Introduction to Logic
CSA1070

Kristian Guillaumier
http://webster.cs.um.edu.mt/kguil
kguil@cs.um.edu.mt

What is Logic?

- Informally, Logic is a tool that allows us to derive a legitimate conclusion(s) from a set of given premises.
- Logic is described in a language called **First Order Logic (FOL)**.
- We need to use such notation because:
  - English is ambiguous.
  - Convention and Clarity are crucial.

Atomic Sentences

- You can think of **Atomic Sentences** as the most basic of English sentences consisting of names connected by **predicates**:
  - Andrew ran.
  - Andrew saw Mark.
- Individual Constants:
  - These are symbols used to refer to some fixed individual **object** e.g. Andrew.
  - The FOL equivalent of names.

Individual Constants

- Individual Constants:
  - However in FOL these names are required to refer to **exactly one** object. In English it may be possible state that **Andrew saw Andrew**, where the two Andrews are different people. In FOL this is unacceptable.
  - An object may have more than one name. For example Andrew and Andy may refer to the same person (think of a unique nickname).

Predicate Symbols (1)

- Predicate symbols are used to denote:
  - A property of an object.
  - A relation between one or more objects.
- In Andrew saw Mark,
  - **saw** is the predicate.
  - Andrew, and Mark become the **arguments** of the predicate.
  - In this case the predicate saw is used as a **binary predicate**, since it takes two arguments.

Predicate Symbols (2)

- In English some predicates may be optional:
  - Andrew saw
  - Andrew saw Mark
- In FOL however, each predicate has a fixed number of arguments. This number is called the **arity** (from un**ary**, bin**ary**, etc…).
- Examples:
  - **AtHome(**Andrew**)**
  - **Taller(**Andrew,Mark**)**
  - **Between(**6,5,7**)**

Predicates Symbols (3)

- In English, predicates are sometimes vague. For example, **Claire** (who is six years old) **is young**. She will not be young when she's 96 but there is no definite point where she stops being so. In contrast, in FOL predicates are interpreted by a **determinate property or relation**. (A definite fact).

Atomic Sentences

- If we wish to say that a is equal to be, we write **a = b**. The '=' predicate is written in **infix** notation (i.e. it lies between the arguments).
- The predicate Taller in Taller(Andrew,Mark) is written in **prefix** because it precedes the arguments.
- The order of arguments is important Taller(Andrew,Mark) is not the same as Taller(Mark,Andrew).

Propositions and Arguments (1)

- In FOL and Atomic Sentence are used to express **propositions** (claims about something).
- These claims are either **TRUE** or **FALSE**.
- Central in Logic, is the notion of a **argument** (not the same 'argument' seen previously).
- An argument is also called an **inference**.
- This is a sequence of propositions (**premises**) used to derive a **conclusion**.

Propositions and Arguments (2)

If I'm outside and it's raining then I'll get wet

Premises: I'm outside
It is raining

Conclusion: Therefore, I get wet

Validity of Arguments (1)

1. All men are mortal. Socrates is a man. Therefore, Socrates is mortal.
2. Larry is mortal and all men are mortal. So Larry is a man.

- The first is **logically valid** since its conclusion is a **logical consequence** of its premises. It is impossible for this conclusion to be false if the premises are true.
- What if Larry is a goldfish???

Validity of Arguments (2)

- Informally, we can say that:

  An argument is valid if and only if the conclusion must be true on the assumption that the premises are true.
- Note: this does not necessarily mean that the arguments must be true for the argument to be valid (e.g. if Socrates is a robot). In this case the conclusion is not guaranteed to be true.
- So… IF the premises are true AND the argument is valid THEN the conclusion is true.

Validity of Arguments (3)

Hamlet is red in the face
Hamlet is holding an empty bottle
Hamlet is swearing
Hamlet just fell over

Therefore, Hamlet is drunk

- Even though, the premises may be true, the argument may still be invalid if Hamlet was acting.

Validity of Arguments (4)

The SLT is in University
The University is in Malta

Therefore, the SLT is in Malta

- This is a good argument. If we accept the premises then we also accept the conclusion.

