Система FORMAL, обучающая языку логики предикатов первого порядка

И.В. Горская

Разработана система, обучающая представлению знаний на языке логики предикатов. В основе функционирования системы лежат процедуры преобразования формул логики предикатов к стандартному ("читабельному") виду и сравнения эквивалентных относительно специальной системы преобразований формул.

Введение

Система FORMAL обучает представлению знаний в терминах языка логики предикатов первого порядка. Процесс обучения предполагает решение определенной последовательности задач. Каждая задача состоит в формализации, т.е. представлении логической формулой, одного естественно-языкового суждения. А именно, система предлагает русский (английский) текст суждения, сопровождаемый списком необходимых предикатных и функциональных символов, и приглашает пользователя ввести формулу, выражающую, по его мнению, смысл текста. Проверив правильность ответа, система сообщает результат проверки, и если ответ неверный, предпринимает попытку объяснить разницу между смыслом введенной формулы и заданного суждения путем перевода предложенного ответа на русский (английский) язык. А если пользователь не может дать никакого ответа, система предлагает в качестве подсказки более понятный вариант текста того же самого логического суждения.

В отличие от широко известной системы TARSKI'S WORLD (Мир Тарского) [1], также предназначенной для обучения языку первого порядка, предлагаемая система FORMAL ориентирована на другой класс задач и другие методы обучения. А именно, TARSKI'S WORLD использует язык логики первого порядка для описания различных отношений между демонстрируемыми на экране кубами, тетраэдрами и додекаэдрами, и ставит целью создание представления о логическом следовании, тогда как FORMAL обучает переводу естественно-языковых текстов в логические формулы. Методика обучения основана на новом представлении, связанном

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 93-01-16006).

с семантикой логических формул, согласно которому все формулы делятся на "читабельные" и "нечитабельные", в том смысле, что первые можно правильно понять при "буквальном прочтении", а вторые — нельзя. Поэтому в системе FORMAL предлагаемые пользователем формулы сначала приводятся к "читабельному" виду, а затем преобразуются в естественно-языковые тексты, что позволяет обнаруживать и исправлять некоторые типичные ошибки. Другими словами, предполагается, что правильно представить логическое суждение формулой означает найти такую "читабельную" формулу, буквальный перевод которой на естественный язык имеет тот же смысл, что исходное суждение. Поэтому принятая в системе FORMAL система подсказок представляет собой последовательность синонимичных естественно-языковых текстов, начинающуюся с исходного суждения и постепенно приближающуюся к "буквальному прочтению" соответствующей ему формулы. Для сравнения предложенного ответа с образцом и для построения естественно-языкового текста по формуле в системе FORMAL используется специальный язык схем формул.

1. "Читабельные" и "нечитабельные" формулы

Рассмотрим решение задач типа "представить суждение в виде формулы" на примере суждения "*На любой прямой существует точка*" и "словаря"

$$Line(x) = "x \text{ — прямая}";$$
$$Point(x) = "x \text{ — точка}";$$
$$Lie(x,y) = "x \text{ лежит на } y".$$

Нетрудно видеть, что в данном случае подходит формула

$$\forall x(Line(x) \to \exists y(Point(y) \land Lie(y,x))). \qquad \text{(A)}$$

Труднее объяснить, почему это же суждение не может быть представлено в виде

$$\forall x(Line(x) \land \exists y(Point(y) \land Lie(y,x)) \qquad \text{(B)}$$

или

$$\forall x(Line(x) \to \exists y(Point(y) \to Lie(y,x)). \qquad \text{(C)}$$

Например, можно рассуждать так: суждение, выражаемое формулой B, в отличие от исходного, ложно в Евклидовой геометрии, поскольку утверждает, что "все объекты — прямые": "*Каков бы ни был x, x — прямая и ...*". А для того, чтобы выявить разницу между суждениями, соответствующими A и C, удобнее всего преобразовать формулу C сначала в

$$\forall x(Line(x) \to (\forall y Point(y) \to \exists y Lie(y,x)),$$

а затем в формулу

$$\forall y Point(y) \to \forall x(Line(x) \to \exists y Lie(y,x)). \qquad \text{(D)}$$

Последняя формула означает "*Если все объекты точки, то на любой прямой что-нибудь лежит*" и явным образом отличается как от заданного суждения, так и от формулы A.

