Figure 9.12, so we need not repeat it here. Instead, we give two example proofs. The first is the crime example from Section 9.3. The sentences in CNF are

\[
\begin{align*}
\neg \text{American}(x) & \lor \neg \text{Weapon}(y) \lor \neg \text{Sells}(x, y, z) \lor \neg \text{Hostile}(z) \lor \text{Criminal}(x) \\
\neg \text{Missile}(x) & \lor \text{Owns}(\text{Nono, V Sells( West, x, None)}) \\
\neg \text{Enemy}(\text{West, America}) & \lor \text{Hostile}(x) \\
\neg \text{Missile}(x) & \lor \text{Weapon}(x) \\
\text{Owns}(\text{Nono, Missile}) \\
\text{American}(\text{West}) & \\
\text{Enemy}(\text{Nono, America}) \\
\end{align*}
\]

We also include the negated goal \( \neg \text{Criminal}(\text{West}) \). The resolution proof is shown in Figure 9.11. Notice the structure: single "spine" beginning with the goal clause, resolving against clauses from the knowledge base until the empty clause is generated. This is characteristic of resolution on Horn clause knowledge bases. In fact, the clauses along the main spine correspond exactly to the consecutive values of the goals variable in the backward-chaining algorithm of Figure 9.6. This is because we always choose to resolve with a clause whose positive literal unified with the leftmost literal of the "current" clause on the spine; this is exactly what happens in backward chaining. Thus, backward chaining is just a special case of resolution with a particular control strategy to decide which resolution to perform next.

Our second example makes use of Skolemization and involves clauses that are not definite clauses. This results in a somewhat more complex proof structure. In English, the problem is as follows:

Everyone who loves all animals is loved by someone.
Anyone who kills an animal is loved by no one.
Jack loves all animals.
Either Jack or Curiosity killed the cat, who is named Tuna.
Did Curiosity kill the cat?
First, we express the original sentences, some background knowledge, and the negated goal \( G \) in first-order logic:

A. \( \forall y \forall x \text{Animal}(y) = \text{Loves}(x, y) \lor \text{Loves}(y, x) \)
B. \( \forall x \exists y \text{Animal}(x) \lor \text{Kills}(x, y) \lor \forall y \neg \text{Loves}(x, y) \)
C. \( \forall x \forall y \text{Animal}(x) \lor \text{Loves}(\text{Jack}, x) \)
D. \( \text{Kills}(\text{Jack}, \text{Tuna}) \lor \text{Kills}(\text{Curiosity}, \text{Tuna}) \)
E. \( \text{Cat}(\text{Tuna}) \)
F. \( \forall x \forall y \text{Cat}(x) \lor \text{Animal}(x) \lor \text{Loves}(\text{Jack}, x) \)
G. \( \neg \text{Kills}(\text{Curiosity}, \text{Tuna}) \)

Now we apply the conversion procedure to convert each sentence to CNF:

A1. \( \forall x \forall y \text{Animal}(y) \lor \text{Loves}(\text{F}(x), y) \)
A2. \( \forall y \forall x \neg \text{Loves}(\text{F}(x), y) \lor \text{Loves}(\text{G}(x), y) \)
B. \( \forall x \forall y \neg \text{Loves}(\text{F}(x), y) \lor \text{Animal}(x) \lor \text{Kills}(\text{Jack}, y) \lor \forall y \neg \text{Loves}(x, y) \)
C. \( \forall x \forall y \neg \text{Animal}(x) \lor \text{Loves}(\text{Jack}, y) \lor \forall y \neg \text{Loves}(x, y) \)
D. \( \text{Kills}(\text{Jack}, \text{Tuna}) \lor \text{Kills}(\text{Curiosity}, \text{Tuna}) \)
E. \( \text{Cat}(\text{Tuna}) \)
F. \( \forall x \forall y \neg \text{Cat}(x) \lor \text{Animal}(x) \lor \text{Loves}(\text{Curiosity}, x) \)
G. \( \neg \text{Kills}(\text{Curiosity}, \text{Tuna}) \)

The resolution proof that Curiosity killed the cat is given in Figure 9.12. In English, the proof could be paraphrased as follows:

Suppose Curiosity did not kill Tuna. We know that either Jack or Curiosity did; thus Jack must have. Now, Tuna is a cat and cats are animals, so Tuna is an animal. Because anyone who kills an animal is loved by no one, we know that no one loves Jack. On the other hand, Jack loves all animals, so someone loves him; so we have a contradiction. Therefore, Curiosity killed the cat.