Fundamentals of Artificial Intelligence

Laboratory

Dr. Mauro Dragoni

Department of Information Engineering and Computer Science

Academic Year 2020/2021

Exercise 9.1

- Consider the following axioms:

1. Mother(Lulu, Fifi)
2. Alive(Lulu)
3. ∀x ∀y Mother(x,y) ⇒ Parent(x,y)
4. ∀x ∀y (Parent(x,y) ∧ Alive(x)) ⇒ Older(x,y)
5. (Conclusion) Older(Lulu, Fifi)

Exercise 9.1

- Consider the following axioms:

1. **Mother(Lulu, Fifi)**

2. **Alive(Lulu)**

3. ∀x ∀y Mother(x,y) ⇒ Parent(x,y)
   ¬Mother(x,y) ∨ Parent(x,y)

4. ∀x ∀y (Parent(x,y) ∧ Alive(x)) ⇒ Older(x,y)
   ¬Parent(x,y) ∨ ¬Alive(x) ∨ Older(x,y)

5. (Conclusion) Older(Lulu, Fifi)
   ¬Older(Lulu, Fifi)

Exercise 9.1

- Consider the following axioms:

1. **Mother(Lulu, Fifi)**

2. **Alive(Lulu)**

3. ∀x ∀y Mother(x,y) ⇒ Parent(x,y)
   ¬Mother(x,y) ∨ Parent(x,y)

4. ∀x ∀y (Parent(x,y) ∧ Alive(x)) ⇒ Older(x,y)
   ¬Parent(x,y) ∨ ¬Alive(x) ∨ Older(x,y)

5. (Conclusion) Older(Lulu, Fifi)
   ¬Older(Lulu, Fifi)

6. [1, 3] Parent(Lulu,Fifi) {x/Lulu, y/Fifi}

Exercise 9.1

- Consider the following axioms:

1. **Mother(Lulu, Fifi)**

2. **Alive(Lulu)**

3. ∀x ∀y Mother(x,y) ⇒ Parent(x,y)
   ¬Mother(x,y) ∨ Parent(x,y)

4. ∀x ∀y (Parent(x,y) ∧ Alive(x)) ⇒ Older(x,y)
   ¬Parent(x,y) ∨ ¬Alive(x) ∨ Older(x,y)

5. (Conclusion) Older(Lulu, Fifi)
   ¬Older(Lulu, Fifi)

6. [1, 3] Parent(Lulu,Fifi) {x/Lulu, y/Fifi}

7. [4, 6] ¬Alive(Lulu) ∨ Older(Lulu, Fifi) {x/Lulu, y/Fifi}

Exercise 9.1

- Consider the following axioms:

1. **Mother(Lulu, Fifi)**

2. **Alive(Lulu)**

3. ∀x ∀y Mother(x,y) ⇒ Parent(x,y)
   ¬Mother(x,y) ∨ Parent(x,y)

4. ∀x ∀y (Parent(x,y) ∧ Alive(x)) ⇒ Older(x,y)
   ¬Parent(x,y) ∨ ¬Alive(x) ∨ Older(x,y)

5. (Conclusion) Older(Lulu, Fifi)
   ¬Older(Lulu, Fifi)

6. [1, 3] Parent(Lulu,Fifi) {x/Lulu, y/Fifi}

7. [4, 6] ¬Alive(Lulu) ∨ Older(Lulu, Fifi) {x/Lulu, y/Fifi}

8. [2, 7] Older(Lulu, Fifi)

Exercise 9.1

- Consider the following axioms:

1. **Mother(Lulu, Fifi)**

2. **Alive(Lulu)**

3. ∀x ∀y Mother(x,y) ⇒ Parent(x,y)
   ¬Mother(x,y) ∨ Parent(x,y)

4. ∀x ∀y (Parent(x,y) ∧ Alive(x)) ⇒ Older(x,y)
   ¬Parent(x,y) ∨ ¬Alive(x) ∨ Older(x,y)

5. (Conclusion) Older(Lulu, Fifi)
   ¬Older(Lulu, Fifi)

6. [1, 3] Parent(Lulu,Fifi) {x/Lulu, y/Fifi}

7. [4, 6] ¬Alive(Lulu) ∨ Older(Lulu, Fifi) {x/Lulu, y/Fifi}

8. [2, 7] Older(Lulu, Fifi)

