a. Show that the clause \( \neg P_1 \lor \ldots \lor \neg P_n, V Q \) is logically equivalent to the implication sentence \( (P_1 \land \ldots \land Q) \land (V Q) \).

b. Show that every clause (regardless of the number of positive literals) can be written in the form \( (P_1 \land \ldots \land P_n) \lor (Q_1 \land \ldots \land Q_m) \), 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.

### 7.14

According to some political pundits, a person who is radical \( (R) \) is electable \( (E) \) if he/she is conservative \( (C) \), but otherwise is not electable.

a. Which of the following are correct representations of this assertion?

   (i) \( (R \land E) \lor C \)
   
   (ii) \( R = (E \land C) \)
   
   (iii) \( R = ((C = E) \lor V) \)

b. Which of the sentences in (a) can be expressed in Ham form?

### 7.15

This question considers representing satisfiability (SAT) problems as CSPs.

a. Draw the constraint graph corresponding to the SAT problem

\[
\neg X_1 \land V \land X_2 \land (\neg X_2 \land V \land X_3) \land \ldots \land (\neg X_{n-1} \land V \land X_n)
\]

for the particular case \( n = 5 \).

b. How many solutions are there for this general SAT problem as a function of \( n \)?

c. Suppose we apply BACKTRACKING-SEARCH (page 215) to find all solutions to a SAT CSP of the type given in (a). (To find all solutions to a CSP, we simply modify the basic algorithm so it continues searching after each solution is found.) Assume that variables are ordered \( X_1, \ldots, X_n \) and false is ordered before true. How much time will the algorithm take to terminate? (Write an \( O(\cdot) \) expression as a function of \( n \).)

d. We know that SAT problems in Horn form can be solved in linear time by forward chaining (unit propagation). We also know that every tree-structured binary CSP with discrete, finite domains can be solved in time linear in the number of variables (Section 6.5). Are these two facts connected? Discuss.

### 7.16

Explain why every nonempty propositional clause, by itself, is satisfiable. Prove rigorously that every set of 3-SAT clauses is satisfiable, provided that each clause mentions exactly three distinct variables. What is the smallest set of such clauses that is unsatisfiable? Construct such a set.

### 7.17

A propositional 2-CNF expression is a conjunction of clauses, each containing exactly 2 literals, e.g.,

\[
(A \lor B) \land (\neg A \lor C) \land (\neg B \lor V) \land (\neg C \lor G) \land (\neg D \lor G').
\]
a. Prove using resolution that the above sentence entails \( G \).
Exercises 283

b. Two clauses are **semantically distinct** if they are not logically equivalent. How many semantically distinct 2-CNF clauses can be constructed from \( n \) proposition symbols?

c. Using your answer to (b), prove that propositional resolution always terminates in time polynomial in \( n \) given a 2-CNF sentence containing no more than \( n \) distinct symbols.

d. Explain why your argument in (c) does not apply to 3-CNF.

7.1R Consider the following sentence:

\[
[(\text{Food} = \text{Party}) \lor (\text{Drinks} \land \text{Party})] \lor [(\text{Food} \land \text{Drinks}) = \text{Party}].
\]

a. Determine, using enumeration, whether this sentence is valid, satisfiable (but not valid), or unsatisfiable.

b. Convert the left-hand and right-hand sides of the main implication into CNF, showing each step, and explain how the results confirm your answer to (a).

c. Prove your answer to (a) using resolution.

7.19 A sentence is in disjunctive normal form (DNF) if it is the disjunction of conjunctions of literals. For example, the sentence \((A \land B \land \neg C) \lor (\neg A \land C) \lor (B \land \neg C)\) is in DNF.

a. Any propositional logic sentence is logically equivalent to the assertion that some possible world in which it would be true is in fact the case. From this observation prove that any sentence can be written in DNF.

b. Construct an algorithm that converts any sentence in propositional logic into DNF. (Hint: The algorithm is similar to the algorithm for conversion to CNF given in Section 7.5.2)

c. Construct a simple algorithm that takes as input a sentence in DNF and returns a satisfying assignment if one exists, or reports that no satisfying assignment exists.

d. Apply the algorithms in (b) and (c) to the following set of sentences:

\[
A = B  \\
B = C  \\
C
\]

c. Since the algorithm in (b) is very similar to the algorithm for conversion to CNF, and since the algorithm in (c) is much simpler than any algorithm for solving a set of sentences in CNF, why is this technique not used in automated reasoning?

7.20 Convert the following set of sentences to clausal form.

\[
\begin{align*}
S1: A & \iff (B \lor E) \\
S2: E & \\
S3: C & \land F \\
S4: E & \iff B. \\
S5: B & \iff F. \\
S6: B & \iff C
\end{align*}
\]

Give a trace of the execution of DPLL on the conjunction of these clauses.