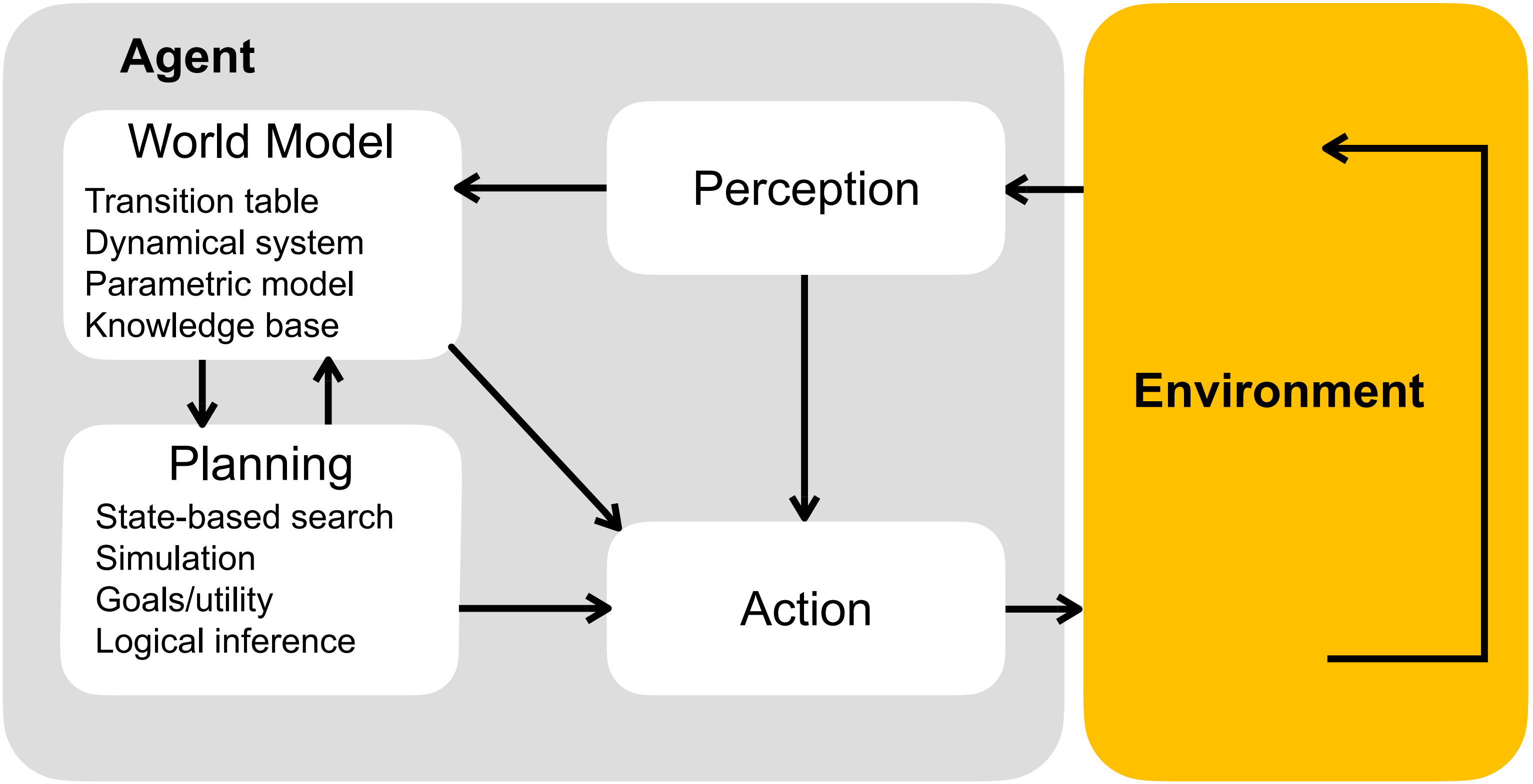
Knowledge Representation

LECTURE 5 - part I

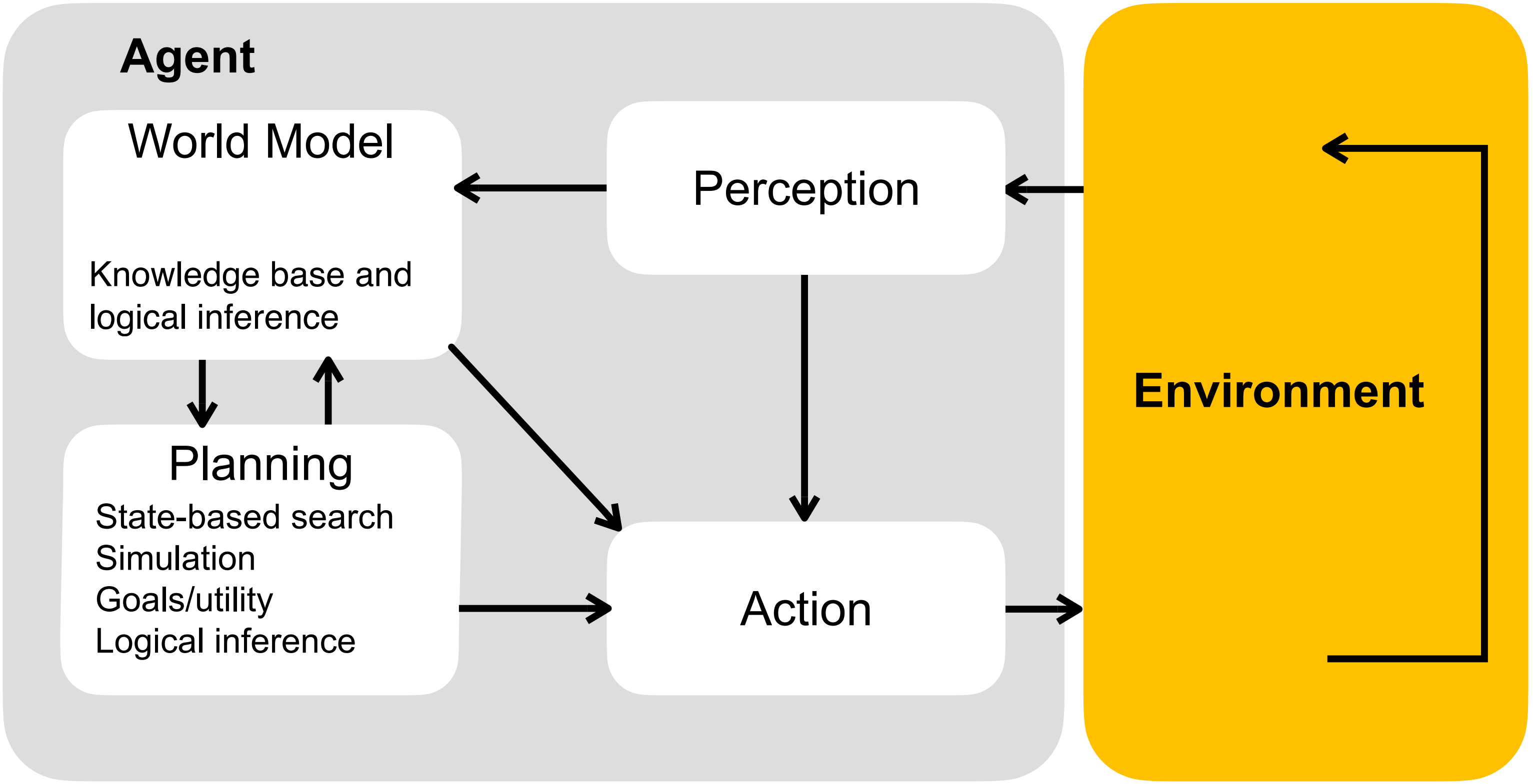
Knowledge Representation & Logics
Logical Arguments • Propositional Logic
Validity, Equivalence, Satisfiability, Entailment
Inference by Natural Deduction

UNSW SYDNEY

Models and Planning



Models and Planning



Knowledge Based Agent

The agent must be able to:

- represent states, actions, etc.
- incorporate new percepts
- update internal representations of the world
- deduce hidden properties of the world
- determine appropriate actions

The Knowledge Level

- **Knowledge Level Hypothesis.** There exists a distinct computer systems level, lying immediately above the symbol level, which is characterised by knowledge as the medium and the principle of rationality as the law of behaviour.

- **Principle of Rationality.** If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action.

- **Knowledge**. Whatever can be ascribed to an agent, such that its behaviour can be computed according to the principle of rationality.

"The Knowledge Level" (Newell, 1982)

Knowledge Representation

- Any agent can be described on different level
    - Knowledge level (knowledge ascribed to agent)
    - Logical level (algorithms for manipulating knowledge)
    - Implementation level (how algorithms are implemented)
- Knowledge Representation is concerned with expressing knowledge explicitly in a computer-tractable way (for use by an agent in reasoning) – not the same as Newell's view

- Reasoning attempts to take this knowledge and draw inferences (e.g. answer queries, determine facts that follow from the knowledge, decide what to do, etc.) – as part of the agent architecture

Knowledge Representation and Reasoning

- A knowledge-based agent has at its core a **knowledge** base
- A knowledge base is an explicit set of **sentences** about some domain expressed in a suitable formal representation language
- Sentences express facts (**true**) or non-facts (**false**)
- Fundamental Questions
  - How do we write down knowledge about a domain/problem?
  - How do we automate reasoning to deduce new facts or ensure consistency of a knowledge base?

Knowledge representation

- We are looking at the technology for knowledge-based agents: the syntax, semantics, and proof theory of propositional and first-order logic, and the implementation of agents that use these logics.

- We also need to address the question: What *content* to put into such an agent's knowledge base?
  - how to represent facts about the world.

本体论

Ontologies and Ontological Engineering

等级制度，集团

- A general **ontology** organises everything in the world into a hierarchy of categories.