Итак, предлагаемый метод объяснения несоответствия между формулами и суждениями состоит в "буквальном прочтении" рассматриваемых формул и сравнении исходных и полученных суждений между собой. В таком случае возможен вопрос: почему "прочтение" полученной в результате эквивалентных преобразований формулы D позволяет отличить ее смысл от смысла заданного текста, а "прочтение" C с этой точки зрения бесполезно: "*Каков бы ни был x, если x — прямая, то существует y, такой что если y — точка, то y лежит на x?*"

Для обсуждения этого и связанных с ним вопросов полезно уточнить понятие "буквального прочтения", т.е. определить соответствующее отображение формул в тексты. Пусть язык логики предикатов первого порядка L имеет логические связки $\land, \lor, \to, \neg, \forall, \exists$ и конечную сигнатуру (множество предикатных и функциональных символов), и пусть V — словарь, представляющий собой множество пар из символов сигнатуры и подходящих фрагментов текста, например:

$$(Point(\_),\quad "\_ \text{ — точка}"),$$
$$(Lie(\_,\_),\quad "\_ \text{ лежит на } \_"),$$
$$(Sum(\_,\_),\quad "\text{сумма } \_ \text{ и } \_"), \quad \text{и т.д.}$$

И пусть A — замкнутая формула языка L. Тогда отображение $Text(A)$ определяется следующим образом ('+' означает конкатенацию):

1) $Text(x) = "x"$, если x — индивидная переменная;
2) $Text(h(t_1,\dots,t_n)) = w_0 + Text(t_1) + w_1 + \dots + Text(t_n) + w_n$, если в словаре V есть пара $(h(\_,\_,\dots,\_), w_0 + \_ + w_1 + \_ \cdots + \_ + w_n)$;
3) $Text((A \to B)) = "\text{если } " + Text(A) + " \text{ то } " + Text(B)$;
4) $Text((A \land B)) = Text(A) + " \text{ и } " + Text(B)$;
5) $Text((A \lor B)) =$
$"\text{верно хотя бы одно: либо } " + Text(A) + ", \text{ либо } " + Text(B)$;
6) $Text(\neg A) = "\text{не верно, что } " + Text(A)$;
7) $Text(\forall x A) = "\text{каков бы ни был } x, " + Text(A)$;
8) $Text(\exists x A) = "\text{существует } x, \text{ такой что } " + Text(A)$;

Кроме того, следует отметить, что рассматриваемый метод объяснения неявно основывается на двух предположениях:

- для того, чтобы текст и замкнутая формула языка L выражали одно и то же суждение, они должны быть одновременно истинными или ложными в произвольной интерпретации (при условии, что словарь V фиксирован);
- все классически эквивалентные замкнутые формулы языка L выражают одно и то же суждение.

Делая эти предположения явными, введем обозначение $\models X \cong Y$ для неформального соотношения "X и Y, во-первых, принадлежат объединению множеств замкнутых формул и текстов, и, во-вторых, при одинаковом понимании выражений, входящих в одну пару словаря V, получают одинаковое истинностное значение в любой интерпретации". Например, если пары

$$(Red(\_),\ \text{"\_ красный "}),$$
$$(Square(\_),\ \text{"\_ — квадрат "}).$$

содержатся в словаре V, то

$\models \exists x(Red(x) \wedge Squre(x)) \cong$

"Существует x, такой что x красный и x — квадрат";

$\models \forall x(Red(x) \rightarrow Square(x)) \cong$

"Каков бы ни был x, если x красный, то x — квадрат".

Теперь можно сформулировать следующий вопрос: верно ли, что утверждение $\models X \cong Text(X)$ истинно для любой замкнутой формулы A?

