## Logic & Knowledge representation

Perspective:

- We have been studying various forms of search:
  - Exploring alternatives (blind, heuristic, game)
  - Exploring state spaces (iterative improvement)
- These have been suitable for:
  - "Puzzles"
  - Optimization problems
  - Constraint satisfaction problems
- We separated two components in our solutions:
  - the search algorithm
  - an abstract problem description (i.e. states, actions, goals)
- We now start to tackle *reasoning* with *knowledge*
- This will still be a search problem!
- Questions:
  - How do we "reason?"
  - What kind of knowledge is needed?
  - How do we represent knowledge?
- Logic provides the foundation for these questions…



## Logic & Knowledge representation

- Natural languages:
  - expressive and concise
  - evolved for communication, not representation
- Formal languages:
  - unambiguous and independent of context
  - designed for precise descriptions of algorithms and computation states
- An ideal language for knowledge representation would combine the advantages of natural and formal languages.
- How do people represent knowledge? No one knows…
- Our approach to reasoning:

$$\begin{tikzcd} \boxed{\text{Sentences}} \arrow[r, Rightarrow, "\text{inference}"] & \boxed{\text{Sentences}} \arrow[d, Rightarrow, "\text{semantics}"] \\ \boxed{\text{Facts}} \arrow[u, Rightarrow, "\text{semantics}"] \arrow[r, Rightarrow, "\text{“follows”}"'] & \boxed{\text{Facts}} \end{tikzcd}$$



## Formal logic systems

- Components:
  - *syntax* — the grammar of the language, i.e. what symbols are allowed and how may they be assembled into a *sentence*?
  - *inference rules* — rules for manipulating sentences
  - *semantics* — what is the meaning of sentences?
- A formal logic system is a scheme for symbol manipulation!
- We will study algorithms for "mechanical reasoning" in formal logics.
- These procedure are applicable to *any* knowledge base written for that logic.
- Knowledge base — collection of sentences that are given (from which the system will make deductions)



## Propositional logic

- Grammar:

| | | |
|---|---|---|
| Sentence | $\rightarrow$ | AtomicSentence \| ComplexSentence |
| AtomicSentence | $\rightarrow$ | *True* \| *False* \| P \| Q \| … |
| ComplexSentence | $\rightarrow$ | ( Sentence ) \| Sentence Connective Sentence \| $\neg$ Sentence |
| Connective | $\rightarrow$ | $\wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow$ |

  Note that Nilsson uses $\supset$ instead of $\Rightarrow$.
- Operator precedence (highest to lowest):

$$\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$$

- Example sentences, also called *well-formed formulas* (WFFs):

$$(P \wedge Q) \Rightarrow \neg P$$
$$P \Rightarrow \neg P$$
$$(P \Rightarrow Q) \Rightarrow (\neg Q \Rightarrow \neg P)$$

## Rules of inference

There are many rules of inference that can be used with Propositional logic; here are the most common:

- modus ponens: $\dfrac{\text{P , P} \Rightarrow \text{Q}}{\text{Q}}$
- AND introduction: $\dfrac{\text{P}_1 \text{ , P}_2\text{, } \ldots \text{P}_n}{\text{P}_1 \wedge \text{P}_2 \ldots \wedge \text{P}_n}$
- AND elimination: $\dfrac{\text{P}_1 \wedge \text{P}_2 \ldots \wedge \text{P}_n}{\text{P}_i}$
- OR introduction: $\dfrac{\text{P}}{\text{P} \vee \text{Q}_1 \vee \text{P}_2 \ldots \vee \text{P}_n}$
- NOT elimination: $\dfrac{\neg\neg \text{ P}}{\text{P}}$



## Proofs

- A proof a sentence $w_n$ from a knowledge base $\Delta$ is a set of sentences $\{w_1, w_2, \ldots w_n\}$ where each $w_i$ is either a member of $\Delta$ or can be inferred from $\{w_1, \ldots w_{i-1}\}$.
- Example proof:

Given: $S$
$(S \vee P) \Rightarrow (Q \wedge R)$
$X$

Show: $Q \wedge X$

| Proof: | | |
|---|---|---|
| | $S$ | given |
| | $S \vee P$ | OR introduction |
| | $(S \vee P) \Rightarrow (Q \wedge R)$ | given |
| | $Q \wedge R$ | modus ponens |
| | $Q$ | AND elimination |
| | $X$ | given |
| | $Q \wedge X$ | AND introduction |



## Inference

- If $w_n$ can be proved from $\Delta$ with a set of inference rules $\mathcal{R}$, we will usually express this in one of the following ways:
  - $w_n$ can be derived from $\Delta$ (using $\mathcal{R}$)
  - $w_n$ can be inferred from $\Delta$ (using $\mathcal{R}$)
  - $\Delta \vdash_{\mathcal{R}} w_n$
- The set of inference rules $\mathcal{R}$ determines what can be inferred from a knowledge base.
- We hope that the inference rules make correct inferences…
- We hope that the inference rules can derive everything that is true…
- To address the last two points, we will work towards the concepts of *soundness* and *completeness*.