Compound Statements (1)

- Remember, sentences like “Andrew saw Mark” are atomic because they cannot be decomposed into smaller ones.
- Longer sentences are built by combining atomic sentences into compound ones:

  Andrew saw Mark <u>and</u> Mark was Swimming
- Atomic Sentences:
  - saw(Andrew,Mark)
  - was(Mark,Swimming)

Compound Statements (2)

- Compound sentence:
  - saw(Andrew,Mark) AND was(Mark,Swimming)

This is the SLT (atomic)

This is <u>not</u> the SLT (compound)

- Atomic Sentence:
  - is(This,SLT)
- Compound Statement:
  - NOT is(This,SLT)

Connectives

- The connectives AND, NOT, OR, etc… are described in FOL as follows:

English Version	FOL Name	Symbol
Not	Negation	¬
And	Conjunction	∧
Or	Disjunction	∨
If… Then…	Implication	→
…is equivalent to… …if and only if… (Andrew is alive iff his heart is beating)	Biconditional	↔

Negation (NOT)

- Consider, **strong(Peter)** and ¬**strong(Peter)**.
- Negation is a unary operator – it takes one argument.
- Truth table:

p	¬p
T	F
F	T

Conjunction (AND)

- **strong(Peter)** $\wedge$ **tall(Peter)**.
- Conjunction is a binary, infix operator – it takes two arguments.
- If **p** and **q** are propositions, the statement **p** $\wedge$ **q** is true iff both **p** and **q** are true.

p	**q**	**p** $\wedge$ **q**
T	T	T
T	F	F
F	T	F
F	F	F

Disjunction (OR)

- **strong(Peter)** $\vee$ **tall(Peter)**.
- Disjunction is a binary, infix operator – it takes two arguments.
- If **p** and **q** are propositions, the statement **p** $\vee$ **q** is true iff either **p** or **q** are true.

p	**q**	**p** $\vee$ **q**
T	T	T
T	F	T
F	T	T
F	F	F

Implication (If… Then…)

- **strong(Peter) → tall(Peter)**.
- Implication is a binary, infix operator – it takes two arguments.
- If **p** and **q** are propositions, the statement **p → q** is true iff either **p** if false or **q** is true (or both).
- **p** is called the **antecedent** of the implication whilst **q** is called the **consequent**.
- An implication is sometimes called a **conditional**.

Tip: → is only false when the antecedent is true but the consequent is not.

p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

Biconditional (iff)

- 'An implication that works both ways'.
- If **p** and **q** are propositions, the statement **p ↔ q** is true iff **p** and **q** share the same truth value.

p	q	p ↔ q
T	T	T
T	F	F
F	T	F
F	F	T

More on Truth Tables

- Truth tables help when determining the truth or falsity of compound statements. For example:

  $(p \vee q \wedge r) \rightarrow \neg p \vee r$

p	q	r	$q \wedge r$	$p \vee q \wedge r$	$\neg p$	$\neg p \vee r$	$(p \vee q \wedge r) \rightarrow \neg p \vee r$
T	T	T	T	T	F	T	T
T	T	F	F	T	F	F	F
T	F	T	F	T	F	T	T
T	F	F	F	T	F	F	F
F	T	T	T	T	T	T	T
F	T	F	F	F	T	T	T
F	F	T	F	F	T	T	T
F	F	F	F	F	T	T	T

Logical Equivalence

- Two statements P and Q are **logically equivalent**, if they are true in excactly the same circumstances.
- Logical equivalence is written $\mathbf{P \equiv Q}$
- So...
  - $P \equiv \neg\neg P$
  - $P \wedge (Q \wedge R) \equiv (P \wedge Q) \wedge R \equiv P \wedge Q \wedge R$
  - $\neg(P \wedge Q) \equiv (\neg P \vee \neg Q)$ **(DeMorgan)**
  - $\neg(P \vee Q) \equiv (\neg P \wedge \neg Q)$ **(DeMorgan)**

Logical Equivalence Example

- Show that,

  $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$

P	Q	R	$Q \vee R$	$P \wedge (Q \vee R)$	$P \wedge Q$	$P \wedge R$	$(P \wedge Q) \vee (P \wedge R)$
T	T	T	T	T	T	T	T
T	T	F	T	T	T	F	T
T	F	T	T	T	F	T	T
T	F	F	F	F	F	F	F
F	T	T	T	F	F	F	F
F	T	F	T	F	F	F	F
F	F	T	T	F	F	F	F
F	F	F	F	F	F	F	F

Tautologies

- Statements like '*Andrew plays football or he does not* ' are evidently true whatsoever.
- Tautologies are statements that are always true irrespective of the truth values of their atomic statements.
- Again, a truth table may be used to determine whether a statement is a tautology.
- Check if $p \vee \neg p$ is a tautology.
- A tautology is logically equivalent to TRUE.

