Definite Clauses & Horn Clauses

- **Clause**: A disjunction of literals
  
  e.g. \( \neg R \lor \neg P \lor \neg Q \)

- **Definite Clause**: A clause with exactly one positive literal
  
  e.g. \( \neg R \lor Q \lor P \)

- **Horn Clause**: A clause with at most one positive literal
  
  e.g. \( \neg R \lor \neg Q \lor P \)

| \( \neg R \lor \neg Q \lor P \) | aka Goal |
| \( \neg R \lor \neg Q \) | aka Fact |
| \( P \) | aka Fact |

All definite clauses are Horn Clauses.
Definite & Horn Clauses – Why bother?

• Every definite clause can be written as an implication.

$$\neg R \lor \neg Q \lor P \equiv R \land Q \Rightarrow P$$

$$P \equiv \text{True} \Rightarrow P \quad \text{Fact}$$

• Inference with Horn Clauses can be done using Forward Chaining and Backward Chaining algorithms. This is the basis for Logic programming.

• Entailment with Horn Clauses can be done in linear time!

Agents with Knowledge and Reasoning

• Knowledge Representation & Reasoning (KRR) Systems

**KnowledgeBase**

"Knowledge" + Inference Engine

• **Knowledge** – set of sentences that describe facts about the world (or domain)

• **Inferences** – procedures/rules that operate on facts to infer new facts
Forward Chaining Inference

KnowledgeBase

- Known Facts (positive literals)
- Inference Engine (Forward Chaining)

Tell/Sense/Ask

Response (Answer/Do Something)

Forward Chaining

\[ P \Rightarrow Q \]
\[ L \land M \Rightarrow P \]
\[ B \land L \Rightarrow M \]
\[ A \land P \Rightarrow L \]
\[ A \land B \Rightarrow L \]
\[ A \]
\[ B \]

Forward Chaining Algorithm
Starting from the leaves (facts), propagate the inference up the graph as far as possible. Wherever a conjunction appears, the propagation waits until all conjuncts are known before proceeding.

Is sound and complete (see text).
aka Data Driven Reasoning.
Forward Chaining Inference

\[ R \land Q \Rightarrow P \]

Inference Engine (Forward Chaining)

Tell Q

P!

Backward Chaining Inference

• Works backwards from the query

\[ R \land Q \Rightarrow P \]

Inference Engine (Backward Chaining)

Ask P?

Yes
Backward Chaining

\[ P \Rightarrow Q \]
\[ L \land M \Rightarrow P \]
\[ B \land L \Rightarrow M \]
\[ A \land P \Rightarrow L \]
\[ A \land B \Rightarrow L \]
\[ A \]
\[ B \]

**Backward Chaining Algorithm**

Starting from the query, \( q \) if it is known to be True
Then answer yes. Otherwise, find all implications whose
Conclusion is \( q \). If all the premises of the one of those
Implications can be proved True (by backward chaining), then \( q \) is True.
Is sound and complete (see text).
aka Goal Driven Reasoning.