期望 吓人的

- The prospect of representing *everything* in the world is daunting.
  - We won't actually write a complete description of everything—that would be far too much - but we will leave placeholders where new knowledge for any domain can fit in.
  - We will define what it means to be a physical object, and the details of different types of objects —robots, televisions, books, or whatever—can be filled in later.
- Similar to OO programming framework

Reasoning system for categories

- Categories are the primary building blocks of large-scale knowledge representation schemes.

- There are two closely related families of systems:

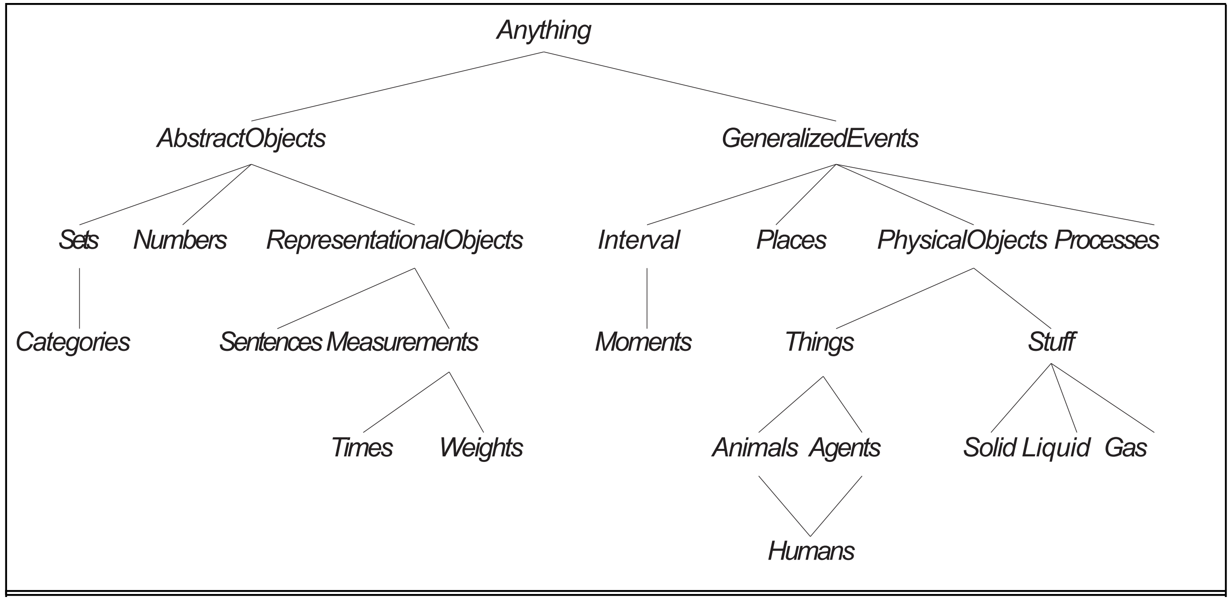
  - **semantic networks** provide graphical aids for visualising a knowledge base and efficient algorithms for inferring properties of an object on the basis of its category membership; and

  - **description logics** provide a formal language for constructing and combining category definitions and efficient algorithms for deciding subset and superset relationships between categories.

Semantic Networks

- Semantic Network consists of
  - Facts, Objects, Attributes and Relationships
  - Relationships exist among instances of objects and classes of objects.
- Attributes and relationships can be represented as a network, known as an **associative network** or **semantic network**
- We can build a **model** – a **semantic network** representation of our example

Ontologies - Example



Each link indicates that the lower concept is a specialisation of the upper one.
Specialisations are not necessarily disjoint

- a human is both an animal and an agent

Knowledge Bases

- A knowledge base is a set of sentences in a formal language.
- Declarative approach to building an agent:
  - Tell the system what it needs to know, then it can ask itself what it needs to do
  - Answers should follow from the knowledge based.
- How do you formally specify how to answer questions?

Knowledge and Semantic Networks

- Facts may be static, in which case they can be written into the knowledge base.
  - Static facts need not be permanent, but they change sufficiency infrequently that changes can be accommodated by updating the knowledge base when necessary.

短暂的

- Facts may be transient - apply at a specific instance only or for a single run of the system

Knowledge and Semantic Networks

- One of the most important aspects of semantic networks is their ability to represent default values for categories.
- The **knowledge base** may contain *defaults* that can be used as facts in the absence of transient facts.

Example – A simple set of statements

- My car is a car
- A car is a vehicle
- A car has four wheels
- A car’s speed is 0 mph
- My car is red
- My car is in my garage
- My garage is a garage
- A garage is a building
- My garage is made from brick
- My car is in the High Street
- The High Street is a street
- A street is a road