9. [5, 8] □

Exercise 9.2

- Consider the following axioms:

    1. If something is intelligent, it has common sense
    2. Deep Blue does not have common sense
    3. (Conclusion) Deep Blue is not intelligent

Exercise 9.2

- Resolution

1. If something is intelligent, it has common sense
   $\forall x\ I(x) \Rightarrow H(x)$

2. Deep Blue does not have common sense
   $\neg H(D)$

3. (Conclusion) Deep Blue is not intelligent
   $\neg I(D)$

Exercise 9.2

- Resolution

1. If something is intelligent, it has common sense
   $\forall x\ I(x) \Rightarrow H(x)$
   $\neg I(x) \vee H(x)$

2. Deep Blue does not have common sense
   $\neg H(D)$

3. (Conclusion) Deep Blue is not intelligent
   $\neg I(D)$
   $I(D)$ (negated conclusion)

Exercise 9.2

- Resolution

1. If something is intelligent, it has common sense
   $\forall x\ I(x) \Rightarrow H(x)$
   $\mathbf{\neg I(x) \vee H(x)}$

2. Deep Blue does not have common sense
   $\mathbf{\neg H(D)}$

3. (Conclusion) Deep Blue is not intelligent
   $\neg I(D)$
   $\mathbf{I(D)}$ (negated conclusion)

4. [1, 2] $\neg I(D)$ $\{x/D\}$

Exercise 9.2

- Resolution

1. If something is intelligent, it has common sense
   $\forall x\ I(x) \Rightarrow H(x)$
   $\mathbf{\neg I(x) \vee H(x)}$

2. Deep Blue does not have common sense
   $\mathbf{\neg H(D)}$

3. (Conclusion) Deep Blue is not intelligent
   $\neg I(D)$
   $\mathbf{I(D)}$ (negated conclusion)

4. [1, 2] $\neg I(D)$ $\{x/D\}$

5. [3, 4] $\square$

Exercise 9.3

- Axioms:

  1. Allergies(x) → Sneeze(x)
  2. Cat(y) ∧ AllergicToCats(x) → Allergies(x)
  3. Cat(Felix)
  4. AllergicToCats(Mary)
  5. (Conclusion) Sneeze(Mary)

Exercise 9.3

- Axioms:

1. Allergies(x) → Sneeze(x)
   ¬Allergies(w) v Sneeze(w)

2. Cat(y) ∧ AllergicToCats(x) → Allergies(x)
   ¬Cat(y) v ¬AllergicToCats(z) ∨ Allergies(z)

3. **Cat(Felix)**

4. **AllergicToCats(Mary)**

5. (Conclusion) Sneeze(Mary)
   ¬Sneeze(Mary)

Exercise 9.3

- Axioms:

1. ¬Allergies(w) v Sneeze(w)
2. ¬Cat(y) v ¬AllergicToCats(z) v Allergies(z)
3. Cat(Felix)
4. AllergicToCats(Mary)
5. (Conclusion) ¬Sneeze(Mary)
6. [1, 2] ¬Cat(y) v Sneeze(z) v ¬AllergicToCats(z) {w/z}

Exercise 9.3

- Axioms:

1. ¬Allergies(w) v Sneeze(w)
2. ¬Cat(y) v ¬AllergicToCats(z) v Allergies(z)
3. Cat(Felix)
4. AllergicToCats(Mary)
5. (Conclusion) ¬Sneeze(Mary)
6. [1, 2] ¬Cat(y) v Sneeze(z) v ¬AllergicToCats(z) {w/z}
7. [3, 6] Sneeze(z) v ¬AllergicToCats(z) {y/Felix}

Exercise 9.3

- Axioms:

1. ¬Allergies(w) v Sneeze(w)
2. ¬Cat(y) v ¬AllergicToCats(z) v Allergies(z)
3. Cat(Felix)
4. AllergicToCats(Mary)
5. (Conclusion) ¬Sneeze(Mary)
6. [1, 2] ¬Cat(y) v Sneeze(z) v ¬AllergicToCats(z) {w/z}
7. [3, 6] Sneeze(z) v ¬AllergicToCats(z) {y/Felix}
8. [4, 7] Sneeze(Mary) {z/Mary}

Exercise 9.3

- Axioms:

1. ¬Allergies(w) v Sneeze(w)
2. ¬Cat(y) v ¬AllergicToCats(z) v Allergies(z)
3. Cat(Felix)
4. AllergicToCats(Mary)
5. (Conclusion) ¬Sneeze(Mary)
6. [1, 2] ¬Cat(y) v Sneeze(z) v ¬AllergicToCats(z) {w/z}
7. [3, 6] Sneeze(z) v ¬AllergicToCats(z) {y/Felix}
8. [4, 7] Sneeze(Mary) {z/Mary}
9. [5, 8] □

Exercise 9.4

- Consider the following axioms:

    1. All hounds howl at night.

    2. Anyone who has any cats will not have any mice.

    3. Light sleepers do not have anything which howls at night.

    4. John has either a cat or a hound.

    5. (Conclusion) If John is a light sleeper, then John does not have any mice.

Exercise 9.4

- The conclusion can be proved using Resolution as shown in the next slides.
- The first step is to write each axiom as a well-formed formula in first-order predicate calculus. The clauses written for the above axioms are shown below, using LS(x) for “light sleeper”.

1. All hounds howl at night.
   ∀x (HOUND(x) → HOWL(x))

2. Anyone who has any cats will not have any mice.
   ∀x ∀y (HAVE(x,y) ∧ CAT(y) → ¬∃z (HAVE(x,z) ∧ MOUSE(z)))

3. Light sleepers do not have anything which howls at night.
   ∀x (LS(x) → ¬∃y (HAVE(x,y) ∧ HOWL(y)))

4. John has either a cat or a hound.
   ∃x (HAVE(John,x) ∧ (CAT(x) ∨ HOUND(x)))

5. (Conclusion) If John is a light sleeper, then John does not have any mice.
   LS(John) → ¬∃z (HAVE(John,z) ∧ MOUSE(z))

Exercise 9.4

- The next step is to transform each well-formed formula into Prenex Normal Form, skolemize, and rewrite as clauses in conjunctive normal form.

1. All hounds howl at night.

   $\forall x\ (\text{HOUND}(x) \rightarrow \text{HOWL}(x))$

   $\mathbf{\neg HOUND(x) \vee HOWL(x)}$

Exercise 9.4

- The next step is to transform each wff into Prenex Normal Form, skolemize, and rewrite as clauses in conjunctive normal form; these transformations are shown below.

2. Anyone who has any cats will not have any mice.

$\forall x \forall y (HAVE(x,y) \land CAT(y) \rightarrow \neg \exists z (HAVE(x,z) \land MOUSE(z)))$

$\forall x \forall y (HAVE (x,y) \land CAT (y) \rightarrow \forall z \neg(HAVE(x,z) \land MOUSE (z)))$

$\forall x \forall y \forall z (\neg(HAVE (x,y) \land CAT (y)) \lor \neg(HAVE(x,z) \land MOUSE (z)))$

$\neg HAVE(x,y) \lor \neg CAT(y) \lor \neg HAVE(x,z) \lor \neg MOUSE(z)$

Exercise 9.4

- The next step is to transform each wff into Prenex Normal Form, skolemize, and rewrite as clauses in conjunctive normal form; these transformations are shown below.

3. Light sleepers do not have anything which howls at night.

$\forall x\ (LS(x) \rightarrow \neg \exists y\ (HAVE(x,y) \wedge HOWL(y)))$

$\forall x\ (LS(x) \rightarrow \forall\ y\ \neg(HAVE\ (x,y) \wedge HOWL(y)))$

$\forall x\ \forall y\ (LS(x) \rightarrow \neg HAVE(x,y) \vee \neg HOWL(y))$

$\forall x\ \forall y\ (\neg LS(x) \vee \neg HAVE(x,y) \vee \neg HOWL(y))$

$\mathbf{\neg LS(x) \vee \neg HAVE(x,y) \vee \neg HOWL(y)}$

Exercise 9.4

- The next step is to transform each wff into Prenex Normal Form, skolemize, and rewrite as clauses in conjunctive normal form; these transformations are shown below.

4. John has either a cat or a hound.

   $\exists x\ (HAVE(John,x) \wedge (CAT(x) \vee HOUND(x)))$

   $HAVE(John,a) \wedge (CAT(a) \vee HOUND(a))$

Exercise 9.4

- The next step is to transform each wff into Prenex Normal Form, skolemize, and rewrite as clauses in conjunctive normal form; these transformations are shown below.

