Example: Representing Facts in FOL

1. Lucy is a professor
2. All professors are people.
3. Fuchs 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 Fuchs.

Example: knowledge base

- is-prof(lucy)
- $\forall x$ (is-prof(x) $\Rightarrow$ is-person(x))
- is-dean(fuchs)
- $\forall x$ (is-dean(x) $\Rightarrow$ is-prof(x))
- $\forall x$ ($\forall y$ (is-prof(x) $\wedge$ is-dean(y) $\Rightarrow$ 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) $\Rightarrow$ is-friend-of (y,x)))
- criticize(lucy,fuchs)

Question: Is Fuchs is friend of Lucy?

is-friend-of(fuchs,lucy)

Resolution Rule of Inference

General Rule:

Assume:	$E \vee E_{12} \vee ... \vee E_{1k}$
and	$\neg E \vee E_{22} \vee ... \vee E_{2l}$
Then:	$E_{12} \vee ... \vee E_{1k} \vee E_{22} \vee ... \vee E_{2l}$

Note: E_{ij} can be negated.

Example:

Assume:	$E_1 \vee E_2$	playing tennis or raining
and	$\neg E_2 \vee E_3$	not raining or working
Then:	$E_1 \vee E_3$	playing tennis or working

Resolution Example: FOL

Example: Prove *bird(tweety)*

Axioms:	Regular	CNF	
1:	$\forall x : feathers(x) \rightarrow bird(x)$	$\neg feathers(x) \vee bird(x)$	NB
2:	$feathers(tweety)$	$feathers(tweety)$	
3:	$\neg bird(tweety)$	$\neg bird(tweety)$	Add negation
4:		$\neg feathers(tweety)$	

Resolution Proof

1. Resolve 3 and 1, specializing (i.e. "unifying") tweety for x. Add :feathers(tweety)
2. Resolve 4 and 2. Add NIL: conclude ***bird(tweety)***

Resolution Theorem Proving

Properties of Resolution Theorem Proving:

- sound (for propositional and FOL)
- (refutation) complete (for propositional and FOL)

Procedure may seem heavy but note that can be easily automated. Just "smash" clauses until empty clause or no more new clauses.

Converting More Complicated Sentences to CNF

Sentence:

$$\forall x : brick(x) \rightarrow \begin{array}{l}((\exists y : on(x,y) \wedge \neg pyramid(y)) \\ \wedge\ (\neg\exists y : on(x,y) \wedge on(y,x)) \\ \wedge\ (\forall y : \neg brick(y) \rightarrow \neg equal(x,y)))\end{array}$$

CNF:

$$\neg brick(x) \vee on(x, support(x))$$
$$\neg brick(w) \vee \neg pyramid(support(w))$$
$$\neg brick(u) \vee \neg on(u,y) \vee \neg on(y,u)$$
$$\neg brick(v) \vee brick(z) \vee \neg equal(v,z)$$

Algorithm: Putting Axioms into Clause Form

1. Eliminate the implications.
2. Move the negations to the atomic formulas.
3. Eliminate the existential quantifiers.
4. Rename the variables, if necessary.
5. Move the universal quantifiers to the left.
6. Move the disjunctions down to the literals.
7. Eliminate the conjunctions.
8. Rename the variables, if necessary.
9. Eliminate the universal quantifiers.

1. Eliminate Implications

Substitute $(\neg E_1 \vee E_2)$ for $(E_1 \rightarrow E_2)$

$$\forall x : brick(x) \rightarrow ((\exists y : on(x,y) \wedge \neg pyramid(y)) \wedge (\neg\exists y : on(x,y) \wedge on(y,x)) \wedge (\forall y : \neg brick(y) \rightarrow \neg equal(x,y))$$

$$\forall x : \neg brick(x) \vee ((\exists y : on(x,y) \wedge \neg pyramid(y)) \wedge (\neg\exists y : on(x,y) \wedge on(y,x)) \wedge (\forall y : \neg(\neg brick(y)) \vee \neg equal(x,y))$$

2. Move negations down to the atomic formulas

Equivalence Transformations:

$$\neg(E_1 \wedge E_2) \iff (\neg E_1) \vee (\neg E_2)$$
$$\neg(E_1 \vee E_2) \iff (\neg E_1) \wedge (\neg E_2)$$
$$\neg(\neg E_1) \iff E_1$$
$$\neg\forall x : E_1(x) \iff \exists x : \neg E_1(x)$$
$$\neg\exists x : E_1(x) \iff \forall x : \neg E_1(x)$$

Result:

$$\forall x : \neg brick(x) \vee$$
$$((\exists y : on(x, y) \wedge \neg pyramid(y))$$
$$\wedge (\neg\exists y : on(x, y) \wedge on(y, x))$$
$$\wedge (\forall y : \neg(\neg brick(y)) \vee \neg equal(x, y)))$$

3. Eliminate Existential Quantifiers: Skolemization

Harder cases:

$\forall x : \exists y : father(y, x)$ becomes $\forall x : father(S1(x), x)$

There is one argument for each universally quantified variable whose scope contains the Skolem function.

