Continue with predicate logic

Implication

- $\Longrightarrow$ *implication* $(A \Longrightarrow B)$ *T / F*
- *There is No doubt about conjunction, disjunction, and negation*

Let say the following

if I win the election then I will lower your tuition.

T	T	T
T	F	F
F	T	T
F	T	T

- *Why it is important ! Because it very good basis for basic rule of reasoning it is called* modus ponens

Basic Terminology

- *Basic operators* v , $\wedge$, $\Rightarrow$, $\neg$
  disjunction *conjunction* *implication* *negation*

- *Prosperities*

1. *OR* *AND*
   v and $\wedge$ are cumulative
   $A \vee B \triangleq B \vee A$, $A \wedge B \triangleq B \wedge A$

2. v and $\wedge$ are distributive
   $A \wedge (B \vee C) \triangleq (A \vee B) \wedge (A \vee C)$
   $A \vee (B \wedge C) \triangleq (A \vee B) \wedge (A \vee C)$

3. De Morgan's laws
   $\neg (A \vee B) \triangleq \neg A \wedge \neg B$
   $\neg (A \wedge B) \triangleq \neg A \vee \neg B$

4. contrapositive law
   $A \Rightarrow B \triangleq \neg B \Rightarrow \neg A$

5. Quantifier
   $\forall$ *for all*
   $\exists$ *there exists,*

$\forall x\, (\underbrace{P}_{\text{Predicate}}(x))$ *Suppose* $\exists x\, P(x)$

How to interpreter this?

There exists X

There exists student in this class who owns IPhone X

$\neg\,(\exists x\, P(x)) \stackrel{\Delta}{=} \forall x\, (\neg P(x))$

It is not true that there exists student in this class who owns Motorola V100 which means as said for all students seating in this classroom, it is not true that they have basically Motorola.

$\neg \forall x\, P(x) \stackrel{\Delta}{=} \exists x\, \neg P(x)$

<u>This the hard part</u>

Writing what it is called ………………. Formal the logic formal that has quantifier, predicate, implication etc. and interpreted then writing sentence.

- *The interpreter process is easy where is kind of modeling particular sentence as predicate logical expression, it is typically more difficult.*
- *In the exam providing predicate, expression*

<u>Example</u>

$\forall x \quad \forall y$ *{ {Block (x) ^ Block (y) ^ [On-top(x,y) v attached(x,y)] ^ Moved(y)}=> moved (x) }*

How to interpreter this?

- *Writing English sentence*

All blocks on-top of blocks that have been moved or that are attached to blocks that have been moved have also been moved

<u>Example (exercise done before next class)</u>

Every city has a dogcatcher who has been bitten by every dog in the town

Proofs

A1 *Barks (fido)*

A2 $\forall\, x\ (Barks\ (x) \Rightarrow Dog(x))$

A1 & A2 are Axiom

Theorem *Fido is a dog (dog (fido))*

Dog (Fido) is theorem

- *Axiom is always true*

How to prove it?

- *In the logic, need to use sound rule of reasoning.*

Sound rule of reasoning

1. *Modus ponens*
   if A is true and (A => B) is true then B must be true

A1 *Barks (fido)* *T*

A2 $\forall\, x\ (Barks\ (x) \Rightarrow Dog(x))$ *T*

For < x , f , Do > x<= fido

Barks (fido) => Dog (fido)

By Modus ponens (MP)

Dog(Fido) must be true

2. *Resolution*

Clause (expression in a disjunction form)

if *A v B and ¬ B v C*

then *A v C* *Resolvent*

A v B and ¬ B v C

Axiom 1 *A v B*

Axiom 2 *¬ B v C*

$A \Rightarrow B \triangleq \neg A \vee B$

Barks (Fido)

Barks (fido) $\Rightarrow$ Dog (fido) $\triangleq$ $\neg$ Barks (fido) $\vee$ Dog (fido)

Dog (fido)

<u>*Notes*</u>

- *There is coming up quiz in April. 13*
- *There are two homework left.*
    1. *Written homework*
    2. *Coding homeworking (as team)*
- *Final exam*