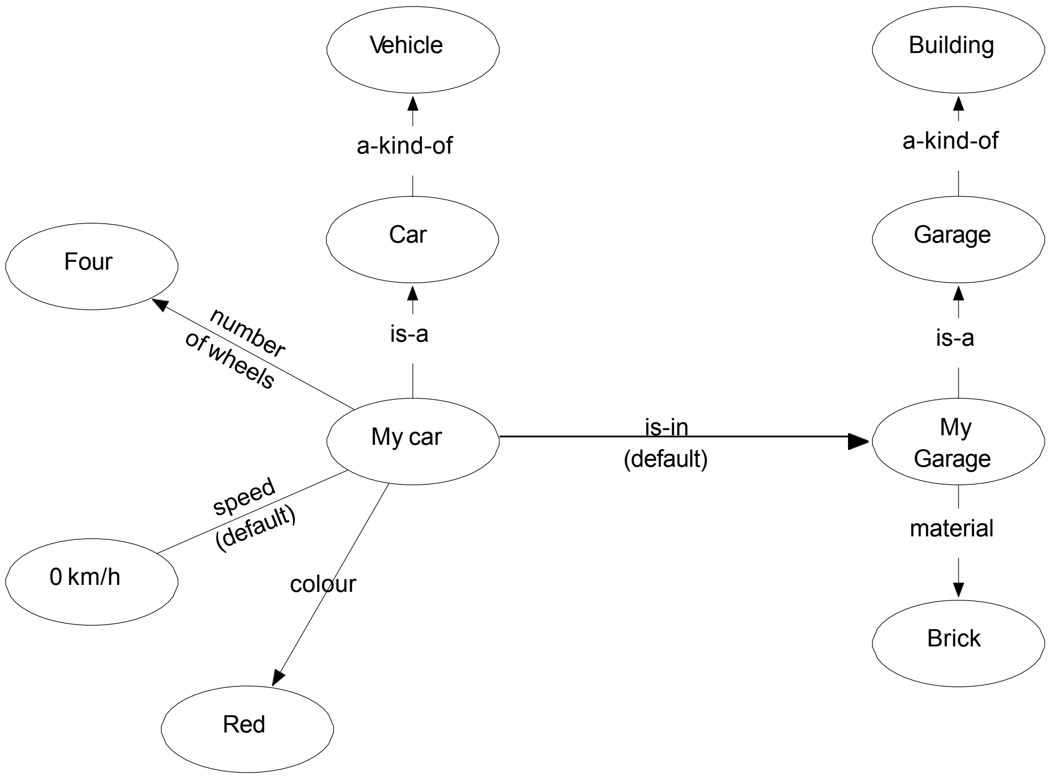
Example – facts, objects and relations

- My car is a **car**
- A car is a **vehicle**
- A car has four wheels
- A car’s speed is 0 mph
- My car is red
- My car is in my garage
- My garage is a **garage**
- A garage is a **building**
- My garage is made from brick
- My car is in the High Street
- The High Street is a **street**
- A street is a **road**

Example – facts, objects and relations

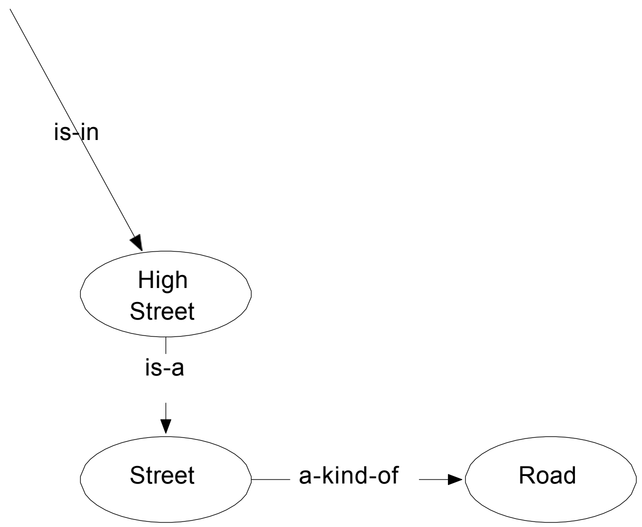
- My car is a **car** (static relationship)
- A car is a **vehicle** (static relationship)
- A car has four wheels (static attribute)
- A car's speed is 0 mph (default attribute)
- My car is red (static attribute)
- My car is in my garage (default relationship)
- My garage is a **garage** (static relationship)
- A garage is a **building** (static relationship)
- My garage is made from brick (static attribute)
- My car is in the High Street (transient relationship)
- The High Street is a **street** (static relationship)
- A street is a **road** (static relationship)

A semantic network (with a default)