Contradictions

- A contradiction is the opposite of a tautology.
- Irrespective of the truth values of the atomic sentences, the statement is always false.
- Andrew plays football and he does not play football ($P \wedge \neg P$).
- A contradiction is logically equivalent to FALSE.

Duality

- Let X be a statement involving $\neg$, $\vee$ and $\wedge$.
- Let X* be the statement X with:
  - $\vee$ replaced by $\wedge$
  - $\wedge$ replaced by $\vee$
  - TRUE replaced by FALSE
  - FALSE replaced by TRUE
- X* is the **dual** of X
- Notes:

  $\mathbf{P} \equiv \mathbf{Q}$ then $\mathbf{P^*} \equiv \mathbf{Q^*}$

  $(\mathbf{P} \wedge \mathbf{Q}) \vee \mathbf{R}$ is the dual of $(\mathbf{P} \vee \mathbf{Q}) \wedge \mathbf{R}$

  $(\mathbf{P^*})^* \equiv \mathbf{P}$

Laws of Logic (1)

- Notation:
  - P, Q and R are atomic statements.
  - $\mathbf{T_0}$ is a tautology.
  - $\mathbf{F_0}$ is a contradiction.

- Commutative Law:

  $\mathbf{P \wedge Q \equiv Q \wedge P}$ $\qquad$ $\mathbf{P \vee Q \equiv Q \vee P}$

- Associative Law:

  $\mathbf{(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)}$ $\qquad$ $\mathbf{(P \vee Q) \vee R \equiv P \vee (Q \vee R)}$

Laws of Logic (2)

- Distributive Law:

  $\mathbf{P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)}$

  $\mathbf{P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)}$

- Identity Law:

  $\mathbf{P \wedge T_0 \equiv P}$ $\qquad$ $\mathbf{P \vee F_0 \equiv P}$

- Inverse Law:

  $\mathbf{P \wedge \neg P \equiv F_0}$ $\qquad$ $\mathbf{P \vee \neg P \equiv T_0}$

Laws of Logic (3)

- Double Negation Law:

  $\neg(\neg P) \equiv P$

- Idempotent Law:

  $P \wedge P \equiv P$ $\qquad$ $P \vee P \equiv P$

- DeMorgan's Laws:

  $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ $\qquad$ $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$

Laws of Logic (4)

- Domination Law:

  $P \vee T_0 \equiv T_0$ $\qquad$ $P \wedge F_0 \equiv F_0$

- Absorption Law:

  $P \wedge (P \vee Q) \equiv P$ $\qquad$ $P \vee (P \wedge Q) \equiv P$

- Negation Law:

  $\neg T_0 \equiv F_0$ $\qquad$ $\neg F_0 \equiv T_0$

Satisfiability (1)

- Intuitively a sentence is satisfiable if it could be true, at least on logical grounds.
- It is irrelevant whether it is physically impossible. For example: RunsFasterThanLight(Andrew) is satisfiable.
- A contradiction can never be satisfied.
- A set of sentences is satisfiable if there is a circumstance under which all sentences are simultaneously true.

Satisfiability (2)

$$\text{Happy(Mark)} \vee \text{Happy(Andrew)}$$
$$\neg\text{Happy(Mark)}$$
$$\neg\text{Happy(Andrew)}$$

- In this example, all sentences are satisfiable.
- Any two are satisfiable.
- But all three can never be true simultaneously (i.e. the set of sentences is not satisfiable).

Logical Truth

- A sentence is logically true regardless of the circumstances.
- For example:

  Home(Andrew) $\vee$ $\neg$**Home(Andrew)**

  $\neg$**(Happy(Andrew)** $\wedge$ $\neg$**Happy(Andrew))**
- All tautologies are logically true, but not all logically true sentences are tautologies.

