

# Propositional Logic

## Review Class

1

Adapted from slides by  
Tim Finin and  
Marie desJardins.

Some material adopted from notes  
by Andreas Geyer-Schulz  
and Chuck Dyer

# Propositional logic

- **Logical constants:** true, false
- **Propositional symbols:** P, Q, S, ... (**atomic sentences**)
- Wrapping **parentheses:** ( ... )
- Sentences are combined by **connectives**:
  - $\wedge$  ...and [conjunction]
  - $\vee$  ...or [disjunction]
  - $\Rightarrow$  ...implies [implication / conditional]
  - $\Leftrightarrow$  ..is equivalent [biconditional]
  - $\neg$  ...not [negation]
- **Literal:** atomic sentence or negated atomic sentence

# Examples of PL sentences

- P means “It is hot.”
- Q means “It is humid.”
- R means “It is raining.”
- $(P \wedge Q) \rightarrow R$   
“If it is hot and humid, then it is raining”
- $Q \rightarrow P$   
“If it is humid, then it is hot”
- A better way:  
Hot = “It is hot”  
Humid = “It is humid”  
Raining = “It is raining”

# Propositional logic (PL)

- A simple language useful for showing key ideas and definitions
- User defines a set of propositional symbols, like P and Q.
- User defines the **semantics** of each propositional symbol:
  - P means “It is hot”
  - Q means “It is humid”
  - R means “It is raining”
- A sentence (well formed formula) is defined as follows:
  - A symbol is a sentence
  - If S is a sentence, then  $\neg S$  is a sentence
  - If S is a sentence, then (S) is a sentence
  - If S and T are sentences, then  $(S \vee T)$ ,  $(S \wedge T)$ ,  $(S \rightarrow T)$ , and  $(S \leftrightarrow T)$  are sentences
  - A sentence results from a finite number of applications of the above rules

# A BNF grammar of sentences in propositional logic

`S := <Sentence> ;`

`<Sentence> := <AtomicSentence> | <ComplexSentence> ;`

`<AtomicSentence> := "TRUE" | "FALSE" |  
"P" | "Q" | "S" ;`

`<ComplexSentence> := "(" <Sentence> ")" |  
<Sentence> <Connective> <Sentence> |  
"NOT" <Sentence> ;`

`<Connective> := "NOT" | "AND" | "OR" | "IMPLIES" |  
"EQUIVALENT" ;`

# Some terms

- The meaning or **semantics** of a sentence determines its **interpretation**.
- Given the truth values of all symbols in a sentence, it can be “evaluated” to determine its **truth value** (True or False).
- A **model** for a KB is a “possible world” (assignment of truth values to propositional symbols) in which each sentence in the KB is True.

# More terms

- A **valid sentence** or **tautology** is a sentence that is True under all interpretations, no matter what the world is actually like or how the semantics are defined. Example: “It’s raining or it’s not raining.”
- An **inconsistent sentence** or **contradiction** is a sentence that is False under all interpretations. The world is never like what it describes, as in “It’s raining and it’s not raining.”
- **P entails Q**, written  $P \models Q$ , means that whenever P is True, so is Q. In other words, all models of P are also models of Q.

# Truth tables

*And*

$p$	$q$	$p \cdot q$
$T$	$T$	$T$
$T$	$F$	$F$
$F$	$T$	$F$
$F$	$F$	$F$

*Or*

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

*If ... then*

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

*Not*

$p$	$\sim p$
$T$	$F$
$F$	$T$



A : crashed  
B : reboot  
C : A-> B : script is working

A -> B : C

crashed -> reboot	: script is working
crashed -> not reboot	: script is not working
not crashed -> reboot	: script is working
not crashed -> not reboot	: script is working

# Truth tables II

The five logical connectives:

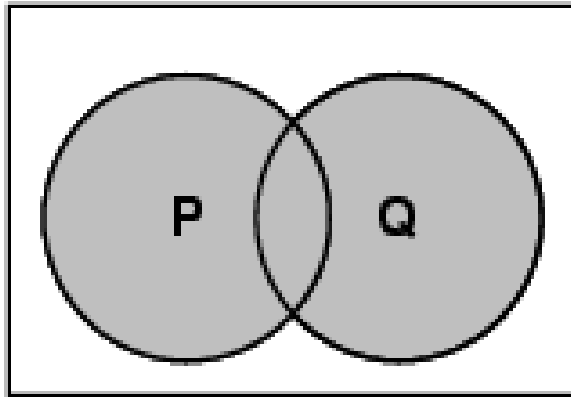
$P$	$Q$	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
<i>False</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>False</i>	<i>True</i>	<i>True</i>
<i>False</i>	<i>True</i>	<i>True</i>	<i>False</i>	<i>True</i>	<i>True</i>	<i>False</i>
<i>True</i>	<i>False</i>	<i>False</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>False</i>
<i>True</i>	<i>True</i>	<i>False</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>

A complex sentence:

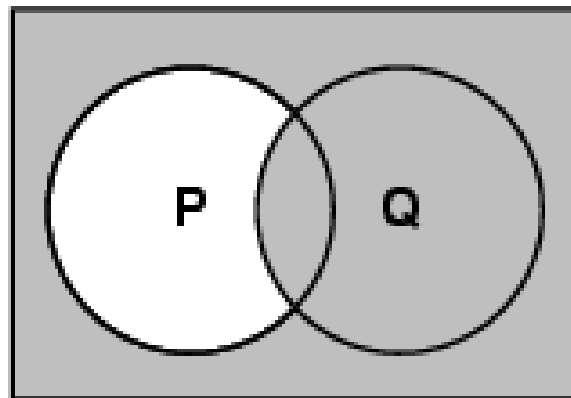
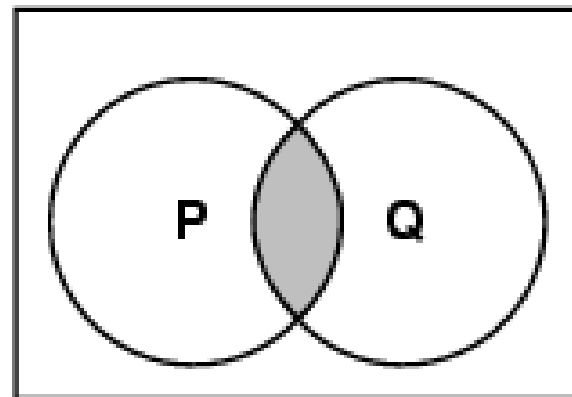
$P$	$H$	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
<i>False</i>	<i>False</i>	<i>False</i>	<i>False</i>	<i>True</i>
<i>False</i>	<i>True</i>	<i>True</i>	<i>False</i>	<i>True</i>
<i>True</i>	<i>False</i>	<i>True</i>	<i>True</i>	<i>True</i>
<i>True</i>	<i>True</i>	<i>True</i>	<i>False</i>	<i>True</i>

# Models of complex sentences

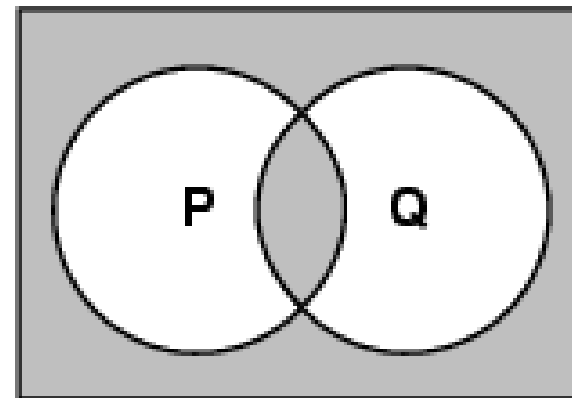
$P \vee Q$



$P \wedge Q$



$P \Rightarrow Q$



$P \Leftrightarrow Q$

# Inference rules

- **Logical inference** is used to create new sentences that logically follow from a given set of predicate calculus sentences (KB).
- An inference rule is **sound** if every sentence X produced by an inference rule operating on a KB logically follows from the KB. (That is, the inference rule does not create any contradictions)
- An inference rule is **complete** if it is able to produce every expression that logically follows from (is entailed by) the KB. (Note the analogy to complete search algorithms.)