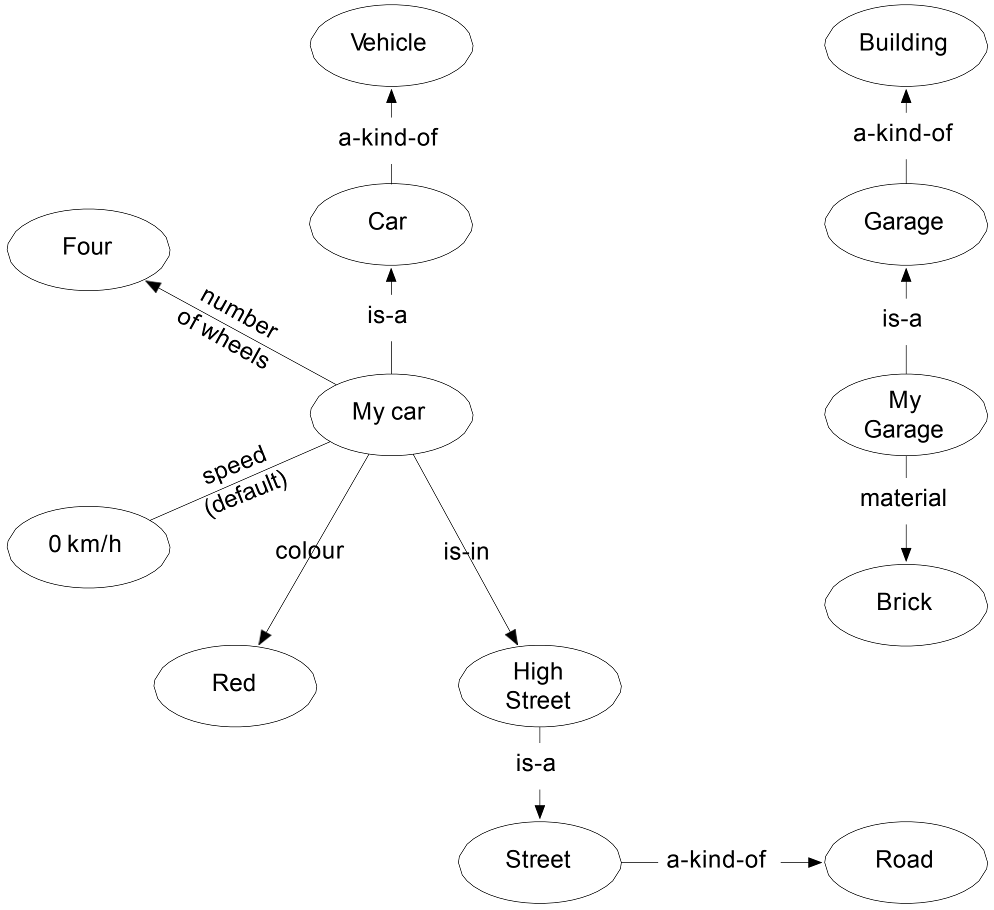


New fact

My car is in the High Street
The High Street is a street
A street is a road



A semantic network with an overridden default



Why Formal Languages – not English, or other natural language?

歧义

- Natural languages exhibit **ambiguity**
- "The fisherman went to the bank" (lexical)
- "The boy saw a girl with a telescope" (structural)
- "The table won't fit through the doorway because it is too [wide/narrow]"
- (co-reference)
- Ambiguity makes it difficult to interpret meaning of phrases/sentences
    - But also makes inference harder to define and compute

句法上

- Symbolic logic is a syntactically **unambiguous** language (originally developed in an attempt to formalise mathematical reasoning)

Syntax vs Semantics

句法vs语义

Syntax - legal sentences in knowledge representation language (e.g. in the language of arithmetic expressions $x < 4$)
Semantics - meaning of sentences.

- Refers to a sentence's relationship to the "real world" or to some model of the world.
- Semantic properties of sentences include truth and falsity (e.g. $x < 4$ is true for $x = 3$ and false when $x = 5$).
- Semantic properties of names and descriptions include referents.
- The meaning of a sentence is not intrinsic 固有的 to that sentence.
  - An interpretation is required to determine sentence meanings.
  - Interpretations 理解 are agreed amongst a linguistic community.

Propositions

- Propositions are entities (facts or non-facts) that can be **true** or **false**
- Use ordinary declarative sentences (not questions)
  - "The sky is blue" - the sky is blue (here and now).
- Is this proposition true?
- Examples
- "Socrates is bald" (assumes 'Socrates', 'bald' are well defined)
- "The car is red" (requires 'the car' to be identified)
- "Socrates is bald and the car is red" (complex proposition)
- In Propositional Logic, use single letters to represent propositions, a **scheme of abbreviation**, e.g. P: Socrates is bald
- Reasoning is independent of propositional substructure

Logical Arguments

An **argument** relates a set of premises to a conclusion
- valid if the conclusion necessarily follows from the premises 前提

All humans have 2 eyes
Jane is a human
Therefore Jane has 2 eyes

All humans have 4 eyes
Jane is a human
Therefore Jane has 4 eyes

- Both are (logically) correct **valid** arguments
- Which statements are true/false?

Logical Arguments

An **argument** relates a set of premises to a conclusion
- valid if the conclusion necessarily follows from the premises

All humans have 2 eyes
Jane has 2 eyes
Therefore Jane is human

No human has 4 eyes
Jane has 2 eyes
Therefore Jane is not human

- Both are (logically) incorrect **invalid** arguments
- Which statements are true/false?