Logical Truth vs. Satifiability vs. Tautologies (1)

- Consider the sentence:

  [Box(a) $\wedge$ **Box(b)]** $\vee$ $\neg$ **Box(c)**

A	B	C	A $\wedge$ B	$\neg$C	(A $\wedge$ B) $\vee$ $\neg$C
T	T	T	T	F	T
T	T	F	T	T	T
T	F	T	F	F	F
T	F	F	F	T	T
F	T	T	F	F	F
F	T	F	F	T	T
F	F	T	F	F	F
F	T	F	F	T	T

Logical Truth vs. Satifiability vs. Tautologies (2)

- Now, suppose the abbreviation **A** is **Circle(C)**.
  [Circle(c) ∧ Box(b)] ∨ ¬ Box(c)

A	B	C	A ∧ B	¬C	(A ∧ B) ∨ ¬C
T	T	T	T	F	T
T	T	F	T	T	T
T	F	T	F	F	F
T	F	F	F	T	T
F	T	T	F	F	F
F	T	F	F	T	T
F	F	T	F	F	F
F	T	F	F	T	T

Rows are **SPURIOUS** since A and C can never be true together (c can never be a box and circle at the same time)

Logical Truth vs. Satifiability vs. Tautologies (3)

- To determine if a sentence S is satisfiable, logically true or a tautology:

1. Construct the true table for S.
2. S is a tautology IFF it is TRUE in every row.
3. Eliminate any spurious rows.
4. S is satisfiable IFF there is at least one row TRUE.
5. S is logically true IFF all the remaining rows are TRUE.

Variables

- So far we have discussed the use of constants in relation to a predicate. For example, we wrote:

  IsHome(John)

  IsTaller(John,Mark)
- Now we introduce the notion of a **variable**.
- Consider the following expression using a variable:

  IsTaller(John,y)
- In FOL there can be infinitely many variables.
- By convention the letters u, v, w, x, y, and z are used.

Atomic Well-Formed Formulas

- **Atomic WFFs** are like the normal sentences we have seen, but make use of one or more variables in place of constants.
- So the following are Atomic WFFs:
  - IsHome(x)
  - IsTaller(John,z)
  - IsTaller(x,y)

Quantifiers

- An atomic WFF is not regarded as a sentence.
- However they can be converted into sentences by applying **quantifiers**.
- The quantifiers considered here are:
  - $\exists$ **Existential Quantifier "There exists"**
  - $\forall$ **Universal Quantifier "For all"**
- These allow us to specify that **all objects** or **at least one object** satisfies a condition.
- "There does not exist" may be written as $\neg\exists$ or $\nexists$

Universal Quantifier

- The universal quantifier is used to express claims like:

Everyone is at home	$\forall x$ AtHome(x)
Every lecturer is cool	$\forall x$ (Lecturer(x)$\rightarrow$Cool(x))

Existential Quantifier

- The existential quantifier is used to express claims like:

Someone is at home	$\exists$x AtHome(x)
At least a lecturer is cool	$\exists$x (Lecturer(x) $\wedge$ Cool(x))

Why do we need Quantifiers?

- One way of expressing the universal quantification $\forall$**x P(x)** as:

  $\mathbf{P(n_1) \wedge P(n_2) \wedge P(n_3) \wedge \ldots}$
- Similarly we can express the existential quantification of $\exists$**x P(x)** as:

  $\mathbf{P(n_1) \vee P(n_2) \vee P(n_3) \vee \ldots}$
- This becomes impractical. Say we are considering **x** to be:
  - Grains of sand (this is impractical)
  - The real numbers (this is impossible)
- Informally the universe of discourse is the “set of individual objects we are discussing now”.

DeMorgan's Laws for Quantifiers

- $\neg\forall x\ P(x) \equiv \exists x\ \neg P(x)$

  To see why:

  $\neg P(n_1) \vee \neg P(n_2) \vee \neg P(n_3) \vee \ldots$

- $\neg\exists x\ P(x) \equiv \forall x\ \neg P(x)$

  To see why:

  $\neg P(n_1) \wedge \neg P(n_2) \wedge \neg P(n_3) \wedge \ldots$

Free and Bound Variables

- If **P** is a WFF (well-formed-formula) and **v** is a variable then $\forall$**v P** and $\exists$**v P** are WFF too is **v** is bound to **P**.

$\forall x$ (Pet(x) $\rightarrow$ Noisy(x)) **All occurences of x are bound within the scope of $\forall$**

$\exists x$ (Pet(x) $\wedge$ Noisy(y)) **y is not bound within the scope of $\exists$**

- A sentence is a WFF of it has no unbound (free) variables.

Notes on Quantification

- $\forall$ is often used when we wish to say things like 'every P is a Q'
  - $\forall x\ (P(x) \rightarrow Q(x))$
- $\exists$ is often used when we wish to say things like 'There is a P which also has the property Q'
  - $\exists x\ (P(x) \wedge Q(x))$
- Sometimes the latter sentence is incorrectly translated to $\exists x\ (P(x) \rightarrow Q(x))$
- The meaning is different – compare the truth tables for $\wedge$ and $\rightarrow$.