# Sound rules of inference

- Here are some examples of sound rules of inference
  - A rule is sound if its conclusion is true whenever the premise is true*
- Each can be shown to be sound using a truth table

<u>RULE</u>	<u>PREMISE</u>	<u>CONCLUSION</u>
Modus Ponens	$A, A \rightarrow B$	$B$
And Introduction	$A, B$	$A \wedge B$
And Elimination	$A \wedge B$	$A$
Double Negation	$\neg\neg A$	$A$
Unit Resolution	$A \vee B, \neg B$	$A$
<b>Resolution</b>	<b><math>A \vee B, \neg B \vee C</math></b>	<b><math>A \vee C</math></b>

# Soundness of modus ponens

A	B	$A \rightarrow B$	OK?
True	True	True	✓
True	False	False	✓
False	True	True	✓
False	False	True	✓



# Soundness of the resolution inference rule

$\alpha$	$\beta$	$\gamma$	$\alpha \vee \beta$	$\neg\beta \vee \gamma$	$\alpha \vee \gamma$
<i>False</i>	<i>False</i>	<i>False</i>	<i>False</i>	<i>True</i>	<i>False</i>
<i>False</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>True</i>	<i>True</i>
<i>False</i>	<i>True</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>False</i>
<u><i>False</i></u>	<u><i>True</i></u>	<u><i>True</i></u>	<u><i>True</i></u>	<u><i>True</i></u>	<u><i>True</i></u>
<u><i>True</i></u>	<u><i>False</i></u>	<u><i>False</i></u>	<u><i>True</i></u>	<u><i>True</i></u>	<u><i>True</i></u>
<u><i>True</i></u>	<u><i>False</i></u>	<u><i>True</i></u>	<u><i>True</i></u>	<u><i>True</i></u>	<u><i>True</i></u>
<i>True</i>	<i>True</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>True</i>
<u><i>True</i></u>	<u><i>True</i></u>	<u><i>True</i></u>	<u><i>True</i></u>	<u><i>True</i></u>	<u><i>True</i></u>

# Proving things

- A **proof** is a sequence of sentences, where each sentence is either a premise or a sentence derived from earlier sentences in the proof by one of the rules of inference.
- The last sentence is the **theorem** (also called goal or query) that we want to prove.
- Example for the “weather problem” given above.

1 Humid	Premise	“It is humid”
2 Humid $\rightarrow$ Hot	Premise	“If it is humid, it is hot”
3 Hot	Modus Ponens(1,2)	“It is hot”
4 (Hot $\wedge$ Humid) $\rightarrow$ Rain	Premise	“If it’s hot & humid, it’s raining”
5 Hot $\wedge$ Humid	And Introduction(1,2)	“It is hot and humid”
6 Rain	Modus Ponens(4,5)	“It is raining”



# Horn sentences

- A **Horn sentence** or **Horn clause** has the form:

$$P1 \wedge P2 \wedge P3 \dots \wedge Pn \rightarrow Q$$

or alternatively

$$\neg P1 \vee \neg P2 \vee \neg P3 \dots \vee \neg Pn \vee Q$$

$$(P \rightarrow Q) = (\neg P \vee Q)$$

where Ps and Q are non-negated atoms

- To get a proof for Horn sentences, apply Modus Ponens repeatedly until nothing can be done
- We will use the Horn clause form later

# Entailment and derivation

## ■ Entailment: $KB \models Q$

- Q is entailed by KB (a set of premises or assumptions) if and only if there is no logically possible world in which Q is false while all the premises in KB are true.
- Or, stated positively, Q is entailed by KB if and only if the conclusion is true in every logically possible world in which all the premises in KB are true.

## ■ Derivation: $KB \vdash Q$

- We can derive Q from KB if there is a proof consisting of a sequence of valid inference steps starting from the premises in KB and resulting in Q

# Two important properties for inference

## Soundness: If $KB \vdash Q$ then $KB \models Q$

- If  $Q$  is derived from a set of sentences  $KB$  using a given set of rules of inference, then  $Q$  is entailed by  $KB$ .
- Hence, inference produces only real entailments, or any sentence that follows deductively from the premises is valid.

## Completeness: If $KB \models Q$ then $KB \vdash Q$

- If  $Q$  is entailed by a set of sentences  $KB$ , then  $Q$  can be derived from  $KB$  using the rules of inference.
- Hence, inference produces all entailments, or all valid sentences can be proved from the premises.

