**Datalog**

1. **Datalog**

Logic as a data model, refer to chapter 12 of AHV.

The meaning of conjunctive queries
- Find valuations from subgoals into facts in a database; emit the output tuple under that valuation.

Example of a datalog program $P$:

$$\begin{align*}
TC(x, y) & : \neg E(x, y) \\
TC(x, z) & : \neg E(x, y), TC(y, z)
\end{align*}$$

**Fact:** The datalog program $P$ cannot be expressed in conjunctive queries.

Three interpretations (or meanings) of a datalog program:
1) Model-Theoretic Interpretation
2) Fixpoint Interpretation
3) Proof-Theoretic Interpretation

They all coincide.

**Syntax of datalog program**

A datalog rule is an expression of the form

$$R_0(x_0) : - R_1(x_1), ..., R_n(x_n)$$

Where $x_i$ are variables of appropriate entry. Each variable occurring in $x_0$ must also occur in $x_i$ for some $i \in [1, n]$. Each $R_i$, $i \in [1, n]$ is a relation (edb: extensional database or idb: intentional database) name.

A datalog program is a finite set of datalog rules

**Logic Program vs. Datalog programs**

Functions symbols are allowed in logic program, but are not allowed in datalog programs.

Example:

$$\begin{align*}
\text{leq}(0, x) & : - \\
\text{leq}(s(x), s(y)) & : \neg \text{leq}(x, y) \\
\text{leq}(x, \text{plus}(x, y)) & : - \\
\text{leq}(x, z) & : \neg \text{leq}(x, y), \text{leq}(y, z)
\end{align*}$$

$s(y)$ and $\text{plus}(x, y)$ are functions.

2. **Model Theoretic Interpretation**

Idea: view each rules as a logical sentence.

$$\rho \ R_0(x_0) : - R_1(x_1), ..., R_n(x_n)$$

has an associated logical sentence.

$$\rho \ \forall x_1, ..., \forall x_n \ ((R_1(x_1) \wedge ... \wedge (R_n(x_n) \rightarrow R_0(x_0)))$$

$x_1, ..., x_n$ are all variables in the rule.
A model (or instance if you prefer) \( I \) satisfies the rule (or its logical sentence) if for every valuation \( \mu \) such that \( R_1(\overline{x_1}) \ldots \ R_n(\overline{x_n}) \) is in \( I \), \( R_0(\overline{x_0}) \) is also in \( I \). Denoted as \( I = \rho \).

Given a datalog program \( P \), let \( \Sigma_\rho \) denote the conjunction of sentences associated with each rules in \( P \). A model of \( P \) is an instance \( I \) over the relation in \( P \) such that \( I = \Sigma_\rho \).

Example:

\[
\begin{align*}
\rho_1 & : R(x) : \neg S(x) \\
\rho_2 & : S(x) : T(x)
\end{align*}
\]

\[ \Sigma = \rho_1 \wedge \rho_2 \]

\[
\begin{align*}
M_0 &= \{s(1), T(0)\} \\
M_1 &= \{T(1), S(1), R(1)\} \\
M_2 &= \{T(1), S(1), R(1), S(2), R(2)\} \\
M_3 &= \{R(1)\} \\
M_4 &= \{R(2)\} \\
M_5 &= \{T(1), S(1)\}
\end{align*}
\]

The right model for \( P \)

Assume our database contains only the relation \( T \) (\( T \) is our edb and \( S \) & \( R \) are our idbs). Given the database instance \( I = \{T(1)\} \) all the following are models of \( P \) w.r.t \( I \).

\[
\begin{align*}
M_1 &= \{T(1), S(1), R(1)\} \\
M_2 &= \{T(1), S(1), R(1), S(2), R(2)\} \\
M_3 &= \{T(1), S(1), R(1), R(3)\}
\end{align*}
\]

Which is “right” model?

\( M_1 \) is special because it is the minimal model. I.e. we cannot discard any fact from \( M_1 \) and still have a model for \( P \) that contains \( I \).

