complementary literals} \]

\[ \hat{=} (C) \text{ for every name } n, t \neq n \text{ occurs in an unsubsumed clause} \]
Semantics in detail

- $s \models (\neg) t = n$ iff $(\neg) t = n \in s$
- $s \models (\alpha \vee \beta)$ iff $(\alpha \vee \beta) \in s$ or $s \models \alpha$ or $s \models \beta$
- $s \models \neg (\alpha \vee \beta)$ iff $s \models \neg \alpha$ and $s \models \neg \beta$
- $s \models \exists x \alpha$ iff $s \models \alpha^x_n$ for some name $n$
- $s \models \neg \exists x \alpha$ iff $s \models \neg \alpha^x_n$ for every name $n$
- $s \models \neg \neg \alpha$ iff $s \models \alpha$

- $s \models K_0 \alpha$ iff $s$ is obviously inconsistent or $s \models \alpha$
- $s \models K_{k+1} \alpha$ iff for some $t$ and all $n$, $s \cup \{t = n\} \models K_k \alpha$
- $s \models M_0 \alpha$ iff $s$ is obviously consistent and $s \models \alpha$
- $s \models M_{k+1} \alpha$ iff for some $t$ and $n$, $s \cup \{t = n\} \models M_k \alpha$
- $s \models O \alpha$ iff $s$ is minimal s.t. $s \models \alpha$
Semantics in detail

- $s \models (\lnot) t = n$ if and only if $(\lnot) t = n \in s$
- $s \models (\alpha \lor \beta)$ if and only if $(\alpha \lor \beta) \in s$ or $s \models \alpha$ or $s \models \beta$
- $s \models \lnot(\alpha \lor \beta)$ if and only if $s \models \lnot \alpha$ and $s \models \lnot \beta$
- $s \models \exists x \alpha$ if and only if $s \models \alpha^x_n$ for some name $n$
- $s \models \lnot \exists x \alpha$ if and only if $s \models \lnot \alpha^x_n$ for every name $n$
- $s \models \lnot \lnot \alpha$ if and only if $s \models \alpha$
- $s \models K_0 \alpha$ if and only if $s$ is obviously inconsistent or $s \models \alpha$
- $s \models K_{k+1} \alpha$ if and only if for some $t$ and all $n$, $s \cup \{t = n\} \models K_k \alpha$
- $s \models M_0 \alpha$ if and only if $s$ is obviously consistent and $s \models \alpha$
- $s \models M_{k+1} \alpha$ if and only if for some $t$ and $n$, $s \cup \{t = n\} \models M_k \alpha$
- $s \models O \alpha$ if and only if $s$ is minimal s.t. $s \models \alpha$

Obviously inconsistent $\triangleleft$ contains the empty clause

Obviously consistent $\triangleleft$ not potentially inconsistent

Potentially inconsistent $\triangleleft$

(a) Obviously consistent

(b) Two unsubsumed clauses mention two complementary literals

(c) For every name $n$, $t \neq n$ occurs in an unsubsumed clause
 Semantics in detail

- $s_0, s, v \models (\neg t = n)$ iff $(\neg t = n) \in s$
- $s_0, s, v \models (\alpha \lor \beta)$ iff $(\alpha \lor \beta) \in s$ or $s_0, s, v \models \alpha$ or $s_0, s, v \models \beta$
- $s_0, s, v \models \neg (\alpha \lor \beta)$ iff $s_0, s, v \models \neg \alpha$ and $s_0, s, v \models \neg \beta$
- $s_0, s, v \models \exists x \alpha$ iff $s_0, s, v \models \alpha^x_n$ for some name $n$
- $s_0, s, v \models \neg \exists x \alpha$ iff $s_0, s, v \models \neg \alpha^x_n$ for every name $n$
- $s_0, s, v \models \neg \neg \alpha$ iff $s_0, s, v \models \alpha$

- $s_0, s, v \models K_0 \alpha$ iff $s_0 \cup v$ is obv. inconsistent or $s_0, s_0 \cup v, \emptyset \models \alpha$
- $s_0, s, v \models K_{k+1} \alpha$ iff for some $t$ and all $n$, $s_0, s, v \cup \{t = n\} \models K_k \alpha$
- $s_0, s, v \models M_0 \alpha$ iff $s_0 \cup v$ is obv. consistent and $s_0, s_0 \cup v, \emptyset \models \alpha$
- $s_0, s, v \models M_{k+1} \alpha$ iff for some $t$ and $n$, $s_0, s, v \cup \{t = n\} \models M_k \alpha$
- $s_0, s, v \models O \alpha$ iff $s_0$ is minimal s.t. $s_0, s_0, \emptyset \models \alpha$

obviously inconsistent $\triangleleft$ contains the empty clause
obviously consistent $\triangleleft$ not potentially inconsistent
potentially inconsistent $\triangleleft$

(a) obviously consistent
(b) two unsubsumed clauses mention two complementary literals
(c) for every name $n, t \neq n$ occurs in an unsubsumed clause