AI-hw5

张芷苒 PB21081601

> **7.13** This exercise looks into the relationship between clauses and implication sentences.
>
> a. Show that the clause $(\neg P_1 \vee \ldots \vee \neg P_m \vee Q)$ is logically equivalent to the implication sentence. $(P_1 \wedge \cdots \wedge P_m) \Rightarrow Q$
>
> b. Show that every clause (regardless of the number of positive literals) can be written in the form $(P_1 \wedge \cdots \wedge P_m) \Rightarrow (Q_q \vee \cdots \vee Q_n)$, 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).
>
> c. Write down the full resolution rule for sentences in implicative normal form.
>
> **Proof.** Prove the completeness of the forward chaining algorithm.

Part a: Show that the clause

$$(\neg P_1 \vee \ldots \vee \neg P_m \vee Q)$$

is equivalent to the implication

$$(P_1 \wedge \cdots \wedge P_m) \Rightarrow Q$$

.

The implication

$$(P_1 \wedge \cdots \wedge P_m) \Rightarrow Q$$

translates to

$$\neg(P_1 \wedge \cdots \wedge P_m) \vee Q$$

using the definition of implication. Applying De Morgan's Law, this becomes

$$(\neg P_1 \vee \ldots \vee \neg P_m) \vee Q$$

, which matches the given clause.

Part b: Show that any clause can be written as

$$(P_1 \wedge \cdots \wedge P_m) \Rightarrow (Q_1 \vee \cdots \vee Q_n)$$

.

A clause in propositional logic,

$$(L_1 \vee L_2 \vee \ldots \vee L_k)$$

, consists of literals that may be positive or negative. Rearranging negative literals on one side (antecedent) and positive on the other (consequent) and using the equivalence from part a, the clause is rewritten as

$$(P_1 \wedge \ldots \wedge P_m) \Rightarrow (Q_1 \vee \ldots \vee Q_n)$$

.

Part c: Write down the resolution rule for sentences in implicative normal form.

For implicative sentences:

$$(P_1 \wedge \ldots \wedge P_m) \Rightarrow (Q_1 \vee \ldots \vee Q_n)$$

and

$$(R_1 \wedge \ldots \wedge R_p) \Rightarrow (S_1 \vee \ldots \vee S_q)$$

if Q_i and $\neg S_j$ are contradictory, resolve to:

$$(P_1 \wedge \ldots \wedge P_m \wedge R_1 \wedge \ldots \wedge R_p)$$
$$\Rightarrow (Q_1 \vee \ldots \vee Q_{i-1} \vee Q_{i+1} \vee \ldots \vee Q_n \vee S_1 \vee \ldots \vee S_{j-1} \vee S_{j+1} \vee \ldots \vee S_q)$$

Proof of Completeness for Forward Chaining Algorithm:

Completeness implies that if a fact is logically entailed by the knowledge base, the forward chaining algorithm will infer it. The algorithm iteratively applies modus ponens to implications until no new facts are inferred. If Q is entailed, there is a series of rule applications that will lead to Q, and the algorithm will derive it given the exhaustive application of rules. Therefore, the algorithm is complete for knowledge bases in implicative normal form.