В качестве примера рассмотрим формулу

$$\exists x(\neg Square(x) \rightarrow Square(x)) \tag{E}$$

и соответствующий ей текст $Text(E) =$ *"Существует x, такой что если не верно, что x — квадрат, то x — квадрат"*: выражаемое ими суждение легко может показаться ложным. Но формулу E можно эквивалентным образом преобразовать в

$$\exists x Square(x),$$

и тогда станет ясно, что она истинна на любом множестве объектов, содержащих "квадраты". А если V содержит пары

$$(RAngles(\_),\ \text{"все углы \_ — прямые "}),$$
$$(AAngles(\_),\ \text{"все углы \_ — острые "}),$$

и рассматривается формула

$$\forall x(RAngles(x) \vee (Square(x) \rightarrow AAngles(x))), \tag{F}$$

то, по-видимому, следует считать, что соотношение

$\models F \cong$ *"Каков бы ни был x, верно хотя бы одно: либо все углы x — прямые, либо, если x — квадрат, то все углы x — острые"*,

не выполнено, поскольку F формула эквивалентна следующей:

$$\forall x(Square(x) \rightarrow (RAngles(x) \vee AAngles(x))).$$

И наконец, формулы

$$\forall x((\neg BLine(x,y) \vee BPlane(x,z)) \rightarrow BPlane(x,z)),$$
$$\forall x((BLine(x,y) \rightarrow BPlane(x,z)) \rightarrow BPlane(x,z)) \tag{G}$$

эквивалентны, поэтому если пары

$$(BLine(\_,\_),\ \text{"\_ принадлежит прямой \_ "}),$$
$$(BPlane(\_,\_),\ \text{"\_ принадлежит плоскости \_ "})$$

присутствуют в словаре V, то не выполнено соотношение

$\models G \cong$ *"Каков бы ни был x, если из $'x$ принадлежит прямой y' следует $'x$ принадлежит плоскости z', то x принадлежит плоскости z."*

Таким образом, выявлен ряд "нечитабельных" формул, поэтому "читабельными" можно считать только те формулы, которые не имеют "нечитабельных" подформул. Другими словами, формула

$$(A_1 \rightarrow (A_2 \rightarrow \cdots \rightarrow A_m)\ldots), \quad m \geq 1,$$

является "читабельной", если все ее подформулы $A_i (1 \leq i \leq m)$ являются R-формулами, т.е. имеют следующую структуру:

1) атомарная формула;
2) $\neg C$, где C — R-формула;
3) $C \wedge D$, где C и D — R-формулы;
4) $C \vee D$, где C и D — R-формулы;
5) $\exists x C$, где C — R-формула;
6) $\forall x_1 \ldots \forall x_n (C_1 \rightarrow (C_2 \rightarrow \cdots \rightarrow C_m)\ldots)$, где $C_1, \ldots, C_m$ — R-формулы, каждая из которых имеет свободные переменные $x_1, \ldots, x_n$ ($m \geq 1$, $n \geq 1$).

В системе FORMAL в качестве "читабельных" рассматриваются формулы еще более узкого класса, включающего только дизъюнкции и конъюнкции $R1$-формул следующего вида:

1) A или $\neg A$, где A — атомарная формула;
2) $\exists x C(x)$ или $\neg \exists x C(x)$, где $C(x)$ — конъюнкция $R1$-формул, каждая из которых содержит свободную переменную x;
3) $\forall x(C_1(x) \rightarrow \ldots (C_n(x) \rightarrow D(x))\ldots)$ или $\neg \forall x(C_1(x) \rightarrow \ldots (C_n(x) \rightarrow D(x))\ldots)$, где для каждого i $(1 \leq i \leq n,\ n \geq 0)$ $C_i(x)$ — конъюнкция, а $D(x)$ — дизъюнкция $R1$-формул, каждая из которых содержит x свободно.