## Interpretations & truth tables

- The *ontological commitment* of a logic is what exists in the world.
- The *epistemological commitment* of a logic is what it believes about the world.
- Propositional logic:
  - the world consists of propositions (i.e. statements)
  - propositions are either *True* or *False*
- An *interpretation* is
  - (an association between atoms and propositions)
  - an assignment of values (*True* or *False*) to all atoms
- For $N$ atoms, there are $2^N$ interpretations.
- We use a *truth table* to enumerate all interpretations and determine the value of a WFF under each interpretation:

| P | Q | ¬P | P ∧ Q | P ∨ Q | P ⇒ Q | P ⇔ Q |
|---|---|---|---|---|---|---|
| F | F | T | F | F | T | T |
| F | T | T | F | T | T | F |
| T | F | F | F | T | F | F |
| T | T | F | T | T | T | T |

## Models & Satisfiability

- An interpretation *satisfies* a WFF if the WFF is *True* under that interpretation.
- An interpretation that satisfies a WFF is called a *model* of that WFF.
- Example for (P ∨ Q) ⇒ R:

| interpretation | P | Q | R | P ∨ Q | (P ∨ Q) ⇒ R |
|---|---|---|---|---|---|
| 1 | F | F | F | F | T |
| 2 | F | F | T | F | T |
| 3 | F | T | F | T | F |
| 4 | F | T | T | T | T |
| 5 | T | F | F | T | F |
| 6 | T | F | T | T | T |
| 7 | T | T | F | T | F |
| 8 | T | T | T | T | T |

Interpretations 1, 2, 4, 6, and 8 are models of (P ∨ Q) ⇒ R

- We can also speak of models of a set of WFFs.
- If there are no models for a WFF, it is *inconsistent* or *unsatisfiable*



## Entailment

- If all models of a knowledge base $\Delta$ are models of a WFF $w$, we can say $\Delta$ *entails* $w$ which is written:

$$\Delta \models w$$

This may also be expressed as "$w$ (logically) follows from $\Delta$" or "$w$ is a (logical) consequence of $\Delta$."

- Entailment (intuitively) is the concept of the "absolute" truth of a sentence $w$ given a set of facts $\Delta$, independent of any inference rules or procedure.
- Simple examples:

$$\begin{array}{ll} \{P\} & \models P \\ \{P, P \Rightarrow Q\} & \models Q \\ \{P \wedge Q\} & \models Q \\ \{\mathit{False}\} & \models X \end{array}$$



## Entailment example

$$\begin{aligned} \Delta &= \{P \Rightarrow Q, R, P \vee Q\} \\ w &= P \vee (Q \wedge R) \end{aligned}$$

| interpretation | P | Q | R | P ⇒ Q | P ∨ Q | P ∨ (Q ∧ R) |
|---|---|---|---|---|---|---|
| 1 | F | F | F | T | F | F |
| 2 | F | F | T | T | F | F |
| 3 | F | T | F | T | T | F |
| 4 | F | T | T | T | T | T |
| 5 | T | F | F | F | T | T |
| 6 | T | F | T | F | T | T |
| 7 | T | T | F | T | T | T |
| 8 | T | T | T | T | T | T |

| sentence(s) | models |
|---|---|
| P ⇒ Q | 1, 2, 3, 4, 7, 8 |
| R | 2, 4, 6, 8 |
| P ∨ Q | 3, 4, 5, 6, 7, 8 |
| $\Delta$ | 4, 8 |
| $w$ = P ∨ (Q ∧ R) | 4, 5, 6, 7, 8 |

Since all models of $\Delta$ (i.e. 4 and 8) are models of $w$,

$$\Delta \models w$$



## Soundness & Completeness

- We say that a set of inference rules $\mathcal{R}$ is *sound* if:

$$(\Delta \vdash_{\mathcal{R}} w) \Rightarrow (\Delta \models w)$$

- Soundness (intuitively) means that the set of inference rules is correct — the sentences that they infer are in fact true!
- We say that a set of inference rules $\mathcal{R}$ is *complete* if:

$$(\Delta \models w) \Rightarrow (\Delta \vdash_{\mathcal{R}} w)$$

- Completeness (intuitively) means that the set of inference rules can infer anything that is "true."