Propositional Logic

- Use letters to stand for “basic” propositions; combine them into more complex sentences using operators for **not, and, or, implies, iff**
- Propositional **connectives**:

$\neg$	*negation*	$\neg P$	*“not P”*
$\wedge$	*conjunction*	$P \wedge Q$	*“P and Q”*
$\vee$	*disjunction*	$P \vee Q$	*“P or Q”*
$\rightarrow$	*implication*	$P \rightarrow Q$	*“If P then Q”*
$\leftrightarrow$	*bi-implication*	$P \leftrightarrow Q$	*“P if and only if Q”*

From English to Propositional Logic

- “It is not the case that the sky is blue”: $\neg B$
- (alternatively “the sky is not blue”)
- “The sky is blue and the grass is green”: $B \wedge G$
- “Either the sky is blue or the grass is green”: $B \vee G$
- “If the sky is blue, then the grass is not green”: $B \rightarrow \neg G$
- “The sky is blue if and only if the grass is green”: $B \leftrightarrow G$
- “If the sky is blue, then if the grass is not green, the plants will not grow”: $B \rightarrow (\neg G \rightarrow \neg P)$

Improving Readability

- $(P \rightarrow (Q \rightarrow (\neg(R))))$ vs $P \rightarrow (Q \rightarrow \neg R)$
- Rules for omitting brackets 删除括号规则
    - Omit brackets where possible (except maybe last example below!)
    - Precedence from highest to lowest is: ¬, ∧, ∨, → , ↔
    - All binary operators are left associative
- – so $P \rightarrow Q \rightarrow R$ abbreviates $(P \rightarrow Q) \rightarrow R$
- Questions
    - Is $(P \vee Q) \vee R$ (always) the same as $P \vee (Q \vee R)$?
    - Is $(P \rightarrow Q) \rightarrow R$ (always) the same as $P \rightarrow (Q \rightarrow R)$?
- NO!

Truth Table Semantics

- The semantics of the connectives can be given by **truth tables**

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
True	*True*	*False*	*True*	*True*	*True*	*True*
True	*False*	*False*	*False*	*True*	*False*	*False*
False	*True*	*True*	*False*	*True*	*True*	*False*
False	*False*	*True*	*False*	*False*	*True*	*True*

- One row for each possible assignment of True/False to variables
- Important: P and Q are any sentences, including complex sentences

Example – Complex Sentence

R	S	$\neg R$	$R \wedge S$	$\neg R \vee S$	$(R \wedge S) \rightarrow (\neg R \vee S)$
True	True	False	True	True	True
True	False	False	False	False	True
False	True	True	False	True	True
False	False	True	False	True	True

Thus $(R \wedge S) \rightarrow (\neg R \vee S)$ is a **tautology**

Definitions

- A sentence is valid if it is True under all possible assignments of True/False to its variables (e.g. $P \vee \neg P$)
- A tautology is a valid sentence

- Two sentences are equivalent if they have the same truth table, e.g. $P \wedge Q$ and $Q \wedge P$
  - So P is equivalent to Q if and only if $P \leftrightarrow Q$ is valid
- A sentence is satisfiable if there is some assignment of True/False to its variables for which the sentence is True

- A sentence is unsatisfiable if it is not satisfiable (e.g. $P \wedge \neg P$)
  - Sentence is False for all assignments of True/False to its variables
  - So P is a tautology if and only if $\neg P$ is unsatisfiable

Material Implication

- $P \to Q$ evaluates to False only when P is True and Q is False
- $P \to Q$ is equivalent to $\neg P \vee Q$: material implication P->Q为假的情况只有P为真且Q为假 -P且Q为假的条件也只有这一个 因此二者等价
- English usage often suggests a causal connection between antecedent
- (P) and consequent (Q) – this is not reflected in the truth table
- Examples
  - $(P \wedge Q) \to \quad Q$ is a tautology for any Q
  - $P \to (P \vee Q)$ is a tautology for any Q
  - $(P \wedge \neg P) \to \quad Q$ is a tautology for any Q

Logical Equivalences – All Valid

Commutativity:	$p \wedge q \leftrightarrow q \wedge p$	$p \vee q \leftrightarrow q \vee p$
Associativity:	$p \wedge (q \wedge r) \leftrightarrow (p \wedge q) \wedge r$	$p \vee (q \vee r) \leftrightarrow (p \vee q) \vee r$
Distributivity:	$p \wedge (q \vee r) \leftrightarrow (p \wedge q) \vee (p \wedge r)$	$p \vee (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r)$
Implication:	$(p \rightarrow q) \leftrightarrow (\neg p \vee q)$	
Idempotent:	$p \wedge p \leftrightarrow p$	$p \vee p \leftrightarrow p$
Double negation:	$\neg\neg p \leftrightarrow p$	
Contradiction:	$p \wedge \neg p \leftrightarrow$ FALSE	
Excluded middle:		$p \vee \neg p \leftrightarrow$ TRUE
De Morgan:	$\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$	$\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$

Proof of Equivalence

Let $P \Leftrightarrow Q$ mean "P is equivalent to Q" ($P \Leftrightarrow Q$ is not a formula)