Vacuously True Sentences

- Consider the expression $\forall$**x (Pet(x)** $\rightarrow$ **Noisy(x))** in a world where pets do not exist.
- Clearly nothing can possibly satisfy the first part of the implication – Pet(x) = True, so from the truth table of $\rightarrow$, all the sentence will always be true.

p	**q**	**p $\rightarrow$ q**
T	T	T
T	F	F
F	T	T
F	F	T

- We conclude that a universal statement is Vacuously true in a world where the first part does not hold.

Complex Translations (1)

- Every small dog that is at home is happy:

  $\forall x\ (((Small(x) \wedge Dog(x) \wedge AtHome(x)) \rightarrow Happy(x))$

- A small dog at home is happy:

  $\exists x\ (Small(x) \wedge Dog(x) \wedge AtHome(x) \wedge Happy(x))$

Complex Translations (2)

- Some boys sits on the left of some girl:

  $\exists x\ \exists y\ (Boy(x) \wedge Girl(y) \wedge LeftOf(x,y))$

- Every boy sits on the left of every girl:

  $\forall x\ \forall y\ ((Boy(x) \wedge Girl(y)) \rightarrow LeftOf(x,y))$

Complex Translations (3)

- In the previous example, we had more than one quantifier on the 'outside'. This is called **Prenex** form.
- However, Prenex is not necessarily required. The previous 2 examples could have been expressed as:

  $\exists x\ (Boy(x) \rightarrow \exists y\ (Girl(y) \wedge LeftOf(x,y))$

  $\forall x\ (Boy(x) \rightarrow \forall y\ (Girl(y) \rightarrow LeftOf(x,y))$

Complex Translations (4)

- Using multiple quantifiers sometimes makes things slightly tricky. Consider:

  $\forall x \forall y\ ((Boy(x) \wedge Boy(y)) \rightarrow (LeftOf(x,y) \vee RightOf(x,y)))$

- This sentence may be read as "if x and y are boys, then either x is on the left of y or x is on the right of y".
- This makes the wrong assumption that x and y are distinct.

Complex Translations (5)

- To see why, consider:
  $\forall y\ ((Boy(Mark) \wedge Boy(y)) \rightarrow$
  $(LeftOf(Mark,y) \vee RightOf(Mark,y)))$

- But $\forall y$ includes Mark, so we can have:
  $\forall y\ ((Boy(Mark) \wedge Boy(Mark)) \rightarrow$
  $(LeftOf(Mark,Mark) \vee RightOf(Mark,Mark)))$

- It is clear that Mark cannot be left or right of himself.

Complex Translations (6)

- The wrong way was:

  $\forall x \forall y\ ((Boy(x) \wedge Boy(y)) \rightarrow (LeftOf(x,y) \vee RightOf(x,y)))$

- The correct way of making that statement is:

  $\forall x \forall y\ ((Boy(x) \wedge Boy(y) \wedge \mathbf{x} \neq \mathbf{y}) \rightarrow (LeftOf(x,y) \vee RightOf(x,y)))$

Complex Translations (7)

- Universal and existential quantifiers may be mixed. Consider "every boy is on the left of a girl":

  $\forall$**x (Boy(x)** $\rightarrow$ $\exists$**y (Girl(y)** $\wedge$ **LeftOf(x,y)))**

- This can be expressed in Prenex form as:

  $\forall$**x**$\exists$**y (Boy(x)** $\rightarrow$ **(Girl(y)** $\wedge$ **LeftOf(x,y)))**

Complex Translations (8)

- When mixing quantifiers, their order is important. Consider the following two sentences:

  $\forall$**x**$\exists$**y Likes(x,y)**

  $\exists$**y**$\forall$**x Likes(x,y)**

- The first states that "everyone ($\forall$**x)** likes someone ($\exists$**y)**".
- The second states that "there is someone ($\exists$**y)** who everyone ($\forall$**x)** likes".

Disjunctions and Conjunctions

- A **Literal** is an atomic sentence or its negation (e.g. P, $\neg$P)
- A **disjunction** is a sequence of literals connected by $\vee$'s

  $P \vee Q \vee R$
- A **conjunction** is a sequence of literals connected by $\wedge$'s

  $P \wedge Q \wedge R$

Normal Forms

- A sentence is in **Negative Normal Form (NNF)** if it is built from literals using only $\vee$'s and $\wedge$'s.