- *Express particular condition through predicate what is called well-formed formula (WFF)*
    - *Predicate with negation with conjunction disjunction implication and quantifier from well-formed formula.*
    - *Take a list of* [*Axioms And have theorem*] *employ sound rules of reasoning to prove the theorem*
    - *Do it* → *"Automatically"*

Refereeing to the idea of building AI production system that can reason and prove theorem

This is the process and is not trial because have to put pieces together

Solving the pervious exercise dogcatchers

Every city has a dogcatcher who has been bitten by every dog in the town

Let propose WFF

$\forall x \{ City(x) \Rightarrow \exists y \{ Dog\ catcher(x,y) \wedge \forall z \{ [Dog(z) \wedge lives\text{-}in(x,z)] \Rightarrow bitten(y,z)\}\}\}$

Continue using Resolution

Proving theorem by resolution using refutation

Clauses

$E_1 \vee E_2 \qquad \neg E_2 \vee E_3$

$E_1 \vee E_3$ *(resolvent)*

Note: to something be called clauses

- *The well-formed formula (WFF)has to be in disjunction form*
- *Clauses cannot have implication and conjunction*

For proving

1. *Given list of axioms*
2. *Given a theorem*
3. *Ensure all axioms are in clause form*
4. *Negate the theorem and attach it to the list of axioms*
5. *Resolve*
6. *If "NIL" is obtain, your theorem proving*

Trying to prove some properties

The process

- *Take theorem and negate then attach it to the list of axioms*
- *To be able to resolve using the resolution principle when very thing given in conjunction form*

Writing something like

$P(x) \vee Q(x) \qquad \neg P(x) \vee T(x)$

How to do this automatically? $\qquad Q(x) \vee T(x)$

- *Scan for the name of this predicate $P(x)$ and check if there is another clause $\neg P(x)$ it has the same item that is negated*

How this work

- *Having everything in the clause form include the negated theorem*

Example

Barks (Fido)

Axiom

1. *Barks (Fido)*
2. $\forall x\,(\,Barks(x) \Rightarrow Dog(x)\,)$

Theorem

Dog (Fido)

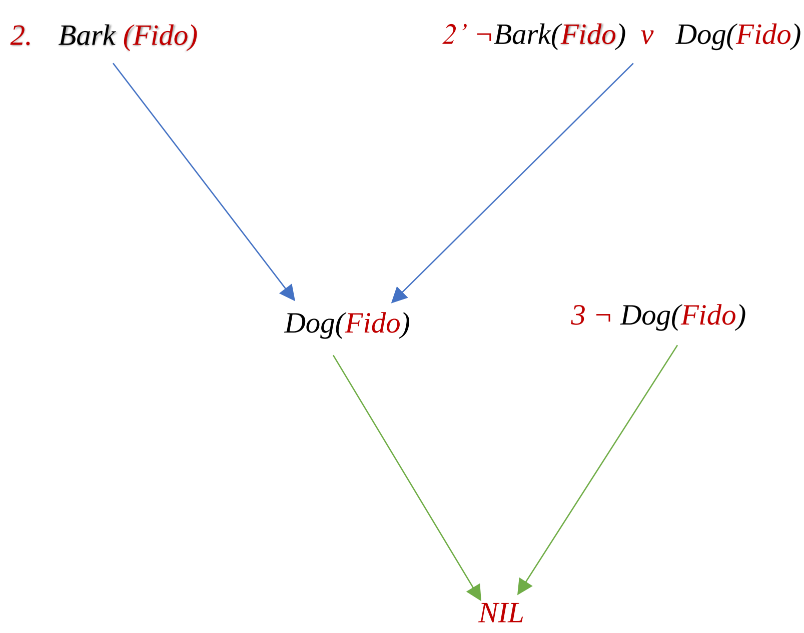
1. Bark (Fido)

1. *Need v disjunction rather than implication =>*
   Substitute implication with disjunction form
   $\forall x\,(\,Bark(x) \lor Dog(x)\,)$
   Drop the quantifier

