Example:
Representing Facts in First-Order Logic

1. Lucy* is a professor
2. All professors are people.
3. John is the dean.
4. Deans are professors.
5. All professors consider the dean a friend or don't know him.
6. Everyone is a friend of someone.
7. People only criticize people that are not their friends.
8. Lucy criticized John .

* Name changed for privacy reasons.

Same example, more formally

Knowledge base:

- is-prof(lucy)
- $\forall$ x (is-prof(x) → is-person(x))
- is-dean(John)
- $\forall$ x (is-dean(x) → is-prof(x))
- $\forall$ x ($\forall$ y (is-prof(x) $\wedge$ is-dean(y) → is-friend-of(y,x) $\vee$ $\neg$knows(x, y)))
- $\forall$ x ($\exists$ y (is-friend-of (y, x)))
- $\forall$ x ($\forall$ y (is-person(x) $\wedge$ is-person(y) $\wedge$ criticize (x,y) → $\neg$is-friend-of (y,x)))
- criticize(lucy, John)

Question: Is John no friend of Lucy?

$\neg$is-friend-of(John ,lucy)

How the machine “sees” it:

Knowledge base:

- P1(A)
- $\forall$ x (P1(x) $\rightarrow$ P3(x))
- P4(B)
- $\forall$ x (P4(x) $\rightarrow$ P1(x))
- $\forall$ x ($\forall$ y (P1(x) $\wedge$ P4(y) $\rightarrow$ P2(y,x) $\vee$ $\neg$P5(x, y)))
- $\forall$ x ($\exists$ y (P2(y, x)))
- $\forall$ x ($\forall$ y (P3 (x) $\wedge$ P3(y) $\wedge$ P6(x,y) $\rightarrow$ $\neg$P2(y,x)))
- P6(A, B)

Question: $\neg$P2(B ,A)?

Knowledge Engineering

1. Identify the task.
2. Assemble the relevant knowledge.
3. Decide on a vocabulary of predicates, functions, and constants.
4. Encode general knowledge about the domain.
5. Encode a description of the specific problem instance.
6. Pose queries to the inference procedure and get answers.
7. Debug the knowledge base.

Knowledge Engineering

1. All professors are people.
2. Deans are professors.
3. All professors consider the dean a friend or don't know him.
4. Everyone is a friend of someone.
5. People only criticize people that are not their friends.
6. Lucy* is a professor
7. John is the dean.
8. Lucy criticized John.
9. Is John a friend of Lucy's?