Easy case:

$\exists x : President(x)$ becomes $President(S2)$

$\forall x : \neg brick(x) \vee ((\exists y : on(x, y) \wedge \neg pyramid(y)) \wedge \ldots$

4. Rename variables as necessary

We want no two variables of the same name.

$$\forall x : \neg brick(x) \vee \quad (on(x, S1(x)) \wedge \neg pyramid(S1(x))) \\ \wedge (\forall y : (\neg on(x, y) \vee \neg on(y, x))) \\ \wedge (\forall y : (brick(y) \vee \neg equal(x, y)))$$

$$\forall x : \neg brick(x) \vee \quad (on(x, S1(x)) \wedge \neg pyramid(S1(x))) \\ \wedge (\forall y : (\neg on(x, y) \vee \neg on(y, x))) \\ \wedge (\forall z : (brick(z) \vee \neg equal(x, z)))$$

5. Move the universal quantifiers to the left

This works because each quantifier uses a unique variable name.

$$\forall x : \neg brick(x) \vee \quad (on(x, S1(x)) \wedge \neg pyramid(S1(x))) \\ \wedge (\forall y : (\neg on(x, y) \vee \neg on(y, x))) \\ \wedge (\forall z : (brick(z) \vee \neg equal(x, z)))$$

$$\forall x \forall y \forall z : \neg brick(x) \vee \quad (on(x, S1(x)) \wedge \neg pyramid(S1(x))) \\ \wedge (\neg on(x, y) \vee \neg on(y, x)) \\ \wedge (brick(z) \vee \neg equal(x, z))$$

6. Move disjunctions down to the literals

$E_1 \vee (E_2 \wedge E_3) \iff (E_1 \vee E_2) \wedge (E_1 \vee E_3)$

$$\begin{aligned}\forall x \forall y \forall z : \; & (\boxed{\neg brick(x)} \vee \boxed{(on(x, S1(x)) \wedge \neg pyramid(S1(x)))}) \\ & \wedge (\neg brick(x) \vee \neg on(x, y) \vee \neg on(y, x)) \\ & \wedge (\neg brick(x) \vee brick(z) \vee \neg equal(x, z))\end{aligned}$$

$$\begin{aligned}\forall x \forall y \forall z : \; & (\neg brick(x) \vee on(x, S1(x))) \\ & \wedge (\neg brick(x) \vee \neg pyramid(S1(x))) \\ & \wedge (\neg brick(x) \vee \neg on(x, y) \vee \neg on(y, x)) \\ & \wedge (\neg brick(x) \vee brick(z) \vee \neg equal(x, z))\end{aligned}$$

7. Eliminate the conjunctions

$$\begin{aligned}\forall x \forall y \forall z : \; & (\neg brick(x) \vee on(x, S1(x))) \\ & \wedge (\neg brick(x) \vee \neg pyramid(S1(x))) \\ & \wedge (\neg brick(x) \vee \neg on(x, y) \vee \neg on(y, x)) \\ & \wedge (\neg brick(x) \vee brick(z) \vee \neg equal(x, z))\end{aligned}$$

$\forall x : \neg brick(x) \vee on(x, S1(x))$
$\forall x : \neg brick(x) \vee \neg pyramid(S1(x))$
$\forall x \forall y : \neg brick(x) \vee \neg on(x, y) \vee \neg on(y, x)$
$\forall x \forall z : \neg brick(x) \vee brick(z) \vee \neg equal(x, z)$

8. Rename all variables, as necessary, so no two have the same name

$\forall x : \neg brick(x) \vee on(x, S1(x))$
$\forall x : \neg brick(x) \vee \neg pyramid(S1(x))$
$\forall x \forall y : \neg brick(x) \vee \neg on(x, y) \vee \neg on(y, x)$
$\forall x \forall z : \neg brick(x) \vee brick(z) \vee \neg equal(x, z)$

$\forall x : \neg brick(x) \vee on(x, S1(x))$
$\forall w : \neg brick(w) \vee \neg pyramid(S1(w))$
$\forall u \forall y : \neg brick(u) \vee \neg on(u, y) \vee \neg on(y, u)$
$\forall v \forall z : \neg brick(v) \vee brick(z) \vee \neg equal(v, z)$

9. Eliminate the universal quantifiers

$\neg brick(x) \vee on(x, S1(x))$
$\neg brick(w) \vee \neg pyramid(S1(w))$
$\neg brick(u) \vee \neg on(u, y) \vee \neg on(y, u)$
$\neg brick(v) \vee brick(z) \vee \neg equal(v, z)$