2' $\neg Bark(Fido) \lor Dog(Fido)$

$< x, Fido >$

3 $\neg Dog(Fido)$



Hint for finding resolvents

1. $P \vee Q$ $\quad\quad\quad \neg P \vee Q$

   Q

2. P $\quad\quad\quad \neg P \vee Q$

   Q

3. $P \vee Q$ $\quad\quad\quad \neg P \vee Q$

   $P \vee \neg P \rightarrow T$

   $Q \vee \neg Q \rightarrow T$

4. P $\quad\quad\quad \neg P$

   NIL

- *Convert these to axioms and theorem*

Example

Head, I Tails, You

Prove that I win

Axioms

1. $H \Rightarrow win(I)$
2. $T \Rightarrow lose(you)$
3. $\neg H \Rightarrow T$
4. $lose(you) \Rightarrow win(I)$

Prove

Theorem win (I)

Going through steps

All axioms in clause form

1. ¬H v win(i)
2. ¬T v lose (you)
3. H v T
4. ¬ lose (you) v win (i)
5. ¬ win (i)

Do you see any analogy to AI production system?

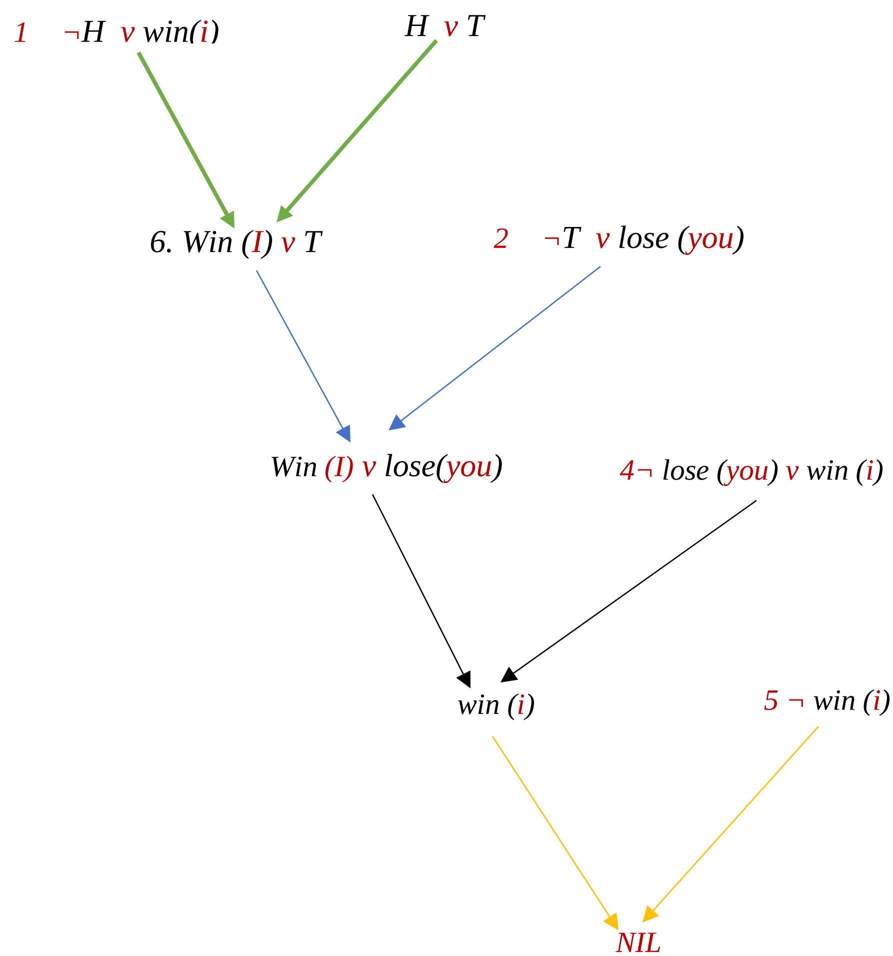
- *Database*
- *Operator*
- *Control strategy*

¬H v win(i)   H v T

resolution

Win (I) v T

- *The initial state is the above list (1-5)*
- *The goal state is empty clause the "NIL"*



<u>*Key requirement for the axioms to work*</u>

- *Clause form*
- *Unified*

<u>*For the next lecture*</u>

- *In the d2l, look for clause form example in the content*
  - *Clause form conversion example*
  - *Existential quantifier skolem*

- *Continue showing how to convert axioms and theorem to normal disjunction form (clause form)*
- *All predicates P(x) and negate predicates ¬P(x) are given in form of disjunction and there is no implication => and conjunction ^ in the form*

Proof using resolution refutation

- *Axioms and negated theorem have to be in clause form*

How do we convert a well-formed formula (WFF) to a clause?

