Knowledge Representation and Reasoning

Logics for Artificial Intelligence

Stuart C. Shapiro

Department of Computer Science and Engineering
and Center for Cognitive Science
University at Buffalo, The State University of New York
Buffalo, NY 14260-2000

`shapiro@cse.buffalo.edu`

Contents

3 Elementary Predicate Logic

3.1 CarPool World

Propositional Logic

$$\begin{array}{ll}
\textit{Tom drives Betty} & \textit{Betty drives Tom} \\
\textit{Tom is the driver} & \textit{Betty is the driver} \\
\textit{Tom is the passenger} & \textit{Betty is the passenger}
\end{array}$$

related only by the domain rules.

Predicate Logic

$$\begin{array}{ll}
Drives(Tom, Betty) & Drives(Betty, Tom) \\
Driver(Tom) & Driver(Betty) \\
Passenger(Tom) & Passenger(Betty)
\end{array}$$

shows two properties, one relation, and two individuals.

3.2 The "Standard" Elementary Predicate Logic

3.2.1 Syntax of the "Standard" Elementary Predicate Logic Atomic Symbols

Individual Constants:

- Any letter of the alphabet (preferably early),
- any (such) letter with a numeric subscript,
- any character string not containing blanks nor other punctuation marks.

For example: a, B_{12}, Tom, *Tom's_mother-in-law*.

Atomic Symbols, Part 2

Arbitrary Individuals: Look like individual constants.

Skolem Constants: Look like individual constants.

Variables:

- Any letter of the alphabet (preferably late),
- any (such) letter with a numeric subscript.

For example: u, v_6.

Atomic Symbols, Part 3

Predicate Symbols:

- Any letter of the alphabet (preferably late middle),
- any (such) letter with a numeric subscript,
- any character string not containing blanks.

For example: $P, Q_4, Drives$.

Each Predicate Symbol must have a particular **arity**.

Use superscript for explicit arity.

For example: $P^1, Drives^2, Q_2^3$

Atomic Symbols, Part 4

In any specific predicate logic language

Individual Constants,
Arbitrary Individuals,
Skolem Constants,
Variables,
Predicate Symbols

must be disjoint.

Terms

- Every individual constant, Arbitrary Individual, Skolem Constant, and variable is a term.

- Nothing else is a term.

Atomic Formulas

If P^n is a predicate symbol of arity n,

and $t_1, \ldots, t_n$ are terms,

then $P^n(t_1, \ldots, t_n)$ is an atomic formula.

E.g.: $\mathit{Passenger}^{\mathit{1}}(\mathit{Tom}), \mathit{Drives}^{\mathit{2}}(\mathit{Betty}, y)$

(The superscript may be omitted if no confusion results.)

Well-Formed Formulas (wffs):

Every atomic formula is a wff.

If P is a wff, then so is $(\neg P)$.

If P and Q are wffs, then so are
$(P \wedge Q)$ $(P \vee Q)$
$(P \Rightarrow Q)$ $(P \Leftrightarrow Q)$

If P is a wff and x is a variable,
then $\forall x(P)$ and $\exists x(P)$ are wffs.

Parentheses may be omitted or replaced by square brackets if no confusion results.

$\forall x(\forall y(P))$ may be abbreviated as $\forall x, y(P)$.
$\exists x(\exists y(P))$ may be abbreviated as $\exists x, y(P)$.

Quantifiers:

In $\forall x P$ and $\exists x P$

$\forall$ called the **universal quantifier**.

$\exists$ called the **existential quantifier**.

P is called the **scope** of quantification.

Free and Bound Variables

Every occurrence of x in P, not in the scope of some other occurrence of $\forall x$ or $\exists x$, is said to be **free** in P and **bound** in $\forall xP$ and $\exists xP$.

Every occurrence of every variable other than x that is free in P is also free in $\forall xP$ and $\exists xP$.

$$\forall x[P(x,y) \Leftrightarrow [(\exists x \exists z Q(x,y,z)) \Rightarrow R(x,y)]]$$

Open, Closed, and Ground

A wff with a free variable is called **open**.