Теорема 1. *Для каждой формулы A языка L существует формула B, такая что $\models A \leftrightarrow B$ и B есть дизъюнкция конъюнкций (конъюнкция дизъюнкций) $R1$-формул.*

Доказательство. Формулу A можно превратить в B при помощи следующих эквивалентных преобразований:

1) $(A * B) \leftrightarrow (B * A)$;
2) $(A * (B * C)) \leftrightarrow ((A * B) * C)$;
3) $(A * (B + C)) \leftrightarrow ((A * B) + (A * C)$;
4) $\neg(A + B) \leftrightarrow (\neg A * \neg B)$;
5) $(A \to B) \leftrightarrow (\neg A \vee B)$;
6) $\neg\neg A \leftrightarrow A$;
7) $\neg QxA \leftrightarrow Gx\neg A$;
8) $Qx(A(-x) * B(x)) \leftrightarrow (A(-x) * QxB(x))$;
9) $\forall x(A(x) \wedge B(x)) \leftrightarrow (\forall xA(x) \wedge \forall xB(x))$;
10) $\exists x(A(x) \vee B(x)) \leftrightarrow (\exists xA(x) \vee \exists xB(x))$; где либо $+ = \wedge$ и $* = \vee$, либо $+ = \vee$ и $* = \wedge$, либо $Q = \forall$ и $G = \exists$, либо $Q = \exists$ и $G = \forall$, $A(x)$ = 'формула A имеет свободные вхождения переменной x', $A(-x)$ = 'формула A не имеет свободных вхождений переменной x'.

Именно эти эквивалентные преобразования используются в системе FORMAL для приведения формул к "читабельному" виду.

2. Система подсказок

Итак, установлено, что формулы $\exists x(A \to B)$, $((A \to B) \to C)$, $((A \to B) \vee C)$, и др., "нечитабельны" в том смысле, что для них не выполнено соотношение $\models Text(X) \cong X$. С другой стороны, оказывается, что если взять какие-либо книги, в которых язык первого порядка используется для представления содержательных математических суждений (например, [4, 5]), то нетрудно заметить, что "нечитабельные" формулы в них отсутствуют. А это значит, что их авторы для выражения смысла какого-либо суждения X привлекают всякий раз формулу Y, обладающую свойством $\models X \cong Y$ и $\models Text(Y) \cong Y$, или, в другой записи, $\models X \cong Text(Y)$.

Таким образом, для преобразования текста X в формулу подбирается такая "читабельная" формула, "прочтение" которой имеет тот же смысл, что и исходный текст. Поэтому в системе FORMAL в качестве "подсказок" используются естественно-языковые варианты одного и того же логического суждения, постепенно приближающиеся к "буквальному прочтению" формулы, выражающей смысл этого суждения: $X = T_1, T_2, \ldots, T_n = Text(Y)$, где Y — образец правильного ответа.

3. Сравнение ответа с образцом

В системе FORMAL верификация ответа происходит в три этапа, а именно, проверяется, является ли введенная формула

- (синтаксически) правильно построенной,
- 'читабельной' (с приведением к "читабельному" виду, если необходимо),
- соответствующей исходному суждению.

Последняя проверка выполняется путем сопоставления формулы-ответа с образцом: они должны соответствовать друг другу с точностью до следующих эквивалентных преобразований:

1) $\neg\neg A \leftrightarrow A$;
2) $(A \to B) \leftrightarrow (\neg A \vee B)$;
3) $(A * B) \leftrightarrow (B * A)$, где $* = \vee$ или $* = \wedge$;
4) $(A * (B * C)) \leftrightarrow ((A * B) * C)$, где $* = \vee$ или $* = \wedge$;
5) $\neg(A + B) \leftrightarrow (\neg A * \neg B)$, где либо $+ = \wedge$ и $* = \vee$, либо $+ = \vee$ и $* = \wedge$;
6) $\neg QxA \leftrightarrow Gx\neg A$, где либо $Q = \forall$ и $G = \exists$, либо $Q = \exists$ и $G = \forall$;
7) $Qx(A * B(x)) \leftrightarrow (A * QxB(x))$, где либо $Q = \forall$ и $* = \vee$, либо $Q = \exists$ и $* = \wedge$, и формула $B(x)$ имеет свободные вхождения переменной x, а формула A — нет;
8) $QxQyA \leftrightarrow QyQxA$, где $Q = \forall$ или $Q = \exists$;
9) $QxA(x) \leftrightarrow QyA(y)$, где $Q = \forall$ или $Q = \exists$.