  $\mathbf{P \wedge (\neg Q \vee R)}$
- A sentence is in **Disjunctive Normal Form (DNF)** if it is a disjunction of one or more conjunctions (of literals)

  $\mathbf{(P \wedge Q) \vee (R \wedge S)}$
- A sentence is in **Conjunction Normal Form (CNF)** if it is a conjunction of one or more disjunctions (of literals)

  $\mathbf{(P \vee Q) \wedge (R \vee S)}$

Conversion between Normal Forms (1)

- Recall that in algebra it is useful to use distributive laws to convert expressions (no matter how complex) involving +'s and ×'s into ones that are sums of products. E.g.:

$$
\begin{aligned}
(a+b)(c+d) &= (a + b)c + (a+b)d \\
&= ac + bc + (a+b)d \\
&= ac + bc + ad + bd
\end{aligned}
$$

Conversion between Normal Forms (2)

- In exactly the same way, the distributive laws of logic (see previous slides) allow us to convert a sentence in NNF to DNF. E.g.:

$$
\begin{aligned}
(a \vee b) \wedge (c \vee d) &= (a \vee b) \wedge c \vee (a \vee b) \wedge d \\
&= a \wedge c \vee b \wedge c \vee (a \vee b) \wedge d \\
&= a \wedge c \vee b \wedge c \vee a \wedge d \vee b \wedge d
\end{aligned}
$$

Conversion between Normal Forms (3)

- Likewise we can use distribution to convert NNF to CNF. E.g.

$$
\begin{aligned}
(a \wedge b) \vee (c \wedge d) &= ((a \wedge b) \vee c) \wedge ((a \wedge b) \vee d) \\
&= (a \vee c) \wedge (b \vee c) \wedge ((a \wedge b) \vee d) \\
&= \\
&(a \vee c) \wedge (b \vee c) \wedge (a \vee d) \wedge (b \vee d)
\end{aligned}
$$

Conversion between Normal Forms (4)

- Sometimes DeMorgan's laws should be used to convert basic formulae (WFF) into NNF. E.g.

  $\neg(P \wedge Q) \equiv (\neg P \vee \neg Q)$
- Some sentences are WFF, NNF, DNF and CNF simultaneously:

  $(P \wedge Q \wedge R)$

Proof Systems

- The technique we used so far for establishing the validity of arguments was based on the evaluation of their corresponding truth tables.
- The a problem starts appearing when the number of variables grows large – the truth table starts becoming too complex to work out.
- We will consider the use of **Formal deductive Systems** to establish the validity of a sentence or argument.

Formal Deductive Systems

- In formal systems we use an algebraic technique.
- A Formal Deductive System (FDS) requires:
  - A formal language.
  - A set of axioms. These are basic formulae from which theorems are derived. Axioms don't need to be proven. They are fundamental stated facts.

    (e.g. $b \times 1 = b$)
  - Inference rules. These are rules that allow us to derive new sentences from existing ones.

More on Inference Rules

- Using Inference Rules we can derive new rules from existing ones.
- A common rule is called **Modus Ponens**:
  - If we know a fact **P**, and also know **P→Q**, then we know that **Q** is provable too.
- Another one is **Modus Tollens**:
  - If we know **Q** is false, and **P→Q**, then we know that **P** must be false as well.
  - **If there is fire here, then there is oxygen here.**
    There is no oxygen here.
    Therefore, there is no fire here.

Proofs vs. Theorems

- A proof is a sequence of formulae such that:
  - It is an axiom.
  - It is derivable from earlier formulae in the sequence using inference rules.
- A proof whose last element is the formulae **F**, is called the **Proof of F**.
- A formula **F** is a theorem in a system if there is a **Proof of F** in the system, denoted by σ^F.
- σ is the system.

Deduction

- A proof is derived from axioms.
- A deduction allows the possibility of a number of assumptions.
- So, a deduction is a sequence of formulae such that:
  - It is an axiom.
  - **A number of assumptions/hypothesis.**
  - It is derivable from earlier formulae in the sequence using inference rules.

System L (1)

- System L is a FDS.
- It requires the following objects:
  - An alphabet.
  - A set of WFFs.
  - A set of axioms (possibly empty).
  - A set of inference rules.
- The **alphabet** consists of: the set of propositional variables, punctuation symbols, and logical operators.