A wff with no free variables is called **closed**,

An expression with no variables is called **ground**.

CarPool World Domain Rules

Propositional Logic

$$
\begin{array}{l}
Betty\ is\ the\ driver \Leftrightarrow \neg Betty\ is\ the\ passenger \\
Tom\ is\ the\ driver \Leftrightarrow \neg Tom\ is\ the\ passenger \\
Betty\ drives\ Tom \Rightarrow Betty\ is\ the\ driver \land Tom\ is\ the\ passenger \\
Tom\ drives\ Betty \Rightarrow Tom\ is\ the\ driver \land Betty\ is\ the\ passenger \\
Tom\ drives\ Betty \lor Betty\ drives\ Tom
\end{array}
$$

Predicate Logic

$$
\begin{array}{l}
\forall x(Driver(x) \Leftrightarrow \neg Passenger(x)) \\
\forall x, y(Drives(x, y) \Leftrightarrow (Driver(x) \land Passenger(y))) \\
Drives(Tom, Betty) \lor Drives(Betty, Tom)
\end{array}
$$

shows the generalization.

3.2.2 Substitutions

Syntax

Pairs: t/v

- t is any term
- v is any variable

Substitutions: $\{t_1/v_1, \ldots, t_n/v_n\}$

- $i \neq j \Rightarrow v_i \neq v_j$

Terminology

$\sigma = \{t_1/v_1, \ldots, t_n/v_n\}$

t_i is a **term** in σ

v_i is a **variable of** σ

Say $t_i/v_i \in \sigma$ and $v_i \in \sigma$,
but not $t_i \in \sigma$

Note: x is not a variable of $\{x/y\}$,
i.e. $x/y \in \{x/y\}, y \in \{x/y\}, x \notin \{x/y\}$

Substitution Application

For expression $\mathcal{A}$ and substitution $\sigma = \{t_1/v_1, \ldots, t_n/v_n\}$

$\mathcal{A}\sigma$: replace every free occurrence of each v_i in $\mathcal{A}$ by t_i

E.g.: $P(x,y)\{x/y, y/x\} = P(y,x)$

Substitution Composition

$P\sigma\tau = ((P\sigma)\tau) = P(\sigma \circ \tau)$

Let $\sigma = \{t_1/v_1, \ldots, t_n/v_n\}$

$\sigma \circ \tau = \{t_1\tau/v_1, \ldots, t_n\tau/v_n\} \uplus \tau$

$\sigma \uplus \tau = \sigma \cup \{t_i/v_i \mid (t_i/v_i \in \tau) \wedge v_i \notin \sigma\}$

E.g.: $\{x/y, y/z\} \circ \{u/y, v/w\} = \{x/y, u/z, v/w\}$

3.2.3 Semantics of the "Standard" Elementary Predicate Logic

Assumes a **Domain**, $\mathcal{D}$, of

- individuals,
- sets of individuals,
- relations on individuals

Let $\mathcal{I}$ be set of all individuals in $\mathcal{D}$.

Semantics of Constants

Individual Constants:

$[\![a]\!]$ is some particular individual in $\mathcal{I}$.

Quasi-Individual Constants:

Arbitrary Individuals: $[\![a]\!]$ is a representative of all individuals in $\mathcal{I}$. Everything True about all of them, is True of it.

Skolem Constants: $[\![s]\!]$ is a representative of some individual in $\mathcal{I}$, but it's unspecified which one.

Semantics of Predicate Symbols

Predicate Symbols:

- $[\![P^1]\!]$ is some particular subset of $\mathcal{I}$.
- $[\![P^n]\!]$ is some particular subset of the relation
$$\underbrace{\mathcal{I} \times \cdots \times \mathcal{I}}_{n \text{ times}}.$$

Semantics of Atomic Formulas

Atomic Formulas:

- If P^1 is some unary predicate symbol, and t is some individual constant, then $[\![P^1(t)]\!]$ is True if $[\![t]\!] \in [\![P^1]\!]$, and False otherwise.