# Propositional logic is a weak language

- Hard to identify “individuals” (e.g., Mary, 3)
- Can’t directly talk about properties of individuals or relations between individuals (e.g., “Bill is tall”)
- Generalizations, patterns, regularities can’t easily be represented (e.g., “all triangles have 3 sides”)
- First-Order Logic (abbreviated FOL or FOPC) is expressive enough to concisely represent this kind of information
  - FOL adds relations, variables, and quantifiers, e.g.,
    - “*Every elephant is gray*”:  $\forall x (\text{elephant}(x) \rightarrow \text{gray}(x))$
    - “*There is a white alligator*”:  $\exists x (\text{alligator}(X) \wedge \text{white}(X))$

# Example

- Consider the problem of representing the following information:
  - Every person is mortal.
  - Confucius is a person.
  - Confucius is mortal.
- How can these sentences be represented so that we can infer the third sentence from the first two?

# Example II

- In PL we have to create propositional symbols to stand for all or part of each sentence. For example, we might have:  
P = “person”; Q = “mortal”; R = “Confucius”
- so the above 3 sentences are represented as:  
 $P \rightarrow Q$ ;  $R \rightarrow P$ ;  $R \rightarrow Q$
- Although the third sentence is entailed by the first two, we needed an explicit symbol, R, to represent an individual, Confucius, who is a member of the classes “person” and “mortal”
- To represent other individuals we must introduce separate symbols for each one, with some way to represent the fact that all individuals who are “people” are also “mortal”

# The “**Hunt the Wumpus**” agent

- Some atomic propositions:

S12 = There is a stench in cell (1,2)

B34 = There is a breeze in cell (3,4)

W22 = The Wumpus is in cell (2,2)

V11 = We have visited cell (1,1)

OK11 = Cell (1,1) is safe.

etc

- Some rules:

(R1)  $\neg S11 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W21$

(R2)  $\neg S21 \rightarrow \neg W11 \wedge \neg W21 \wedge \neg W22 \wedge \neg W31$

(R3)  $\neg S12 \rightarrow \neg W11 \wedge \neg W12 \wedge \neg W22 \wedge \neg W13$

(R4)  $S12 \rightarrow W13 \vee W12 \vee W22 \vee W11$

etc

- Note that the lack of variables requires us to give similar rules for each cell

# After the third move

- We can prove that the Wumpus is in (1,3) using the four rules given.
- See R&N section 7.5

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

**A** = Agent  
**B** = Breeze  
**G** = Glitter, Gold  
**OK** = Safe square  
**P** = Pit  
**S** = Stench  
**V** = Visited  
**W** = Wumpus



# Proving W13

- Apply MP with  $\neg S11$  and R1:  
 $\neg W11 \wedge \neg W12 \wedge \neg W21$
- Apply And-Elimination to this, yielding 3 sentences:  
 $\neg W11, \neg W12, \neg W21$
- Apply MP to  $\sim S21$  and R2, then apply And-elimination:  
 $\neg W22, \neg W21, \neg W31$
- Apply MP to S12 and R4 to obtain:  
 $W13 \vee W12 \vee W22 \vee W11$
- Apply Unit resolution on  $(W13 \vee W12 \vee W22 \vee W11)$  and  $\neg W11$ :  
 $W13 \vee W12 \vee W22$
- Apply Unit Resolution with  $(W13 \vee W12 \vee W22)$  and  $\neg W22$ :  
 $W13 \vee W12$
- Apply UR with  $(W13 \vee W12)$  and  $\neg W12$ :  
 $W13$
- QED

# Problems with the propositional Wumpus hunter

- Lack of variables prevents stating more general rules
  - We need a set of similar rules for each cell
- Change of the KB over time is difficult to represent
  - Standard technique is to index facts with the time when they're true
  - This means we have a separate KB for every time point

# Summary

- The process of deriving new sentences from old one is called **inference**.
  - **Sound** inference processes derives true conclusions given true premises
  - **Complete** inference processes derive all true conclusions from a set of premises
- A **valid sentence** is true in all worlds under all interpretations
- If an implication sentence can be shown to be valid, then—given its premise—its consequent can be derived
- Different logics make different **commitments** about what the world is made of and what kind of beliefs we can have regarding the facts
  - Logics are useful for the commitments they do not make because lack of commitment gives the knowledge base engineer more freedom
- **Propositional logic** commits only to the existence of facts that may or may not be the case in the world being represented
  - It has a simple syntax and simple semantics. It suffices to illustrate the process of inference
  - Propositional logic quickly becomes impractical, even for very small worlds