Then $P \wedge (Q \rightarrow R) \Leftrightarrow \neg(P \rightarrow Q) \vee (P \wedge R)$

$$
\begin{aligned}
P \wedge (Q \rightarrow R) &\Leftrightarrow (P \wedge \neg Q) \vee (P \wedge R) && \text{[Implication]} \\
&\Leftrightarrow (\neg\neg P \wedge \neg Q) \vee (P \wedge R) && \text{[Distributivity]} \\
&\Leftrightarrow P \wedge (\neg Q \vee R) && \text{[Double negation]} \\
&\Leftrightarrow \neg(\neg P \vee Q) \vee (P \wedge R) && \text{[De Morgan]} \\
&\Leftrightarrow \neg(P \rightarrow Q) \vee (P \wedge R) && \text{[Implication]}
\end{aligned}
$$

Assumes substitution: if $A \Leftrightarrow B$, replace A by B in any sub-formula

Assumes equivalence is transitive: if $A \Leftrightarrow B$ and $B \Leftrightarrow C$ then $A \Leftrightarrow C$

Interpretations and Models

- An **interpretation** is an assignment of values to all variables.
- A **model** is an interpretation that satisfies the constraints.
  - A model is a **possible world** in which a sentence (0r set of sentences) is true, e.g.
  - x + y = 4 in a world where x and y = 2
- Often we don't want to just find a model, but want to know what is true in all models.
- A proposition is statement that is true or false in each interpretation.

Entailment 蕴含

- Entailment means that one sentence follows logically from another sentence, or set of sentences (i.e. a knowledge base):

$$KB \models \alpha$$

- Knowledge base *KB* entails sentence α if and only if α is true in all models (possible worlds) where *KB* is true.

  e.g. the KB containing “the Moon is full” and “the tide is high” entails “Either the Moon is full or the tide is high”.

  e.g. $x + y = 4$ entails $4 = x + y$

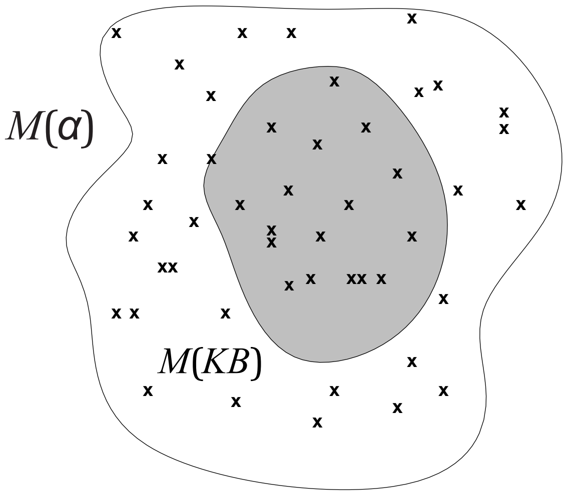
- Entailment is a relationship between sentences based on semantics.

Models

- For propositional logic, a model is **one** row of the truth table
- A model M **is a model of** a sentence if is True in M

Let $M(\alpha)$ be the set of all models of

Then $KB \models \alpha$ if and only if $M(KB) \subseteq M(\alpha)$



Entailment

- S **entails** P (S $\models$ P) if whenever all formulae in S are True, P is True
  - Semantic definition – concerns truth (not proof)

- Compute whether P by calculating a truth table for S and P
- Syntactic notion – concerns computation/proof
- Not always this easy to compute (how inefficient is this?)

- A tautology is a special case of entailment where S is the empty set
  - All rows of the truth table are True

Entailment

P	Q	$P \rightarrow Q$		Q
True	True	True		True
True	False	False		False
False	True	True		True
False	False	True		False

- $P, P \rightarrow Q\} \models Q$ since when both P and $P \rightarrow Q$ are True (row 1), Q is also True

- $P \rightarrow Q$ is calculated from P and Q using the truth table definition, and Q is used again to check the entailment

Entailment - $S \models P$

Each row is an interpretation of S.
Only the first row is a model of S.

$S = \{p \to q, q \to p, p \vee q\}$

$P = p \wedge q$

p	q	$p \to q$	$q \to p$	$p \vee q$	S	$p \wedge q$
T	T	T	T	T	T	T
T	F	F	T	T	F	F
F	T	T	F	T	F	F
F	F	T	T	F	F	F

Entailment - $S \models P$

$S = \{q \lor r, q \to {\sim}p, \neg(r \land p)\}$

$P = \neg p$

p	q	r	$q \lor r$	$q \to \neg p$	$\neg(r \land p)$	S	$\neg p$
T	T	T	T	F		F	
T	T	F	T	F		F	
T	F	T	T	T	F	F	
T	F	F	F			F	
F	T	T	T	T	T	T	T
F	T	F	T	T	T	T	T
F	F	T	T	T	T	T	T
F	F	F	F			F	

Simple Entailment

Write $P \models Q$ for $\{P\} \models Q$

$P \land Q \models P$

$P \models P \lor Q$

$P \models \neg\neg P$

$\{P, P \rightarrow Q\} \models Q$

$If\ P \models Q\ then \models P \rightarrow Q$

$\{P, P \rightarrow Q\} \models Q$

$If\ P \models Q\ then \models P \rightarrow Q$

$P \land Q \models Q$

$Q \models P \lor Q$

$P \models \neg\neg P$

Entailment - Tautology

R	S	$\neg R$	$R \wedge S$	$\neg R \vee S$	$(R \wedge S) \rightarrow (\neg R \vee S)$
True	True	False	True	True	True
True	False	False	False	False	True
False	True	True	False	True	True
False	False	True	False	True	True