- If P^n is some n-ary predicate symbol, and $t_1, \ldots, t_n$ are individual constants, then $[\![P^n(t_1, \ldots, t_n)]\!]$ is True if $\langle [\![t_1]\!], \ldots, [\![t_n]\!] \rangle \in [\![P^n]\!]$, and False otherwise.

Semantics of WFFs, Part 1

- $[\![\neg P]\!]$ is True if $[\![P]\!]$ is False. Otherwise, it is False.

- $[\![P \wedge Q]\!]$ is True if $[\![P]\!]$ is True and $[\![Q]\!]$ is True. Otherwise, it is False.

- $[\![P \vee Q]\!]$ is False if $[\![P]\!]$ is False and $[\![Q]\!]$ is False. Otherwise, it is True.

- $[\![P \Rightarrow Q]\!]$ is False if $[\![P]\!]$ is True and $[\![Q]\!]$ is False. Otherwise, it is True.

- $[\![P \Leftrightarrow Q]\!]$ is True if $[\![P]\!]$ and $[\![Q]\!]$ are both True, or both False. Otherwise, it is False.

Semantics of WFFs, Part 2

- $[\![\forall x P]\!]$ is True if $[\![P\{t/x\}]\!]$ is True for every individual constant, t. Otherwise, it is False.

- $[\![P(a)]\!]$, where a is an arbitrary individual is True if $[\![P\{t/a\}]\!]$ is True for every individual constant, t. Otherwise, it is False.

- $[\![\exists x P]\!]$ is True if there is some individual constant, t such that $[\![P\{t/x\}]\!]$ is True. Otherwise, it is False.

- $[\![P(s)]\!]$, where s is a Skolem constant is True if there is some individual constant, t such that $[\![P\{t/s\}]\!]$ is True. Otherwise, it is False.

Intensional Semantics of 2-Car CarPool World 1

Individual Constants:

$[Tom]$ = Someone named Tom.
$[Betty]$ = Someone named Betty.
$[John]$ = Someone named John.
$[Jane]$ = Someone named Jane.

Intensional Semantics of 2-Car CarPool World 2

Predicate Symbols:

$[Driver^{1}(x)] = [x]$ is the driver of the car.
$[Passenger^{1}(x)] = [x]$ is the passenger of the car.
$[Drives^{2}(x, y)] = [x]$ drives $[y]$ to work.

Extensional Semantics of a 2-Car CarPool World Situation

$[\![Tom]\!] =$ Tom.
$[\![Betty]\!] =$ Betty.
$[\![John]\!] =$ John.
$[\![Jane]\!] =$ Jane.
$[\![Driver]\!] = \{Betty, John\}$.
$[\![Passenger]\!] = \{Jane, Tom\}$.
$[\![Drives]\!] = \{\langle Betty, Tom\rangle, \langle John, Jane\rangle\}$.

3.2.4 Hilbert-Style Axioms for Elementary Predicate Logic

(A1). $(\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{A}))$

(A2). $((\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C})))$

(A3). $((\neg\mathcal{B} \Rightarrow \neg\mathcal{A}) \Rightarrow ((\neg\mathcal{B} \Rightarrow \mathcal{A}) \Rightarrow \mathcal{B}))$

(A4). $\forall x\mathcal{A} \Rightarrow \mathcal{A}\{t/x\}$
where t is any term free for x in $A(x)$.

(A5). $(\forall x(\mathcal{A} \Rightarrow \mathcal{B})) \Rightarrow (\mathcal{A} \Rightarrow \forall x\mathcal{B})$
if $\mathcal{A}$ is a wff containing no free occurrences of x.

Hilbert-Style Rules of Inference for "Standard" Elementary Predicate Logic

$$\frac{\mathcal{A}, \mathcal{A} \Rightarrow \mathcal{B}}{\mathcal{B}}$$

$$\frac{\mathcal{A}}{\forall x \mathcal{A}}$$

Note: $\exists x \mathcal{A}$ is just an abbreviation of $\neg \forall x \neg \mathcal{A}$.