5. (Conclusion) If John is a light sleeper, then John does not have any mice.

   LS(John) → ¬∃z (HAVE(John,z) ∧ MOUSE(z))

   ¬[LS(John) → ¬∃z (HAVE(John,z) ∧ MOUSE(z))] (negated conclusion)

   ¬[¬LS(John) ∨ ¬∃z (HAVE(John, z) ∧ MOUSE(z))]

   LS(John) ∧ ∃z (HAVE(John, z) ∧ MOUSE(z)))

   LS(John) ∧ HAVE(John,b) ∧ MOUSE(b)

Exercise 9.4

- The set of CNF clauses for this problem is thus as follows:

    1. ¬HOUND(x) ∨ HOWL(x)
    2. ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,z) ∨ ¬MOUSE(z)
    3. ¬LS(x) ∨ ¬HAVE(x,y) ∨ ¬HOWL(y)
    4.
        a. HAVE(John,a)
        b. CAT(a) ∨ HOUND(a)
    5.
        a. LS(John)
        b. HAVE(John,b)
        c. MOUSE(b)

Exercise 9.4

- Now we proceed to prove the conclusion by resolution using the above clauses. Each result clause is numbered; the numbers of its parent clauses are shown to its left.

1. ¬HOUND(x) ∨ HOWL(x)
2. ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,z) ∨ ¬MOUSE(z)
3. ¬LS(x) ∨ ¬HAVE(x,y) ∨ ¬HOWL(y)
4.
    a. HAVE(John,a)
    b. CAT(a) ∨ HOUND(a)
5.
    a. LS(John)
    b. HAVE(John,b)
    c. MOUSE(b)

6. [1, 4(b)] CAT(a) ∨ HOWL(a)

Exercise 9.4

- Now we proceed to prove the conclusion by resolution using the above clauses. Each result clause is numbered; the numbers of its parent clauses are shown to its left.

1. ¬HOUND(x) ∨ HOWL(x)
2. ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,z) ∨ ¬MOUSE(z)
3. ¬LS(x) ∨ ¬HAVE(x,y) ∨ ¬HOWL(y)
4.
    a. HAVE(John,a)
    b. CAT(a) ∨ HOUND(a)
5.
    a. LS(John)
    b. HAVE(John,b)
    c. MOUSE(b)

6. [1, 4(b)] CAT(a) ∨ HOWL(a)
7. [2, 5(c)] ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,b)

Exercise 9.4

- Now we proceed to prove the conclusion by resolution using the above clauses. Each result clause is numbered; the numbers of its parent clauses are shown to its left.

1. ¬HOUND(x) ∨ HOWL(x)
2. ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,z) ∨ ¬MOUSE(z)
3. ¬LS(x) ∨ ¬HAVE(x,y) ∨ ¬HOWL(y)
4.
    a. HAVE(John,a)
    b. CAT(a) ∨ HOUND(a)
5.
    a. LS(John)
    b. HAVE(John,b)
    c. MOUSE(b)

6. [1, 4(b)] CAT(a) ∨ HOWL(a)
7. [2, 5(c)] ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,b)
8. [7, 5(b)] ¬HAVE(John,y) ∨ ¬CAT(y)

Exercise 9.4

- Now we proceed to prove the conclusion by resolution using the above clauses. Each result clause is numbered; the numbers of its parent clauses are shown to its left.

1. ¬HOUND(x) ∨ HOWL(x)
2. ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,z) ∨ ¬MOUSE(z)
3. ¬LS(x) ∨ ¬HAVE(x,y) ∨ ¬HOWL(y)
4.
    a. HAVE(John,a)
    b. CAT(a) ∨ HOUND(a)
5.
    a. LS(John)
    b. HAVE(John,b)
    c. MOUSE(b)

6. [1, 4(b)] CAT(a) ∨ HOWL(a)
7. [2, 5(c)] ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,b)
8. [7, 5(b)] ¬HAVE(John,y) ∨ ¬CAT(y)
9. [6, 8] ¬HAVE(John,a) ∨ HOWL(a)

Exercise 9.4

- Now we proceed to prove the conclusion by resolution using the above clauses. Each result clause is numbered; the numbers of its parent clauses are shown to its left.