System L (2)

- If **P** is a **WFF**, it must conform to the following rules (Note that WFFs are written in uppercase):
  - **P** is a propositional variable, or
  - **P** is of the form **¬Q**, where **Q** is a **WFF**, or
  - **P** is of the form **(Q→R)** where **Q** and **R** are WFFs.
- In System L, we have only one rule – Modus Ponens.
- Axiom Schemes:
  - **A1)** **(U→(V→U)**
  - **A2)** **((U→(V→W))→((U→V)→(U→W))**
  - **A3)** **(¬U→¬V)→(V→U)**

Deductions in L

- Given the hypothesis:

(1) (p→(q→r))→(~~p→~(q→r))

(2) p→(q→r)

(3) q→r

- Show that: **¬p**

Deduction:

(a) **~~p→~(q→r)**

by MP on (1) (2)

(b) **(~~p→~(q→r))→((q→r)→~p)**

by A3 ~p/U, (q→r)/V

(c) **(q→r)→~p**

by MP on (a) (b)

(d) **~p**

by MP on (3) (c)

Horn Sentences (1)

- In previous slides, we have seen how to convert any sentence without quantifiers into CNF.
- A literal can be **positive** or **negative** depending on whether it is an atomic sentence or the negation of it (P, $\neg$P).
- A Horn Sentence (names after Alfred Horn) is a sentence in CNF where every disjunction of literals contains at most one positive literal.

Horn Sentences (2)

- None of the following sentences is a Horn sentence:
  - $\neg$Home(Claire) $\wedge$ (Home(Max) $\vee$ Happy(Carl))
  - (Home(Claire) $\vee$ Home(Max) $\vee$ $\neg$Happy(Claire)) $\wedge$ $\neg$Happy(Carl)
- The first is not a Horn sentence because the second disjunction contains two positive literals.
- The second is not because the first disjunction contains two positive literals too.

Horn Sentences (3)

- All of the following are Horn sentences:
  - $\neg$Home(Claire) $\wedge$ ($\neg$Home(Max) $\vee$ Happy(Carl))
  - Home(Claire) $\wedge$ Home(Max) $\wedge$ $\neg$Home(Carl))
  - Home(Claire) $\vee$ $\neg$Home(Max) $\vee$ $\neg$Home(Carl))
- All conjunctions have at most one positive literal in the constituent disjunctions.

Horn Sentences (4)

- We have considered the truth table method to determine whether a sentence is satisfiable or not. The problem is that this technique is completely mechanical, prone to error and consumes a lot of resources (consider the truth table for 50 atomic sentences).
- For Horn sentences we can check for satifiability in one row. This method is known as the **satisfaction algorithm for Horn sentences**.

Satisfaction Algorithm (1)

- Consider the sentence **S:**
  `Home(Claire)` ∧ ¬`Home(Max)` ∧
  (`Home(Max)` ∨ ¬(`Home(Claire)`))
- Let's abbreviate `Home(Claire)` to **C** and to `Home(Max)` **M**.
- **Write the table**:

C	**M**	**C ∧ ¬M ∧ (M ∨ ¬C)**

Satisfaction Algorithm (2)

- **Find is some atomic sentences are themselves conjuncts in S and write True beneath it**. C is atomic and a conjunct.

C	**M**	**C ∧ ¬M ∧ (M ∨ ¬C)**
T		

- **Fill as much as possible in the RHS**.

C	**M**	**C ∧ ¬M ∧ (M ∨ ¬C)**
T		**T** **F**

Satisfaction Algorithm (3)

- If we look at the last conjunct we see that if this must resolve to **T** then **M** must be **T**.

C	M	C ∧ ¬M ∧ (M ∨ ¬C)
T	T	T T F

- But this will make the second conjunct becomes false, rendering the whole sentence unsatisfiable.

C	M	C ∧ ¬M ∧ (M ∨ ¬C)
T	T	T F T F

Satisfaction Algorithm in General

1. **S** is a Horn sentence.
2. Start out as if building a truth table but just write the header.
3. See is any atomic sentences are themselves conjuncts of S. Write **T** beneath them in the reference column.
4. Is some are true, fill in as much as possible of the RHS. For example if you wrote **T** under an atomic sentence **A**, you would write **F** under any ¬**A**.
5. Continue until there is no other way to proceed.
6. If you are forced to assign **F** to one of the conjuncts then the sentence is not satisfiable.

