EXERCISE 1

EXERCISE 1.

CONSIDERING THE FOLLOWING SENTENCES:

1. JULIET LOVE ROMEO, HER FATHER AND HER MOTHER.
2. ROMEO LOVES JULIET AND ALSO EVERYBODY LOVED BY JULIET.
3. THE FATHER OF ROMEO LOVES ROMEO, BUT DOES NOT LOVE THE FATHER OF JULIET.
4. THE FATHER OF ROMEO LOVES EVERYBODY LOVED BY ROMEO.


(ii) INDICATE THE CLOSED ATOMIC SENTENCES CONTAINED IN 1-4, i.e. SENTENCES WHICH DO NOT CONTAIN QUANTIFIERS OR CONNECTIVES.

(iii) FORMALIZE 1-4.

(iv) DO YOU THINK THAT 1-4 ENTAIL A CONTRADDICTION? IF SO, CAN YOU INDICATE A CLOSED ATOMIC SENTENCE SENTENCE SUCH THAT BOTH THIS SENTENCE AND ITS NEGATION FOLLOW FROM 1-4?

(i) LEXICON

Names \{ Romeo, Juliet \}
Predicate \{ _Love_ \}
Functions \{ father of _, mother of _ \}

FORMAL LANGUAGE

\[ L = ( \{ R, J \} ; \{ L ( _ , _) \} ; \{ f(_), m(_) \} ) \]

where Romeo = R, Juliet = J; Love = L; father of = f, mother of = m

(ii) – (iii) CLOSED ATOMIC FORMULAS and FORMALIZATION

1. JULIET LOVES ROMEO, HER FATHER AND HER MOTHER.

Closed atomic formulas:

\[ Juliet \ loves \ Romeo = L(J, R) \]
\[ Juliet \ loves \ her \ father = L(J, f(J)) \]
\[ Juliet \ loves \ her \ mother = L(J, m(J)) \]

Formalization:

\[ L(J, R) \land L(J, f(J)) \land L(J, m(J)) \]
2. ROMEO LOVES JULIET AND ALSO EVERYBODY LOVED BY JULIET.

Closed atomic formulas:

Romeo loves Juliet = \( L(R,J) \)

Formalization:

\[ L(R, J) \land (\forall x. L(J, x) \rightarrow L(R, x)) \]

3. THE FATHER OF ROMEO LOVES ROMEO BUT DOES NOT LOVE THE FATHER OF JULIET.

Closed atomic formulas:

The father of Romeo loves Romeo = \( L(f(R), R) \)
The father of Romeo loves the father of Juliet = \( L(f(R), f(J)) \)

Formalization:

\[ L(f(R), R) \land \neg L(f(R), f(J)) \]

4. THE FATHER OF ROMEO LOVES EVERYBODY LOVED BY ROMEO.

Closed atomic formulas: none.

Formalization:

\[ (\forall x. L(R, x) \rightarrow L(f(R), x)) \]

(iii) CONTRADICTION

The contradiction is in the fourth sentence; the right sequence is:

1. Juliet loves Romeo, her father and her mother.
2. Romeo loves Juliet and also everybody loved by Romeo.
3. The father of Romeo loves Romeo.
4. The father of Romeo loves everybody loved by Romeo, but does not love the father of Juliet.

The fourth sentence expresses a universal quantification with an exception; it can be formalized as follows:

\[ \forall x. (L(R, x) \land \neg x = f(J)) \rightarrow L(f(R), x) \]
EXERCISE 2.

FORMALIZE THE FOLLOWING SENTENCES BY ABRAHAM LINCOLN:

“You can fool all the people some of the time, and some of the people all the time, but you cannot fool all the people all the time.”

There is a problem here: here the pronoun “you” stands for a generic name, i.e., an expression like “John Smith” in English or “Mario Rossi” in Italian. A generic name is a device for expressing generality without using a universal quantification. Therefore the best formalization is one that retains a generic name – unlike the suggestion in the paper.

Lexicon:
Predicates: \{ _ is a person = P(_), _ is a period ot time = T(_) (unary predicates); _ fools _ at time_ = F ( _, _, _ ) (ternary predicates) \}.
Names: = \{a \}.

Formalization:
\[
\begin{align*}
\exists t. T(x) \land (\forall y. P(y) \to (F(a, y, t) \land (\exists y. P(y) \land (\forall t. T(t) \to (F(a, y, t)) \land \neg (\forall y. P(y) \to (\forall t. T(t) \to F(a, y, t)) \land \neg (\forall y. P(y) \to (\forall t. T(t) \to F(a, y, t)))).
\end{align*}
\]

EXERCISE 3.

FORMALIZE THE FOLLOWING SENTENCES IN PROPOSITIONAL LOGIC. INDICATE ALWAYS THE KEY, E.G. J = “JOHN COMES”.

(i) JOHN IS NOT ONLY STUPID BUT NASTY TOO.

\[
A := S \land N;
\]

Formalization:

(ii) IF YOU STAY WITH ME IF I WON’T DRINK ANYMORE, THEN I WILL NOT DRINK ANYMORE.

\[
B := (\neg D \to S) \to \neg D
\]

Formalization:
(iii)    CHARLES COMES IF ELSA AND THE OTHER WAY AROUND.

\[ D = \text{Charles comes if Elsa and the other way around} \]
\[ \text{Charles comes} = C \]
\[ \text{Elsa comes} = E \]

Formalization:
\[ D := ( E \rightarrow C ) \land ( E \rightarrow C ); \]

(iv)    JOHN IS GOING TO SCHOOL AND IF IT IS RAINING SO IS PETER

\[ C = \text{John is going to school and if it is raining so is Peter} \]
\[ J = \text{John is going to school} \]
\[ R = \text{it is raining} \]
\[ P = \text{Peter is going to school} \]

Formalization:
\[ C := J \land ( R \rightarrow P ). \]