Well-Formed Formula (WFF)

It is predicate and negate predicate that join by conjunction, disjunction, implication and all of these sentences could be quantify with universal and existential quantifier.

Example

$\forall x \{ P(x) \Rightarrow \{ \forall y\, [P(y) \Rightarrow P(f(x,y))] \wedge \neg (\forall y)\; [Q(x,y) \Rightarrow P(y)]\}\}$

- *Think of this as initial state and what would you accomplish in the end?*
- *In the end, would break up this formal into sets of axioms (A_1 , A_2 , A_3) and it has everything in disjunctive form*

What is the motivation for doing this?

The motivation is to run this through automated string processing program that will resolve things if the resolution exists and ultimately prove the theorem

How to do this?

1. *Eliminate implications*

   $\forall x \{ P(x) \Rightarrow \{ \forall y\, [P(y) \Rightarrow P(f(x,y))] \wedge \neg (\forall y)\; [Q(x,y) \Rightarrow P(y)]\}\}$

   $\forall x \{ \neg P(x) \vee \{ \forall y\, [\neg P(y) \vee P(f(x,y))] \wedge \neg (\forall y)\; [\neg Q(x,y) \vee P(y)]\}\}$

2. *Move negation down to atomic predicate*

   $\forall x \{ \neg P(x) \vee \{ \forall y\, [\neg P(y) \vee P(f(x,y))] \wedge \boxed{\neg (\forall y)\; [\neg Q(x,y) \vee P(y)]}\}\}$

   $\forall x \{ \neg P(x) \vee \{ \forall y\, [\neg P(y) \vee P(f(x,y))] \wedge \boxed{(\exists y)\; [\neg(\neg Q(x,y) \vee P(y))]}\}\}$

   $\forall x \{ \neg P(x) \vee \{ \forall y\, [\neg P(y) \vee P(f(x,y))] \wedge \boxed{(\exists y)\; [Q(x,y) \wedge \neg P(y)}\,]\}\}$

3. Purge existential quantifiers

Using Skolem function

If have $\exists x \; P(x)$ $\qquad$ $P(A)$

The general rule for eliminating an existential quantifier from a wff is to replace each occurrence of its existentially quantified variable by a Skolem function whose arguments are those universally quantified variables that are bound by universal quantifiers whose scopes include the scope of the existential quantifier being eliminated. Function symbols used in Skolem functions must be new in the sense that they cannot be ones that already occur in the wff. Thus, we can eliminate the $(\exists z)$ from

$$[(\forall w)Q(w)] \Rightarrow (\forall x)\{(\forall y)\{(\exists z)[P(x,y,z) \Rightarrow (\forall u)R(x,y,u,z)]\}\},$$

to yield

$$[(\forall w)Q(w)] \Rightarrow (\forall x)\{(\forall y)[P(x,y,g(x,y)) \Rightarrow (\forall u)R(x,y,u,g(x,y))] \;.$$

