人工智能基础 HW3

PB18111697 王章瀚

2021 年 5 月 1 日

7.13

This exercise looks into the relationship between clauses and implication sentences.

a.

Show that the clause $(\neg P_1 \vee \cdots \vee \neg P_m \vee Q)$ **is logically equivalent to the implication sentence** $(P_1 \wedge \cdots \wedge P_m) \Rightarrow Q$

Proof.

$$
\begin{aligned}
(P_1 \wedge \cdots \wedge P_m) \Rightarrow Q &\Leftrightarrow \neg(P_1 \wedge \cdots \wedge P_m) \vee Q \\
&\Leftrightarrow (\neg P_1 \vee \cdots \vee \neg P_m) \vee Q \\
&\Leftrightarrow \neg P_1 \vee \cdots \vee \neg P_m \vee Q
\end{aligned}
$$

$\square$

b.

Show that every clause (regardless of the number of positive literals) can be written in the form $(P_1 \wedge \cdots \wedge P_m) \Rightarrow (Q_1 \vee \cdots \vee Q_n)$**, where the** P_s **and** Q_s **are proposition symbols. A knowledge base consisting of such sentences is in implicative normal form or Kowalski form (Kowalski, 1979).**

I first show that $(P_1 \wedge \cdots \wedge P_m) \Rightarrow (Q_1 \vee \cdots \vee Q_n)$ is equivalent to $\neg P_1 \vee \cdots \vee \neg P_m \vee Q_1 \vee \cdots \vee Q_n$:

$$
\begin{aligned}
(P_1 \wedge \cdots \wedge P_m) \Rightarrow (Q_1 \vee \cdots \vee Q_n) &\Leftrightarrow \neg(P_1 \wedge \cdots \wedge P_m) \vee (Q_1 \vee \cdots \vee Q_n) \\
&\Leftrightarrow (\neg P_1 \vee \cdots \vee \neg P_m) \vee (Q_1 \vee \cdots \vee Q_n) \\
&\Leftrightarrow \neg P_1 \vee \cdots \vee \neg P_m \vee Q_1 \vee \cdots \vee Q_n
\end{aligned}
$$

Secondly, each clause can be written in the form of $\neg P_1 \vee \cdots \vee \neg P_m \vee Q_1 \vee \cdots \vee Q_n$.
Thus, every clause can be written in the form of $(P_1 \wedge \cdots \wedge P_m) \Rightarrow (Q_1 \vee \cdots \vee Q_n)$

c.

Write down the full resolution rule for sentences in implicative normal form.

The resolution rule is:

$$\frac{l_1 \vee \cdots \vee l_k, \qquad m_1 \vee \cdots \vee m_n}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

where l_i and m_j are complementary literals.
This can be rewritten as:

$$\frac{\neg P_1 \vee \cdots \vee \neg P_m \vee Q_1 \vee \cdots \vee Q_n, \qquad \neg R_1 \vee \cdots \vee \neg R_k \vee S_1 \vee \cdots \vee S_l}{P_1 \vee \cdots \vee P_{i-1} \vee P_{i+1} \vee \cdots \vee P_m \vee Q_1 \vee \cdots \vee Q_n \vee \neg R_1 \vee \cdots \vee \neg R_k S_1 \vee \cdots \vee S_{j-1} \vee S_{j+1} \vee \cdots \vee S_l}$$

where $P_i = S_j$. According to (b), the rule above can be written as:

$$\frac{(P_1 \wedge \cdots \wedge P_m) \Rightarrow (Q_1 \vee \cdots \vee Q_n), \qquad (R_1 \wedge \cdots \wedge R_k) \Rightarrow (S_1 \vee \cdots \vee S_l)}{P_1 \vee \cdots \vee P_{i-1} \vee P_{i+1} \vee \cdots \vee P_m \vee Q_1 \vee \cdots \vee Q_n \vee \neg R_1 \vee \cdots \vee \neg R_k \vee S_1 \vee \cdots \vee S_{j-1} \vee S_{j+1} \vee \cdots \vee S_l}$$

Proof

Prove the completeness of the forward chaining algorithm.

1. 当 FC 到达不动点后, 不会再有新的推理
2. 考虑算法伪代码中的 inferred 表的最终状态, 参与推理过程的均为 true, 否则为 false. 可以把这个推理表看成一个模型 M,
3. 且原始 KB 中的每个子句在模型 M 中都为真. 其证明如下:

   (1). 如果有某个子句 $a_1 \wedge \cdots \wedge a_k \Rightarrow b$ 在 M 中为 false, 那么 $a_1 \wedge \cdots \wedge a_k$ 在 M 中为 true 且 b 在 M 中为 false

   (2). 但由于算法已经到达了不动点, 故此条知识应得以继续推理, 矛盾.

4. 因此 M 是 KB 的一个模型. 若 $KB \models q$, 则 q 在 KB 的所有模型中为 true, 也即在 M 下为 true.
5. 因此 q 在 inferred 表中也为 true, 进而能够被 FC 算法推断出来.