Fitch-Style Natural Deduction

4 Additional Rules of Inference*

$$\begin{array}{rl|l}
i. & \begin{array}{|l} \underline{a\quad\quad} \\ \vdots \\ A \end{array} & ArbI \\
j. & & \\
j+1. & \forall x A\{x/a\} & \forall I, i\text{–}j
\end{array}$$

$$\begin{array}{r|ll}
i. & \forall x A & \\
i+1. & A\{t/x\} & \forall E, i
\end{array}$$

$$\begin{array}{r|ll}
i. & A(t) & \\
i+1. & \exists x A(x) & \exists I, i
\end{array}$$

$$\begin{array}{r|ll}
i. & \exists x A & \\
i+1. & A\{s/x\} & \exists E, j
\end{array}$$

Where a is an arbitrary individual not otherwise used in the proof, and s is a Skolem constant not otherwise used in the proof,

*Adapted from D. Kalish and R. Montague, *Logic: Techniques of Formal Reasoning*, (New York: Harcourt, Brace & World), 1964, pp. 99–101.

Example Fitch-Style Proof

1.	$\neg\forall x A(x)$	Hyp
2.	$\neg\exists x \neg A(x)$	Hyp
3.	a_1	$ArbI$
4.	$\neg A(a_1)$	Hyp
5.	$\exists x \neg A(x)$	$\exists I, 4$
6.	$\neg\exists x \neg A(x)$	$Reit, 2$
7.	$\neg\neg A(a_1)$	$\neg I, 4–6$
8.	$A(a_1)$	$\neg E, 7$
9.	$\forall x A(x)$	$\forall I, 3–8$
10.	$\neg\forall x A(x)$	$Reit, 1$
11.	$\neg\neg\exists x \neg A(x)$	$\neg, 2–10$
12.	$\exists x \neg A(x)$	$\neg E, 11$
13.	$\neg\forall x A(x) \Rightarrow \exists x \neg A(x)$	$\Rightarrow I, 1–12$

CarPool Situation Derivation

1.	$\forall x(Driver(x) \Leftrightarrow \neg Passenger(x))$ $\wedge \forall x, y(Drives(x,y) \Leftrightarrow (Driver(x) \wedge Passenger(y)))$ $\wedge Drives(Betty, Tom)$	Hyp
2.	$Drives(Betty, Tom)$	$\wedge E, 1$
3.	$\forall x, y(Drives(x,y) \Leftrightarrow (Driver(x) \wedge Passenger(y)))$	$\wedge E, 1$
4.	$\forall y(Drives(Betty,y) \Leftrightarrow (Driver(Betty) \wedge Passenger(y)))$	$\forall E, 3$
5.	$Drives(Betty, Tom) \Leftrightarrow (Driver(Betty) \wedge Passenger(Tom))$	$\forall E, 4$
6.	$Driver(Betty) \wedge Passenger(Tom)$	$\Leftrightarrow E, 2,$
7.	$Driver(Betty)$	$\wedge E, 6$
8.	$\forall x(Driver(x) \Leftrightarrow \neg Passenger(x))$	$\wedge E, 1$
9.	$Driver(Betty) \Leftrightarrow \neg Passenger(Betty)$	$\forall E, 8$
10.	$\neg Passenger(Betty)$	$\Leftrightarrow E, 7,$

3.2.5 Model Checking in Elementary Predicate Logic

- n Individual Constants.

- Predicate P^j yields n^j ground atomic propositions.

- k_j predicates of arity j yields $\sum_j (k_j \times n^j)$ ground atomic propositions.

- So $2^{\sum_j (k_j \times n^j)}$ situations (columns of truth table).

- Carpool World has $2^{(2\times 2^1 + 1\times 2^2)} = 2^8 = 256$ situations.