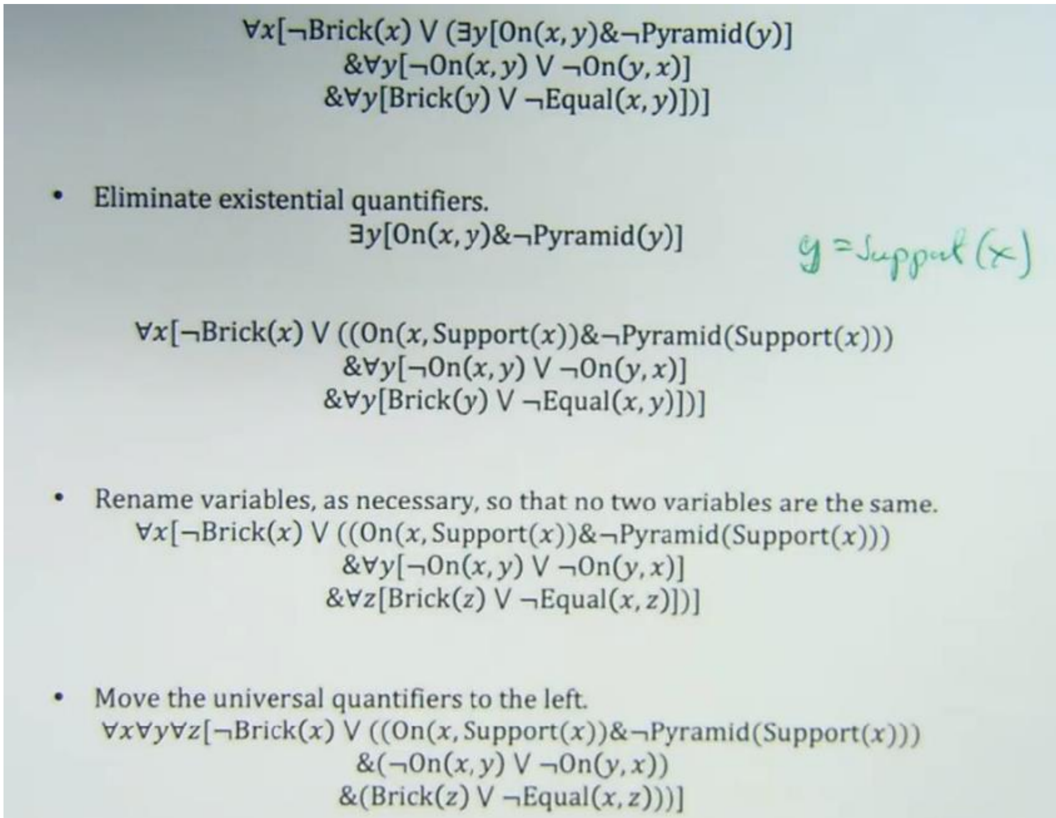
If the existential quantifier being eliminated is not within the scope of any universal quantifiers, we use a Skolem function of no arguments, which is just a constant. Thus, $(\exists x)P(x)$ becomes $P(A)$, where the constant symbol A is used to refer to the entity that we know exists. It is important that A be a new constant symbol and not one used in other formulas to refer to known entities.

To eliminate all of the existentially quantified variables from a wff, we use the above procedure on each formula in turn. Eliminating the existential quantifiers (there is just one) in our example wff yields:

$$(\forall x)\{\sim P(x) \vee \{(\forall y)[\sim P(y) \vee P(f(x,y))] \wedge [Q(x,g(x)) \wedge \sim P(g(x))]\}\},$$

Example

$\forall x\, [\neg\, Brick\,(x) \;v\; (\,\exists\, y\, [\, On\,(x,y) \wedge \neg Pyramid(y)]\&\, \forall\, y[\, \neg On\,(x,y)\; v\; \neg On\,(x,y)\,]\;\&\; \forall\, y[\, \neg Brick\,(y)\; v\; \neg Equal\,(x,y)])]$



$$\forall x[\neg \mathrm{Brick}(x) \vee (\exists y[\mathrm{On}(x,y)\&\neg \mathrm{Pyramid}(y)]$$
$$\&\forall y[\neg \mathrm{On}(x,y) \vee \neg \mathrm{On}(y,x)]$$
$$\&\forall y[\mathrm{Brick}(y) \vee \neg \mathrm{Equal}(x,y)])]$$

- Eliminate existential quantifiers.

$$\exists y[\mathrm{On}(x,y)\&\neg \mathrm{Pyramid}(y)] \qquad y = Support(x)$$

$$\forall x[\neg \mathrm{Brick}(x) \vee ((\mathrm{On}(x,\mathrm{Support}(x))\&\neg \mathrm{Pyramid}(\mathrm{Support}(x)))$$
$$\&\forall y[\neg \mathrm{On}(x,y) \vee \neg \mathrm{On}(y,x)]$$
$$\&\forall y[\mathrm{Brick}(y) \vee \neg \mathrm{Equal}(x,y)])]$$

- Rename variables, as necessary, so that no two variables are the same.