1. ¬HOUND(x) ∨ HOWL(x)
2. ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,z) ∨ ¬MOUSE(z)
3. ¬LS(x) ∨ ¬HAVE(x,y) ∨ ¬HOWL(y)
4.
    a. HAVE(John,a)
    b. CAT(a) ∨ HOUND(a)
5.
    a. LS(John)
    b. HAVE(John,b)
    c. MOUSE(b)

6. [1, 4(b)] CAT(a) ∨ HOWL(a)
7. [2, 5(c)] ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,b)
8. [7, 5(b)] ¬HAVE(John,y) ∨ ¬CAT(y)
9. [6, 8] ¬HAVE(John,a) ∨ HOWL(a)
10. [4(a), 9] HOWL(a)

Exercise 9.4

- Now we proceed to prove the conclusion by resolution using the above clauses. Each result clause is numbered; the numbers of its parent clauses are shown to its left.

1. ¬HOUND(x) ∨ HOWL(x)
2. ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,z) ∨ ¬MOUSE(z)
3. ¬LS(x) ∨ ¬HAVE(x,y) ∨ ¬HOWL(y)
4.
    a. HAVE(John,a)
    b. CAT(a) ∨ HOUND(a)
5.
    a. LS(John)
    b. HAVE(John,b)
    c. MOUSE(b)

6. [1, 4(b)] CAT(a) ∨ HOWL(a)
7. [2, 5(c)] ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,b)
8. [7, 5(b)] ¬HAVE(John,y) ∨ ¬CAT(y)
9. [6, 8] ¬HAVE(John,a) ∨ HOWL(a)
10. [4(a), 9] HOWL(a)
11. [3, 10] ¬LS(x) ∨ ¬HAVE(x,a)

Exercise 9.4

- Now we proceed to prove the conclusion by resolution using the above clauses. Each result clause is numbered; the numbers of its parent clauses are shown to its left.

1. ¬HOUND(x) ∨ HOWL(x)
2. ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,z) ∨ ¬MOUSE(z)
3. ¬LS(x) ∨ ¬HAVE(x,y) ∨ ¬HOWL(y)
4.
    a. HAVE(John,a)
    b. CAT(a) ∨ HOUND(a)
5.
    a. LS(John)
    b. HAVE(John,b)
    c. MOUSE(b)

6. [1, 4(b)] CAT(a) ∨ HOWL(a)
7. [2, 5(c)] ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,b)
8. [7, 5(b)] ¬HAVE(John,y) ∨ ¬CAT(y)
9. [6, 8] ¬HAVE(John,a) ∨ HOWL(a)
10. [4(a), 9] HOWL(a)
11. [3, 10] ¬LS(x) ∨ ¬HAVE(x,a)
12. [4(a), 11] ¬LS(John)

Exercise 9.4

- Now we proceed to prove the conclusion by resolution using the above clauses. Each result clause is numbered; the numbers of its parent clauses are shown to its left.

1. ¬HOUND(x) ∨ HOWL(x)
2. ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,z) ∨ ¬MOUSE(z)
3. ¬LS(x) ∨ ¬HAVE(x,y) ∨ ¬HOWL(y)
4.
    a. HAVE(John,a)
    b. CAT(a) ∨ HOUND(a)
5.
    a. LS(John)
    b. HAVE(John,b)
    c. MOUSE(b)

6. [1, 4(b)] CAT(a) ∨ HOWL(a)
7. [2, 5(c)] ¬HAVE(x,y) ∨ ¬CAT(y) ∨ ¬HAVE(x,b)
8. [7, 5(b)] ¬HAVE(John,y) ∨ ¬CAT(y)
9. [6, 8] ¬HAVE(John,a) ∨ HOWL(a)
10. [4(a), 9] HOWL(a)
11. [3, 10] ¬LS(x) ∨ ¬HAVE(x,a)
12. [4(a), 11] ¬LS(John)
13. [5(a), 12] □

Exercise 9.5

- Convert to First order Logic:

1. Marcus was a man.
2. Marcus was a Roman.
3. All men are people.
4. Caesar was a ruler.
5. All Romans were either loyal to Caesar or hated him (or both).
6. Everyone is loyal to someone.
7. People only try to assassinate rulers they are not loyal to.
8. Marcus tried to assassinate Caesar.

