Topic 2.5 Natural language to FOL Conversion, and application of the Resolution rule

a) Resolution requires, the KB is in CNF, i.e. each sentence is a clause or disjunction of FOL literals, and the clauses are in conjunction to each other.

We show the conversion in an example, and then apply Resolution to answer a query.

b) Example of conversion

i) KB in Natural Language:

1. Every guest receives at least one gift.
2. Karim is a guest.

ii) KB in FOL :

1. $\forall x$ (Guest(x) $\Rightarrow$ $\exists y$ (Gift(y) $\wedge$ Receives(x, y)))
2. Guest(Karim)

1. $\forall x\ (\ Guest(x) \Rightarrow \exists y\ (Gift(y) \wedge Receives(x, y)))$
2. $Guest(Karim)$

iii) Conversion to CNF:

I. Standardize variables. [Use separate variables for different quantifiers.]

II. Eliminate $\Rightarrow$ and $\Leftrightarrow$, and move $\neg$ inward.

$\forall x\ (\neg Guest(x) \vee (\exists y(Gift(y) \wedge Receives(x, y))))$

III. Skolemize

$\forall x\ (\neg Guest(x) \vee (Gift(GiftFor(x)) \wedge Receives(x, GiftFor(x))))$

IV. Drop all $\forall$s.

$(\neg Guest(x) \vee (Gift(GiftFor(x)) \wedge Receives(x, GiftFor(x))))$

V. Simplify further to get CNF.

$(\neg Guest(x) \vee Gift(GiftFor(x)))\ \wedge$
$(\neg Guest(x) \vee Receives(x, GiftFor(x)))$

iv) KB in CNF of FOL:

1. $\neg Guest(x) \vee Gift(GiftFor(x))$
2. $\neg Guest(x) \vee Receives(x, GiftFor(x))$
3. $Guest(Karim)$

iv) KB in CNF of FOL:

1. $\neg$Guest(x) $\vee$ Gift(GiftFor(x))
2. $\neg$Guest(x) $\vee$ Receives(x, GiftFor(x))
3. Guest(Karim)

d) Query: Does Karim receive a gift?

$\exists$x (Gift(x) $\wedge$ Receives(Karim, x))

i) To answer / prove, add the negation of the sentence to the KB and try to derive a contradiction.

ii) Equivalently,
Add the CNF of the negation of the query to the KB, and try to derive an empty clause ([]) by applying the Resolution rule repeatedly.

iii) If [] can't be derived, it means that the sentence (query) is False.
This is known as the <u>Resolution-Refutation completeness</u> of KB.

e) Conversion of the negation of the query:

i) $\neg(\exists x\ (Gift(x) \wedge Receives(Karim, x)))$

ii) $\forall x\ \neg(Gift(x) \wedge Receives(Karim, x))$

iii) $\forall x\ (\neg Gift(x) \vee \neg Receives(Karim, x))$

iv) $\neg Gift(x) \vee \neg Receives(Karim, x)$

f) The KB after adding the negation of the query:

1. $\neg Guest(x) \vee Gift(GiftFor(x))$
2. $\neg Guest(x) \vee Receives(x, GiftFor(x))$
3. $Guest(Karim)$

T1. $\neg Gift(y) \vee \neg Receives(Karim, y)$

1. $\neg$Guest(x) $\vee$ Gift(GiftFor(x))
2. $\neg$Guest(x) $\vee$ Receives(x, GiftFor(x))
3. Guest(Karim)

T1. $\neg$Gift(y) $\vee$ $\neg$Receives(Karim, y)

g) Finding the answer to the query :

I. Resolving T1 and 1 with θ = {y / GiftFor(x)}:
T2. $\neg$Guest(x) $\vee$ $\neg$Receives(Karim, GiftFor(x))

II. Resolving T1 and 2 with θ = {x / Karim, y / GiftFor(Karim)}:
T3. $\neg$Guest(Karim) $\vee$ $\neg$Gift(GiftFor(Karim))

III. Resolving T2 and 2 with θ = {x / Karim}:
T4. $\neg$Guest(Karim)

IV. Resolving T2 and 3 with θ = {x / Karim}:
T5. $\neg$Receives(Karim, GiftFor(Karim))

~~V. Resolving T3 and 1 with θ = {x / Karim}:~~
~~T6. ¬Guest(Karim)~~

VI. Resolving T3 and 3 with θ = {}:
T7. $\neg$Gift(GiftFor(Karim))

VII. Resolving T4 and 3 with θ = {}:
[]

As an **empty clause** is resolved, the sentence (query) is proved true, that is, the answer is ‘Yes, Karim receives a gift.’.