Therefore $(R \wedge S) \rightarrow (\neg R \vee S)$

Natural Deduction Proofs (optional)

- Proofs are built by putting together smaller proofs, according to the rules.
- Start with a finite set of hypotheses $\{A, B, \ldots\}$
- Use a set of rules to combine proofs
- E.g, the way to read the and-introduction rule:

$$\frac{A \qquad B}{A \wedge B}$$

if you have a proof, P_1, of A from hypotheses, and you have a proof, P_2, of B from hypotheses, then you can put them together using this rule to obtain a proof of $A \wedge B$, which uses all the hypotheses in P_1 together with all the hypotheses of P_2

Natural Deduction Proofs - Rules of Inference (optional)

$\perp$ is the symbol for False

I = Introduction rule

E - Elimination rule

Conjunction

$$\dfrac{A \qquad B}{A \wedge B}\,\wedge\mathrm{I} \qquad \dfrac{A \wedge B}{A}\,\wedge\mathrm{E_l} \qquad \dfrac{A \wedge B}{B}\,\wedge\mathrm{E_r}$$

Truth and Falsity

$$\dfrac{\perp}{A}\,\perp\mathrm{E} \qquad \dfrac{}{\top}\,\top\mathrm{I}$$

Disjunction

$$\dfrac{A}{A \vee B}\,\vee\mathrm{I_l} \qquad \dfrac{B}{A \vee B}\,\vee\mathrm{I_r} \qquad \dfrac{A \vee B \qquad \begin{array}{c}\overline{A}^{\,1}\\ \vdots \\ C\end{array} \qquad \begin{array}{c}\overline{B}^{\,1}\\ \vdots \\ C\end{array}}{C}\,1\ \vee\mathrm{E}$$

Implication

$$\dfrac{\begin{array}{c}\overline{A}^{\,1}\\ \vdots \\ B\end{array}}{A \to B}\,1\ {\to}\mathrm{I} \qquad \dfrac{A \to B \qquad A}{B}\,{\to}\mathrm{E}$$

Negation

$$\dfrac{\begin{array}{c}\overline{A}^{\,1}\\ \vdots \\ \perp\end{array}}{\neg A}\,1\ \neg\mathrm{I} \qquad \dfrac{\neg A \qquad A}{\perp}\,\neg\mathrm{E}$$

Bi-implication

$$\dfrac{\begin{array}{c}\overline{A}^{\,1}\\ \vdots \\ B\end{array} \qquad \begin{array}{c}\overline{B}^{\,1}\\ \vdots \\ A\end{array}}{A \leftrightarrow B}\,1\ {\leftrightarrow}\mathrm{I} \qquad \dfrac{A \leftrightarrow B \qquad A}{B}\,{\leftrightarrow}\mathrm{E}_l \qquad \dfrac{A \leftrightarrow B \qquad B}{A}\,{\leftrightarrow}\mathrm{E}_r$$

Contradiction

$$\dfrac{\begin{array}{c}\overline{\neg A}^{\,1}\\ \vdots \\ \perp\end{array}}{A}\,1\ \mathrm{RAA}$$

Natural Deduction Proofs - Rules of Inference (optional)

Proof of $(A \wedge B) \wedge (B \wedge C)$ from hypotheses A, B and C:

$$\dfrac{\dfrac{A \qquad B}{A \wedge B} \qquad \dfrac{A \qquad C}{A \wedge C}}{(A \wedge B) \wedge (A \wedge C)}$$

- In diagram, the premises of each inference appear immediately above the conclusion.
- Makes it easy to look over a proof and check that it is correct.
- Each inference should be the result of instantiating the letters in one of the rules with particular formulas.
- Each step here used the *and-introduction* rule

Semantic Networks

- Semantic Networks are usually used for:
  - Representing data
  - Revealing structure (relations, proximity, relative importance)
  - Supporting conceptual edition
  - Supporting navigation

Semantic Networks - Reasoning

- Main reasoning mechanism is inheritance:
  - a category and its instances inherit the properties of the categories that contain them.
- Advantages of the semantic networks are:
  - Easy and natural to use
  - Meaning can be defined precisely
  - Inheritance is easy to implement and efficient

Conclusions

- Ambiguity of natural languages avoided with formal languages
- Enables formalisation of (truth preserving) entailment
- Propositional Logic: Simplest logic of truth and falsity
- Knowledge Based Systems: First-Order Logic
- Automated Reasoning: How to compute entailment (inference)
- Many many logics not studied in this course

References

- Poole & Mackworth, Artificial Intelligence: Foundations of Computational Agents, Chapter 5
- Russell & Norvig, *Artificial Intelligence: a Modern Approach*, Chapter 12.