Exercise 9.5

- Convert to First order Logic:

1. Marcus was a man.
   Man(Marcus)

2. Marcus was a Roman.
   Roman(Marcus)

3. All men are people.
   ∀x Man(x) → Person(x)

4. Caesar was a ruler.
   Ruler(Caesar)

5. All Romans were either loyal to Caesar or hated him (or both).
   ∀x Roman(x) → Loyal(x, Caesar) v Hate(x, Caesar)

6. Everyone is loyal to someone.
   ∀x∃y Loyal(x, y)

7. People only try to assassinate rulers they are not loyal to.
   ∀x∀y Person(x) ∧ Ruler(y) ∧ Tryassasin(x, y) → ¬Loyal(x, y)

8. Marcus tried to assassinate Caesar.
   Tryassasin(Marcus, Caesar)

Exercise 9.5

- Convert to Clausal Form

1. Marcus was a man.
   Man(Marcus)

2. Marcus was a Roman.
   Roman(Marcus)

3. All men are people.
   ¬Man(x) v Person(x)

4. Caesar was a ruler.
   Ruler(Caesar)

5. All Romans were either loyal to Caesar or hated him (or both).
   ¬Roman(x) v Loyal(x, Caesar) v Hate(x, Caesar)

6. Everyone is loyal to someone.
   Loyal(x, y)

7. People only try to assassinate rulers they are not loyal to.
   ¬Person(x) v ¬Ruler(y) v ¬Tryassasin(x, y) v ¬Loyal(x, y)

8. Marcus tried to assassinate Caesar.
   Tryassasin(Marcus, Caesar)

Exercise 9.5

1. **Man(Marcus)**
2. **Roman(Marcus)**
3. **¬Man(x) v Person(x)**
4. **Ruler(Caesar)**
5. **¬Roman(x) v Loyal(x, Caesar) v Hate(x, Caesar)**
6. **Loyal(x, y)**
7. **¬Person(x) v ¬Ruler(y) v ¬Tryassasin(x, y) v ¬Loyal(x, y)**
8. **Tryassasin(Marcus, Caesar)**

9. Who hate Caesar

¬Hate(x, Caesar) v Ans(x)

Exercise 9.5

1. Man(Marcus)
2. Roman(Marcus)
3. ¬Man(x) v Person(x)
4. Ruler(Caesar)
5. ¬Roman(x) v Loyal(x, Caesar) v Hate(x, Caesar)
6. Loyal(x, y)
7. ¬Person(x) v ¬Ruler(y) v ¬Tryassasin(x, y) v ¬Loyal(x, y)
8. Tryassasin(Marcus, Caesar)
9. ¬Hate(x, Caesar) v Ans(x)
10. R[9, 5c] ¬Roman(x) v Loyal(x, Caesar) v Ans(x)

Exercise 9.5

1. Man(Marcus)
2. Roman(Marcus)
3. ¬Man(x) v Person(x)
4. Ruler(Caesar)
5. ¬Roman(x) v Loyal(x, Caesar) v Hate(x, Caesar)
6. Loyal(x, y)
7. ¬Person(x) v ¬Ruler(y) v ¬Tryassasin(x, y) v ¬Loyal(x, y)
8. Tryassasin(Marcus, Caesar)
9. ¬Hate(x, Caesar) v Ans(x)

10.	R[9, 5c]	¬Roman(x) v Loyal(x, Caesar) v Ans(x)	
11.	R[10a, 2]	Loyal(Markus, Caesar) v Ans(Markus)	{x/Marcus}

Exercise 9.5

1. **Man(Marcus)**
2. **Roman(Marcus)**
3. **¬Man(x) v Person(x)**
4. **Ruler(Caesar)**
5. **¬Roman(x) v Loyal(x, Caesar) v Hate(x, Caesar)**
6. **Loyal(x, y)**
7. **¬Person(x) v ¬Ruler(y) v ¬Tryassasin(x, y) v ¬Loyal(x, y)**
8. **Tryassasin(Marcus, Caesar)**
9. **¬Hate(x, Caesar) v Ans(x)**

10. R[9, 5c] ¬Roman(x) v Loyal(x, Caesar) v Ans(x)

11. R[10a, 2] Loyal(Markus, Caesar) v Ans(Markus) {x/Marcus}

12. R[11, 7] ¬Person(Markus) v ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus) {x/Markus, y/Caesar}

Exercise 9.5

1. Man(Marcus)
2. Roman(Marcus)
3. ¬Man(x) v Person(x)
4. Ruler(Caesar)
5. ¬Roman(x) v Loyal(x, Caesar) v Hate(x, Caesar)
6. Loyal(x, y)
7. ¬Person(x) v ¬Ruler(y) v ¬Tryassasin(x, y) v ¬Loyal(x, y)
8. Tryassasin(Marcus, Caesar)
9. ¬Hate(x, Caesar) v Ans(x)