$$\forall x[\neg \mathrm{Brick}(x) \vee ((\mathrm{On}(x,\mathrm{Support}(x))\&\neg \mathrm{Pyramid}(\mathrm{Support}(x)))$$
$$\&\forall y[\neg \mathrm{On}(x,y) \vee \neg \mathrm{On}(y,x)]$$
$$\&\forall z[\mathrm{Brick}(z) \vee \neg \mathrm{Equal}(x,z)])]$$

- Move the universal quantifiers to the left.

$$\forall x \forall y \forall z[\neg \mathrm{Brick}(x) \vee ((\mathrm{On}(x,\mathrm{Support}(x))\&\neg \mathrm{Pyramid}(\mathrm{Support}(x)))$$
$$\&(\neg \mathrm{On}(x,y) \vee \neg \mathrm{On}(y,x))$$
$$\&(\mathrm{Brick}(z) \vee \neg \mathrm{Equal}(x,z)))]$$

$$\forall\, x\, \{\neg\, P\,(x)\;\; v\, \{\, \forall\, y\, [\neg\, P\,(y)\; v\;\; P\,(f\,(x,y))\,]\wedge [\, Q\,(x,g(x)) \wedge \neg P(g(x))]\}\}$$

Where $g(x)=y$ is a Skolem function

4. *Rename variables (if needed)*

5. *Move ∀ to the left (as far left as possible)*

$$\forall\, x\, \forall\, y\, \{\neg\, P\,(x)\; v\; [\neg\, P\,(y)\; v\; P\,(f\,(x,y))\,] \wedge [\, Q\,(x,g(x)) \wedge \neg P(g(x))]\}\}$$

6. *Move the disjunctions down to atomic expression*

$$x_1 \;\; v\, [\, x_2 \wedge x_2\,] \stackrel{\Delta}{=} (\, x_1\, v\, x_2\,) \wedge (\, x_1\, v\, x_3\,)$$

$$\forall\, x\, \forall\, y\, \{\, [\, \neg\, P\,(x)\;\; v\;\; \neg\, P\,(y)\; v\;\; P\,(f\,(x,y))\,] \wedge \boxed{[\neg\, P\,(x)\; v\; [\, Q\,(x,g(x)) \wedge \neg P(g(x))]]}\}.$$

$$\forall\, x\, \forall\, y\, \{\, [\, \neg\, P\,(x)\;\; v\;\; \neg\, P\,(y)\; v\;\; P\,(f\,(x,y))\,] \wedge [\neg\, P\,(x)\; v\; Q\,(x,g(x))\,] \wedge [\, \neg\, P\,(x)\; v\; \neg P(g(x))]]\}$$

7. *Eliminate conjunctions*

$\boxed{\forall x \forall y \quad [\neg P(x) \lor \neg P(y) \lor P(f(x,y))]}$ → ***Axioma 1***

$\boxed{\forall x \forall y \quad [\neg P(x) \lor Q(x,g(x))]}$ → ***Axioma 2***

$\boxed{\forall x \forall y \quad [\neg P(x) \lor \neg P(g(x))]}$ → ***Axioma 3***

8. *Eliminate* $\forall$

$\boxed{\begin{array}{l} \forall x \forall y \\ \forall x \forall y \\ \forall x \forall y \end{array}} \begin{array}{l} [\neg P(x) \lor \neg P(y) \lor P(f(x,y))] \\ [\neg P(x) \lor Q(x,g(x))] \\ [\neg P(x) \lor \neg P(g(x))] \end{array}$

$[\neg P(x) \lor \neg P(y) \lor P(f(x,y))]$

$[\neg P(x) \lor Q(x,g(x))]$

$[\neg P(x) \lor \neg P(g(x))]$

<u>Exercise</u>

Brick is on something there is not pyramid there is nothing brick on and on as brick well the there is nothing not the brick also, same thing is brick.

$\forall x$ [Brick (x) => ($\exists y$ [On (x,y) & $\neg$ Pyramid(y)] $\wedge \neg \exists y$[On $(x,y) \wedge$ On (x,y)] $\wedge$
$\forall y$[$\neg$Brick (y) => $\neg$Equal (x,y)])]