CarPool World Situations

$Driver(Tom)$	T	T	F
$Driver(Betty)$	T	F	T
$Passenger(Tom)$	T	F	T
$Passenger(Betty)$	T	T	F
$Drives(Tom, Tom)$	T	F	F
$Drives(Tom, Betty)$	T	T	T
$Drives(Betty, Tom)$	T	T	T
$Drives(Betty, Betty)$	T	F	F
$\forall x(Driver(x) \Leftrightarrow \neg Passenger(x))$	F	T	T
$\forall x \forall y(Drives(x, y) \Rightarrow (Driver(x) \Leftrightarrow Passenger(y)))$	T	T	T
$Drives(Tom, Betty)$	T	T	T
$Drives(Betty, Tom)$	T	T	T

Feasability of Wang's Algorithm

- n Individual Constants.
- $\mathcal{A}$ has k quantified variables.
- $\mathcal{A} \Leftrightarrow$ Boolean combination of n^k ground wffs.
- E.g.:

$$
\begin{aligned}
&\forall x \exists y (Drives(x, y)) \\
&\Leftrightarrow \exists y Drives(Tom, y) \land \exists y Drives(Betty, y) \\
&\Leftrightarrow (Drives(Tom, Tom) \lor Drives(Tom, Betty)) \\
&\quad \land (Drives(Betty, Tom) \lor Drives(Betty, Betty))
\end{aligned}
$$

Use of Predicate-Wang, Input

```
wang(4): (predicate-wang
          '( (all x
                  (all y
                       ((Drives x y)
                        => ((Driver x) & (Passenger y)))))
            (Drives Betty Tom))
          '( ((Driver Betty) & (Passenger Tom))))
```

Use of Predicate-Wang, Output

```
 0[1]: (wang (((((Drives Tom Tom)
                => ((Driver Tom) & (Passenger Tom)))
               & ((Drives Tom Betty)
                  => ((Driver Tom) & (Passenger Betty))))
              & (((Drives Betty Tom)
                  => ((Driver Betty) & (Passenger Tom)))
                 & ((Drives Betty Betty)
                    => ((Driver Betty) & (Passenger Betty)))))
             (Drives Betty Tom))
            (((Driver Betty) & (Passenger Tom))))
...
0[1]: returned t
t
```

3.3 Clause Form
Elementary Predicate Logic

3.3.1 Syntax of Clause Form Elementary Predicate Logic

Individual constants, Skolem constants, variables, predicate symbols, terms, and atomic formulas as in standard elementary predicate logic.
(Arbitrary individuals are not needed.)

Literals and clauses as in propositional clause form logic.

3.3.2 Semantics of Clause Form Elementary Predicate Logic

- Individual constants, Skolem constants, predicate symbols, terms, and ground atomic formulas as in standard predicate logic.
  (Arbitrary individuals are not needed.)

- Ground literals, ground clauses, and sets of clauses as in propositional clause form logic.

Semantics of Open Clauses

If clause C contains a Skolem constant s,
then C is True if there is some individual constant t such that $C\{t/s\}$ is True, and False otherwise.

If clause C contains a variable v,
then C is True if for every individual constant t $C\{t/v\}$ is True, and False otherwise.

Translation of Standard Form to Clause Form Elementary Predicate Calculus

1. Eliminate quantifiers as when using model checking.

2. Translate into clause form as for propositional logic.

GSAT, A Procedure for Finding Satisfying Models (B&L, p. 82)

input: a set of clauses C, and parameters *tries* and *flips*
output: an interpretation satisfying C, or failure
procedure GSAT(C, *tries*, *flips*)
for $i := 1$ **to** *tries* **do**
    $\mathcal{I} :=$ a randomly generated truth assignment
    for $j := 1$ **to** *flips* **do**
        if $\mathcal{I} \models C$ **then return** $\mathcal{I}$
        $p :=$ an atom such that a change in its truth
            assignment gives the largest increase in the total
            number of clauses in C that are satisfied by $\mathcal{I}$
        $\mathcal{I} := \mathcal{I}$ with the truth assignment of p reversed
    end for end for
return "no satisfying interpretation found"