Let \( P \) be a datalog program and \( I \) an instance over the edbs of \( P \). The semantics of \( P \) on input \( I \) denoted as \( P(I) \) is the minimal model of \( P \) containing \( I \).

A fact 1:

\[
B(P,I) \text{ denoted the instance over the schema (edbs & idbs) of } P \text{ such that}
\]

1) For each \( R \) in the edbs of \( P \), a fact \( R(\overline{v}) \in B(P,I) \) iff \( R(\overline{v}) \in I \).

2) For each \( R \) in the idbs of \( P \), a fact \( R(\overline{v}) \) where \( \overline{v} \) contains constants from \( \text{dom}(P,I) \) is in \( B(P,I) \).

Then, \( B(P,I) \) is a model of \( P \) that contains \( I \).

For datalog program \( R(x): S(x) \) and instance \( I = \{T(1), T(2)\} \), \( \text{dom}(P,I) = \{1,2\} \).

And \( B(P,I) = \{T(1), T(2), S(1), S(2), R(1), R(2)\} \)

Proof:

Let \( R_0(\overline{x_0}) : \neg R_1(\overline{x_1}) \ldots \ R_n(\overline{x_n}) \) be a rule in \( P \).

Let \( \mu \) be a valuation so that \( R_1(\mu(x_1)) \ldots \ R_n(\mu(x_n)) \) are facts in \( B(P,I) \).

Then \( B(P,I) \) also contains \( R_0(\mu(x_0)) \) according to (2).

Clearly, \( B(P,I) \) contains \( I \) by (1)
∴ $B(P,I)$ is a model of $P$ that contains $I$.

There is a unique minimal model.

**Fact 2**: let $P$ be a datalog program and $I$ an instance over the edbs of $P$ and $\chi$ the set of models of $P$ containing $I$. Then
1) $\cap \chi$ is the minimal of $P$ containing $I$;
2) $\text{adom}(P(I)) = \text{adom}(P,I)$;
3) For each $R$ in the edbs of $P$, $P(I)(R) = I(R)$.

**Proof**:
For part 1:
Let $\rho$ and $R_0(x_0): -R_1(x_1) \ldots R_k(x_k)$ be a rule in $P$.
Let $\mu$ be a valuation of the variables in $P$.
Claim: if $R_1(\mu(x_1)) \ldots R_k(\mu(x_k))$ occurs in $\cap \chi$ then $R_0(\mu(x_0))$ also occurs in $\cap \chi$.
From the claim, we know $\cap \chi = \rho$, $\cap \chi = \Sigma_P$.
By construction, since every model in $\chi$ contains $I$, $\cap \chi$ contains $I$.
Also by construction, $\cap \chi$ is the minimal model.

**Proof of claim**:
Take any model $K$ in $\chi$, since $\cap \chi \subseteq K$,
$R_1(\mu(x_1)) \ldots R_k(\mu(x_k))$ are all in $K$.
Since $K \models \Sigma_P$, $R_0(\mu(x_0))$ is also in $K$.
∴ $R_0(\mu(x_0)) \in K$ for every $K \in \chi$.

For part 2:
By fact 1, $B(P,I)$ is a model of $P$ containing $I$.
∴ $P(I) \subseteq B(P,I)$
∴ $\text{adom}(P(I)) \subseteq \text{adom}(B(P,I)) = \text{adom}(P,I)$

For part 3:
For each $R$ in the edbs of $P$, $I(R) \subseteq P(I)(R)$.
Because $P(I)$ contains $I$.
Since $P(I)(R) \subseteq B(P,I)(R) = I(R)$
We conclude for each $R$ in the edbs of $P$, $P(I)(R) = I(R)$.

**The choice of minimal model**

Closed world Assumption (CWA)
- Concerns the connection, the database and the world of models.
- A fact that is recorded in the database is considered to be true in the world.
- A fact not recorded in the database may also be true in the world (database may be incomplete).
- CWA assume that every fact recorded in the database is true and otherwise false (the database is complete. All true facts are recorded).
- The minimal model consists of all facts we know must be true in all possible world.
∴ the “right” model