## Inference in Propositional logic

- The five inference rules given earlier:
  - are all sound
  - are not (even taken all together) complete for Propositional logic
- Modus ponens is complete on a restricted form of Propositional logic where:
  - all sentences are in Horn normal form
- *Resolution*, another inference rule, is *refutation complete* for Propositional logic!
  - For resolution, we usually put sentences in Conjunctive normal form (CNF) or Implicative normal form (INF)
- Normal forms are standard formats for WFFs that allow "mechanization" of inference.



## Horn Normal form

- Originally investigated by the logician Alfred Horn.
- A sentence in Horn normal form can be written as an implication where:
  - the antecedent is a conjunction of positive atoms
  - the consequent is a single positive atom
- For example:

$$P \wedge Q \wedge R \Rightarrow X$$
$$Y \Rightarrow Z$$

- Horn sentences can also be written as a disjunction of atoms, all but one of which is negative. For example:

$$\neg P \vee \neg Q \vee \neg R \vee X$$
$$\neg Y \vee Z$$

  These are simply a transformation of the implication form replacing $A \Rightarrow B$ with $\neg A \vee B$ and then applying de Morgan's law to the antecedent.
- Horn sentences cannot express anything in Propositional logic!



## Horn normal form

There are two (sort of) special cases of Horn sentences:

1. To represent a single positive literal, we can take the following sequence of steps:

$$G$$
$$\mathit{False} \vee G$$
$$\mathit{True} \Rightarrow G$$

   This is not seen commonly; instead the positive literal is written by itself.
2. To represent a single negated literal, we can take the following sequence of steps:

$$\neg H$$
$$\neg H \vee \mathit{False}$$
$$H \Rightarrow \mathit{False}$$

   A disjunction of negated literals can be represented in the same way by applying de Morgan's laws:

$$\neg H \vee \neg I \vee \neg J$$
$$(\neg H \vee \neg I \vee \neg J) \vee \mathit{False}$$
$$H \wedge I \wedge J \Rightarrow \mathit{False}$$



## Inference with Horn knowledge bases

- There are linear time algorithms to do inference on Horn databases (i.e. knowledge bases)!
- One basic algorithm is *Forward chaining*
- Generalized modus ponens:

$$\frac{P_1, P_2, \ldots P_N, \quad P_1 \wedge P_2 \wedge \cdots \wedge P_N \Rightarrow Q}{Q}$$

## Resolution in Propositional Logic

- The resolution inference rule:

$$\frac{P_1, P_2, \ldots P_N, \quad P_1 \wedge P_2 \wedge \cdots \wedge P_N \Rightarrow Q}{Q}$$



## Reducing to CNF in propositional logic

1. Eliminate implications: $(P \Rightarrow Q) \Leftrightarrow (\neg P \vee Q)$
2. Move $\neg$ inwards: de Morgan's laws
3. Use associative and distributive laws



## First order logic (or predicate calculus)

- Grammar:

| | | |
|---|---|---|
| Sentence | $\rightarrow$ | AtomicSentence \| |
| | | $\neg$ Sentence |
| | | ( Sentence ) \| |
| | | Sentence Connective Sentence \| |
| | | Quantifier Variable,... Sentence |
| AtomicSentence | $\rightarrow$ | Predicate( Term, ... ) \| |
| | | Term = Term |
| Term | $\rightarrow$ | Function(Term, ... ) \| |
| | | Constant \| |
| | | Variable |
| Connective | $\rightarrow$ | $\wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow$ |
| Qualifier | $\rightarrow$ | $\forall \mid \exists$ |

Here are some examples of constants, variables, predicates, and functionns:

| | | |
|---|---|---|
| Constant | $\rightarrow$ | A \| $X_1$ \| John \| ... |
| Variable | $\rightarrow$ | a \| x \| y \| ... |
| Predicate | $\rightarrow$ | Brother \| Parent \| ... |
| Function | $\rightarrow$ | Sister \| Mother \| ... |

- Operator precedence (highest to lowest):

$$\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow, \{\exists, \forall\}$$