Для вышеописанного сравнения используется язык схем формул, определяемый в терминах "наборов": набор представляет собой множество с повторными вхождениями элементов, причем конструкция

$$P(m) = (P_1, P_2, \ldots, P_m)$$

означает m-элементный набор. Объединение наборов

$$P(k),\ Q(m),\ \ldots,\ R(n),$$

понимаемое как

$$(P_1, \ldots, P_k, Q_1, \ldots, Q_m, \ldots, R_1, \ldots, R_n),$$

обозначается $P(k) \cup Q(m) \cup \cdots \cup R(n)$.

В таком случае схема S есть набор $x(k) \cup A(m) \cup P(n)$, в котором $x(k)$ — набор индивидных переменных, $A(m)$ — набор атомарных формул, а набор $P(n)$ состоит из выражений вида $P_i = \neg Q_i$, в каждом из которых Q_i — схема.

Формулу языка первого порядка L будем называть правильной, если в ней никакие два квантора не связывают одинаковые переменные, и никакая переменная не имеет свободных и связанных вхождений одновременно.

Пусть A и B — произвольные правильные формулы, а E — атомарная формула. Ниже описывается отображение Sc множества правильных формул в множество схем.

0)	$Sc(E) = E$;	0^*)	$Sc^*(E) = \neg E$;
1)	$Sc(A \vee B) = Sc(A) \cup Sc(B)$;	1^*)	$Sc^*(A \wedge B) = Sc^*(A) \cup Sc^*(B)$;
2)	$Sc(\forall xA) = (x) \cup Sc(A)$;	2^*)	$Sc^*(\forall xA) = (x) \cup Sc^*(A)$;
3)	$Sc(\neg A) = Sc^*(A)$;	3^*)	$Sc^*(\neg A) = Sc(A)$;
4)	$Sc(A \wedge B) = \neg Sc^*(A \wedge B)$;	4^*)	$Sc^*(A \vee B) = \neg Sc(A \vee B)$;
5)	$Sc(\exists xA) = \neg Sc^*(\exists xA)$;	5^*)	$Sc^*(\exists xA) = \neg Sc(\exists xA)$.

Теорема 2. *Пусть A и B — регулярные формулы. Тогда $Sc(A) = Sc(B)$, если и только если A может быть преобразована в B путем применения эквивалентных преобразований (1)–(8).*

Доказательство см. в [3].

С учетом вышеизложенного ясно, почему в системе FORMAL сравнение введенной формулы с образцом осуществляется путем сранения их схем с точностью до переобозначения связанных переменных.

4. Построение "хорошего" текста по формуле

Если ответ, предложенный пользователем, не соответствует образцу, то система FORMAL, чтобы помочь обнаружить ошибку, преобразует его в дизъюнкцию конъюнкций $R1$-формул и переводит на естественный язык.

Для построения естественно-языкового текста по "читабельной" формуле Y в естественно-языковый текст можно воспользоваться отображением $Text(Y)$, но при этом обычно получаются слишком громоздкие тексты. Чтобы сделать их короче, удобно подобрать подходящую формулу Z, эквивалентную Y, и сократить Z при помощи следующих двух правил:

$$1)\ \forall x_1 \ldots \forall x_k (A \to B) \quad \Rightarrow \quad \forall x^k(A)\ B,$$

$$2)\ \exists x_1 \ldots \exists x_k (A \wedge B) \quad \Rightarrow \quad \exists x^k(A)\ B.$$

Если Q означает $\forall$ или $\exists$, то "буквальное прочтение" новых конструкций состоит в следующем: $Text(Qx^k(A)\ B)$ = "для любого / некоторого x, такого что " $+Text(A)+$ ", " $+\,Text(B)$.

Для выбора формулы Z и ее сокращения в системе FORMAL также используется язык схем формул. А именно, пусть $S = Sc(Y)$ имеет вид $x(k) \cup A(l) \cup B(m) \cup C(n)$, где $x(k)$ — набор индивидных переменных, $A(l)$ — набор атомарных формул, $B(m)$ — набор отрицаний атомарных формул, и $C(n)$ — набор выражений $C_i = \neg Q_i$, в каждом из которых Q_i схема, отличная от атомарной формулы. В работе [3] доказано, что если $S = Sc(Y)$, где Y — $R1$-формула, то $k \geq 0$, $l+m+n > 0$, причем, если $k = 0$ то $l = 0$, $m = 0$ и $n = 1$. С учетом этого факта определяется отображение $Reduc$ схем $R1$-формул в язык L с сокращениями:

1) если $l = 0, n = 0$, то $k > 0, m > 0$, и $\quad Reduc(S) = \neg\exists x^k \bigwedge_{i=1}^{m} B_i;$

2) если $l = 0, n > 0$, то $k \geq 0, m \geq 0$, и

$$Reduc(S) = \forall x^k \left(\bigwedge_{i=1}^{m} B_i \right) \bigvee_{i=1}^{n} Reduc^{-}(C_i);$$

3) если $l > 0, n = 0, m = 0$, то $k > 0$, и $\quad Reduc(S) = \forall x^k \bigvee_{i=1}^{l} A_i;$

4) если $l > 0, m + n > 0$, то $k > 0$, и

$$Reduc(S) = \left(\bigwedge_{i=1}^{m} B_i \wedge \bigwedge_{i=1}^{n} Reduc^{+}(C_i) \right) \to \bigvee_{i=1}^{l} A_i;$$

1^+) если $l = 0, n = 0$, то $k > 0, m > 0$, и $\quad Reduc^{+}(S) = \neg\exists x^k \left(\bigwedge_{i=1}^{m-1} B_i \right) B_m;$

2^+) если $l > 0, n = 0$, то $k > 0, m \geq 0$, и

$$Reduc^{+}(S) = \forall x^k \left(\bigwedge_{i=1}^{m} B_i \wedge \bigwedge_{i=1}^{l-1} \neg A_i \right) A_l;$$

3^+) если $n > 0$, то $k \geq 0, l \geq 0, m \geq 0$, и

$$Reduc^{+}(S) = \forall x^k \left(\bigwedge_{i=1}^{m} B_i \wedge \bigwedge_{i=1}^{l} \neg A_i \wedge \bigwedge_{i=1}^{n-1} Reduc^{+}(C_i) \right) Reduc^{-}(C_n);$$

1^-) если $l = 0, n = 0$, то $k > 0, m > 0$, и $\quad Reduc^{-}(S) = \exists x^k \left(\bigwedge_{i=1}^{m-1} B_i \right) B_m;$

2^-) если $l > 0, n = 0$, то $k > 0, m \geq 0$, и

$$Reduc^{-}(S) = \exists x^k \left(\bigwedge_{i=1}^{m} B_i \wedge \bigwedge_{i=1}^{l-1} \neg A_i \right) A_l;$$

3^-) если $n > 0$, то $k \geq 0, l \geq 0, m \geq 0$, и

$$Reduc^{-}(S) = \exists x^k \left(\bigwedge_{i=1}^{m} B_i \wedge \bigwedge_{i=1}^{l} \neg A_i \wedge \bigwedge_{i=1}^{n-1} Reduc^{+}(C_i) \right) Reduc^{+}(C_n).$$

Используемый в системе FORMAL алгоритм преобразования $R1$-формул в тексты лишь немного сложнее отображения $Reduc$ с последующим "буквальным прочтением".

Заключение

Таким образом, система FORMAL не только проверяет правильность ответов, но объясняет ошибки и помогает в поиске правильного ответа, причем все ее действия основаны на целостном представлении о природе преобразования

ЕСТЕСТВЕННО-ЯЗЫКОВОЕ СУЖДЕНИЕ $\longrightarrow$ ФОРМУЛА.

Список литературы

1. Goldson D., Reeves S. Using programs to teach logic to computer scientists // Notes of the American Mathematical Society. — 1993. — Vol. 40, No. 2.
2. Горская И.В. Система "FORMAL", обучающая формализации знаний / Препринт Ин. прикл. матем. им. М.В. Келдыша РАН. — 1992. — № 106.
3. Горская И.В. Естественные логические формулы и выводы // Математические вопросы кибернетики. — Вып. 4. — М.: Наука, 1992.
4. Клини С., Весли Р. Основания интуиционистской математики. — М.: Наука, 1978.
5. Мендельсон Э. Введение в математическую логику. — М.: Наука, 1976.