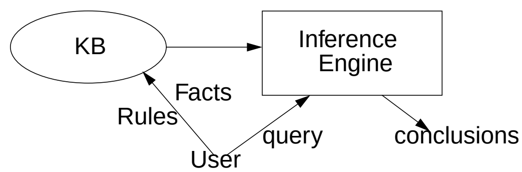
Representation, Reasoning, and Propositional Logic

Representation and Reasoning

- In order to determine appropriate actions to take to achieve goals, an intelligent system needs to compactly represent information about the world and draw conclusions based on general world knowledge and specific facts.

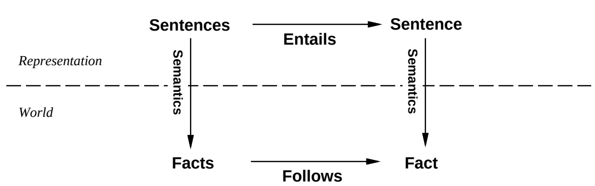
- Knowledge is represented by **sentences** in a particular knowledge representation language stored in a **knowledge base** (KB).

- A system draws conclusions from the KB to answer questions, solve problems, or suggest actions to perform to achieve goals.



Knowledge Representation Languages and Inference

- A KR language is specified by
  - **Syntax**: The atomic symbols used in the language and how they can be composed to formal legal sentences.
  - **Semantics**: What fact about the world is represented by a sentence in the language, which determines wether it is true or false.



- **Logical inference** (**deduction**) derives new sentences in the language from existing ones.

  Socrates is a man.
  All men are mortal.

  Socrates is mortal.

- Proper inference should only derive sound conclusions (ones that are true assuming the premises are true)

Propositional Logic Syntax

- Logical constants: True, False
- Propositional symbols: P, Q, etc. representing specific facts about the world
- If S is a sentence, then (S) is a sentence
- If S and R are sentences then so are:
  - $S \wedge R$: conjunction, S and R are conjuncts
  - $S \vee R$: disjunction, S and R are disjuncts
  - $S \Rightarrow R$: implication, S is a premise or antecedent, R is the conclusion or consequent, also known as a rule or if-then statement
  - $S \Leftrightarrow R$: equivalence (biconditional)
  - $\neg S$: negation
- Constants and symbols are **atomic**, other sentences are **complex**.
- A **literal** is an atomic sentence or its negation (P, $\neg$S)
- Precedence of operators:

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

Propositional Logic Semantics

- True and False indicate truth and falsity in the world
- A proposition denotes whatever fixed statement about the world you want which could be true or false.
- The semantics of complex sentences are derived from the semantics of their parts according to the following **truth table**.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	*False*	*True*	*False*	*False*	*True*	*True*
False	*True*	*True*	*False*	*True*	*True*	*False*
True	*False*	*False*	*False*	*True*	*False*	*False*
True	*True*	*False*	*True*	*True*	*True*	*True*

- Or is inclusive. Alternative is **exclusive or**: $P \oplus Q$
- Implication is material implication. If P is false $P \rightarrow Q$ is true by default. No causal connection implied.

Validity and Inference

- An **interpretation** is an assignment of True or False to each atomic propostion.
- A sentence that is true under any interpretation is **valid** (also called a **tautology** or **analytic sentence).**
- Validity can be checked by exhaustively exploring each possible interpretation in a truth table:

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
False	*False*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
True	*True*	*True*	*False*	*True*

- Inference can be performed by validity checking. If one has a set of sentences $\{S_1,...S_n\}$ defining one's background knowledge, and one want to know wether a conclusion C logically follows, construct the sentence:

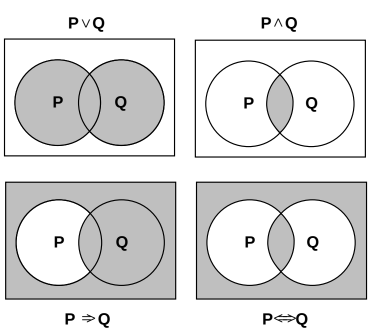
  $S1 \wedge S2 \wedge ... \wedge Sn \Rightarrow C$

  and check wether it is valid.

  How many rows do we need to check?

Models and Entailment

- Any interpretation in which a setence is true is called a **model** of the sentence.



- A sentence A **entails** a sentence B (A |= B) if every model of A is also a model of B. In this case, if A is true then B must be true.
- Correct logical inference is characterized by entailment, we want to be able to infer wether a statement S follows from a knowledge base:

  KB |= S
  or
  (KB $\rightarrow$ S) is valid

Rules of Inference

- As an alternative to checking all rows of a truth table, one can use **rules of inference** to draw conclusions.
- A sequence of inference rule applications that leads to a desired conclusion is called a **logical proof**.
- A |- B denotes that B can be derived by some inference procedure from the set of sentences A.
- Inference rules can be verified by the truth-table method and then used to construct sound proofs.
- Finding a proof is simply a search problem with the inference rules as operators and the conclusion as the goal.
- Logical inference can be more efficient than truth table construction.

Sample Rules of Inference

- Modus Ponens: $\{\alpha \Rightarrow \beta, \alpha\} \vdash \beta$
- And Elimination: $\{\alpha \wedge \beta\} \vdash \alpha$; $\{\alpha \wedge \beta\} \vdash \beta$
- And Introduction: $\{\alpha, B\} \vdash \alpha \wedge \beta$
- Or introduction: $\{\alpha\} \vdash \alpha \vee \beta$
- Double negation Elimination: $\{\neg\neg\alpha\} \vdash \alpha$
- Implication Elimination: $\{\alpha \Rightarrow \beta\} \vdash \neg\alpha \vee \beta$
- Unit resolution: $\{\alpha \vee \beta, \neg\beta\} \vdash \alpha$
- Resolution: $\{\alpha \vee \beta, \neg\beta \vee \gamma\} \vdash \alpha \vee \gamma$

Sample Proof

If John is not married he is a bachelor. ($\neg P \Rightarrow Q$)
John is not a bachelor. ($\neg Q$)
Therefore, he is married. (P)

$\neg P \Rightarrow Q$	

$\neg\neg P \vee Q$	Implication elimination

$P \vee Q$, $\neg Q$	Double negation elimination

P	Unit resolution

Could also check validity of:

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

Form of modus tollens reasoning

Satisfiability and Complexity of Inference

- A sentence is **satisfiable** if it is true under some interpretation (i.e. it has a model), otherwise the sentence is **unsatisfiable**.
- A sentence is valid if and only if its negation is unsatisfiable.
- Therefore, algorithms for either validity or satisfiability checking are useful for logical inference.
- If there are *n* propositional symbols in a sentence, then simple validity checking must enumerate 2^n rows (checking each row is only linear in length of the sentence to compute truth value).
- However, propositional satisfiability is the first problem to be proven NP-complete, and therefore there is assumed to be no polynomial-time algorithm.
- Therefore, sound and complete logical inference in propositional logic is intractable in general.
- But many problems can be solved very quickly.