Resolution - Preliminaries (1)

- Using many different inference rules such as modus ponens and modus tollens can be overwhelming.
- Instead we can use a general purpose inference rule called **resolution**.
- This makes automatic theorem provers such as PROLOG practical to develop.
- Before resolution can be applied, the sentence must be in normal form (i.e. uses only $\wedge$, $\vee$, and possibly $\neg$).
- Note that $P \rightarrow Q \equiv \neg P \vee Q$

Resolution - Preliminaries (2)

- A literal is an atomic formula or its negation.
- A clause is a set of literals, e.g. {A, B, $\neg$C}
- The symbol $\square$ denotes the empty clause {}.
- A clause **C** is satisfied by a truth assignment **h** if at least on literal in the clause is true.
- For example consider the truth assignment {T,F} on the clause {Plays(Kris), Sleeps(Kris)}.
- The first item in the truth assignment says that Plays(Kris) is T, so the clause is satisfied.
- In other words for a Clause **C** $\neq \square$, **h** satisfies **C** iff the disjunction of **C** is true.

Resolution - Preliminaries (3)

- Consider the CNF statement:
  (Play(Kris) $\vee$ Study(Kris)) $\wedge$ ($\neg$Study(Kris) $\vee$ Sleep(Kris))
- We have two clauses:
  - {Play(Kris), Study(Kris)}
  - {$\neg$Study(Kris), Sleep(Kris)}
- Clearly all the clauses must be satisfiable for the whole statement to be.
- CNF statements can be written down in set form:
  {{Play(Kris), Study(Kris)}, {$\neg$Study(Kris), Sleep(Kris)}}

The Resolution Inference Rule

$$\frac{A \vee B \qquad \neg B \vee C}{A \vee C}$$

$$\frac{\text{Play(Kris)} \vee \text{Study(Kris)} \qquad \neg\text{Study(Kris)} \vee \text{Sleep(Kris)}}{\text{Play(Kris)} \vee \text{Sleep(Kris)}}$$

- This is simple enough to understand:
  - It is obvious that either Study(Kris) or $\neg$Study(Kris), must be false.
  - So either Play(Kris) or Sleep(Kris) must be true.

Resolvents

- Let **C1** and **C2** be two clauses.
- **D** is a resolvent of **C1** and **C2** iff
  - A literal **L** exists in **C1**
  - **¬L** exists in **C2**
  - **D** is the merging of **{C1-L}** and **{C2-¬L}**
- For example:
  - **C1={A,¬B}, C2={A,B}, D = {A}**
  - **C1={A,B}, C2={¬A,¬B}, D={A,¬A} and {B,¬B}**

Resolution Proof by Refutation

- Suppose that the current knowledge base (KB) is converted into clause form and we are trying to prove a theorem:

  1) Add the negation of the theorem we are trying to prove in the set of clauses of the KB.

  2) Search for two resolvable clauses in the KB
  - If not found: THEOREM TO BE PROVEN IS FALSE
  - If found: resolve and add the result to the KB

  3) If a contradiction is reached then the theorem is true else goto **(2)**

Proving Modus Ponens

- In Modus Ponens we have:

$$\frac{\begin{array}{c} P \\ P \rightarrow Q \end{array}}{Q}$$

1. P is a unit clause {P}
2. Convert $P \rightarrow Q$ to $\neg P \vee Q$ to get $\{\neg P, Q\}$.
3. Add the negation of the theorem $\{\neg Q\}$
4. Resolvent of (1), (2) is {Q}
5. We have a contradiction {Q} and $\{\neg Q\}$
6. Proven.

Resolution Example (1)

Socrates is a man.
<u>All men are mortal.</u>
Socrates is a frog.

- Man(Socrates)
- Man(x) $\rightarrow$ Mortal(x)
- Frog(Socrates)
- Abbreviate:
  - M
  - $M \rightarrow T$
  - F

Resolution Example (2)

- Clauses:
  - $\{M\}$
  - $\{\neg M, T\}$ remember that $(M \rightarrow T) \equiv (\neg M \vee T)$
  - Add the negation of the conclusion: $\{\neg F\}$
- Apply resolution:
  - $\{M\} + \{\neg M, T\} = \{T\}$
- Nothing left to resolve and there is no contradiction in the conclusion – **The theorem is false.**