10. R[9, 5] ¬Roman(x) v Loyal(x, Caesar) v Ans(x)

11. R[10, 2] Loyal(Markus, Caesar) v Ans(Markus) {x/Marcus}

12. R[11, 7] ¬Person(Markus) v ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus) {x/Markus, y/Caesar}

13. R[12, 3] ¬Man(Markus) v ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus) {x/Markus}

Exercise 9.5

1. Man(Marcus)
2. Roman(Marcus)
3. ¬Man(x) v Person(x)
4. Ruler(Caesar)
5. ¬Roman(x) v Loyal(x, Caesar) v Hate(x, Caesar)
6. Loyal(x, y)
7. ¬Person(x) v ¬Ruler(y) v ¬Tryassasin(x, y) v ¬Loyal(x, y)
8. Tryassasin(Marcus, Caesar)
9. ¬Hate(x, Caesar) v Ans(x)

10. R[9, 5] ¬Roman(x) v Loyal(x, Caesar) v Ans(x)

11. R[10, 2] Loyal(Markus, Caesar) v Ans(Markus) {x/Marcus}

12. R[11, 7] ¬Person(Markus) v ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus)
{x/Markus, y/Caesar}

13. R[12, 3] ¬Man(Markus) v ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus)
{x/Markus}

14. R[13, 1] ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus)

Exercise 9.5

1. Man(Marcus)
2. Roman(Marcus)
3. ¬Man(x) v Person(x)
4. Ruler(Caesar)
5. ¬Roman(x) v Loyal(x, Caesar) v Hate(x, Caesar)
6. Loyal(x, y)
7. ¬Person(x) v ¬Ruler(y) v ¬Tryassasin(x, y) v ¬Loyal(x, y)
8. Tryassasin(Marcus, Caesar)
9. ¬Hate(x, Caesar) v Ans(x)

10. R[9, 5] ¬Roman(x) v Loyal(x, Caesar) v Ans(x)

11. R[10, 2] Loyal(Markus, Caesar) v Ans(Markus) {x/Marcus}

12. R[11, 7] ¬Person(Markus) v ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus)
{x/Markus, y/Caesar}

13. R[12, 3] ¬Man(Markus) v ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus)
{x/Markus}

14. R[13, 1] ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus)

15. R[14, 4] ¬Tryassasin(Markus, Caesar) v Ans(Markus)

Exercise 9.5

1. Man(Marcus)
2. Roman(Marcus)
3. ¬Man(x) v Person(x)
4. Ruler(Caesar)
5. ¬Roman(x) v Loyal(x, Caesar) v Hate(x, Caesar)
6. Loyal(x, y)
7. ¬Person(x) v ¬Ruler(y) v ¬Tryassasin(x, y) v ¬Loyal(x, y)
8. Tryassasin(Marcus, Caesar)
9. ¬Hate(x, Caesar) v Ans(x)

10. R[9, 5] ¬Roman(x) v Loyal(x, Caesar) v Ans(x)

11. R[10, 2] Loyal(Markus, Caesar) v Ans(Markus) {x/Marcus}

12. R[11, 7] ¬Person(Markus) v ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus)
{x/Markus, y/Caesar}

13. R[12, 3] ¬Man(Markus) v ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus)
{x/Markus}

14. R[13, 1] ¬Ruler(Caesar) v ¬Tryassasin(Markus, Caesar) v Ans(Markus)

15. R[14, 4] ¬Tryassasin(Markus, Caesar) v Ans(Markus)

16. **R[15a, 8] Ans(Markus)**

Exercise 9.6

- **Problem Statement**
  Tony and Ellen belong to the Hoofers Club. Every member of the Hoofers Club is either a skier or a mountain climber or both. No mountain climber likes rain, and all skiers like snow. Ellen dislikes whatever Tony likes and likes whatever Tony dislikes. Tony likes rain and snow.

- **Query**
  Is there a member of the Hoofers Club who is a mountain climber but not a skier?