Example

\[ c \rightarrow q \]
\[ c \rightarrow q \]
\[ \text{false} \rightarrow q \land a \]
\[ \text{false} \rightarrow q \land a \]
\[ c \land a \land b \]
\[ c \land a \land b \]

- Can conclude that \( c \) is false in all models of \( KB \)
- If interpretation makes \( c \) true, it would also make \( a \) and \( b \) true, hence would make \( a \land b \) true and hence would need to be made \( false \), contradiction.
- So \( KB \models \neg c \)

Integrity Constraints (Chapter 7.3)

- Have either a clause or a clause on left hand side
- Is either a clause or an integrity constraint
  **Horne Clause**
  - false \( \rightarrow \vdash a \) means that \( a \) has to be false in all models
  - Allows us to specify things that should not be true
  - Which is a contradiction, so body cannot be true
  - If makes the body true, then \( false \) must be true
  - In a model of \( KB \) must make each clause true.
  - Mean that \( a \lor \cdots \lor \neg a \) cannot be true
  - false \( \rightarrow \vdash a \lor \cdots \lor \neg a \)
  
  **Integrity Constraint**

Overview

- Top-Down (Disj. and Neg. Knowledge)
- Proof by Resolution (Horne)
- Bottom-Up
- Resolution Rule
- Disjunctive & Negative Knowledge
- Integrity Constraints
Syntax and Semantics of Or

- **Semantics**
  - If interpretation I and variable assignment U make \( a \) or \( b \) true, then it makes \( a \lor b \) true.

Variations on Integrity Constraints

- **Syntax**
- **Semantics**
  - If interpretation I and variable assignment U make \( c \) false in all models of \( F \), then it makes \( \neg c \) true in all models of \( F \).
  - \( c \) must be false in all models of \( F \).
  - \( \neg c \) must be true in all models of \( F \).

Ad & to Syntax
Overview

• Integrity Constraints
  ⇒ Disjunctive & Negative Knowledge

• Resolution Rule

• Bottom-Up

• Proof by Refutation (Horne)

• Top-Down (Disj. and Neg. Knowledge)

Horne Clauses

• Integrity constraints can be used for diagnostics

Unsatisfiable

• Integrity constraints mean there might not be a model of a KB

Example KB

\[ \text{false} \rightarrow \text{false} \]

Unsatisfiable
Example

Example

Conversion to Conjunctive Normal Form

• Any expression with $\land$, $\lor$, $\neg$, and $\leftarrow$ can be converted into a set of clauses in conjunctive normal form - no literals on right hand side of $\leftarrow$

• Step 1: Eliminate $\leftarrow$

  $\phi \leftarrow \psi$ replaced with $\phi \lor \neg \psi$

• Step 2: Distribute negation so only applies to atoms

  $\neg \neg \phi$ replaced with $\phi$

  $\neg \phi \lor \psi$ replaced with $\neg \phi \land \neg \psi$

  $\neg \phi \land \psi$ replaced with $\neg \phi \lor \neg \psi$

• Step 3: Distribute $\lor$'s over $\land$'s

  $(\phi \lor \psi \land \chi)$ replaced by $((\phi \lor \psi) \land (\phi \lor \chi))$

Disjunctive Knowledge

In Conjunctive Normal Form

Positive literals

Negative literals

Datalog

Exactly 1

Any number

Horne

At most 1

Any number

Disjunctive & Negative

Any number

Any number

• Remove restriction of Horne clauses

  - In fact allow any combination with $\land$, $\lor$, $\neg$, and $\leftarrow$

  - No restrictions on how many positive heads

  - Can be converted into conjunctive normal form

  - Allow atoms to be negated

  - $\phi \rightarrow \psi \land \chi$

  - Allow disjunction in head

  - In Conjunctive Normal Form


Overview

• Integrity Constraints
• Disjunctive & Negative Knowledge
  ⇒ Resolution Rule
• Bottom-Up
• Proof by Refutation (Horne)

Variables

• Recap:
  \( a \lor (b \land c) \)
  \( (a \lor b) \land (a \lor c) \)

Knowledge base:
\{ \{a, b\}, \{a, c\} \}

- Or in set notation for disjuncts:
\{\{a, b\}, \{a, c\}\}

• Now with variables:
  \( \text{male}(X) \lor (\text{female}(X) \land \text{ownsdog}(X)) \)
  \( (\text{male}(X) \lor \text{female}(X)) \land (\text{male}(X) \lor \text{ownsdog}(X)) \)

- If we show the implicit quantifier
  \( \forall X \)

- Variables in two disjuncts seem bound together

- Variables in two disjuncts seem bound together

- Variables in two disjuncts seem bound together

- Can we write it as two separate clauses, each with own X?

- Yes, means the same thing:
  \( \forall X \text{male}(X) \lor \text{female}(X) \land \forall X \text{male}(X) \lor \text{ownsdog}(X) \)

- So, just as in the non-variable case, we can write this as:
  \{\{\text{male}(X), \text{female}(X)\}, \{\text{male}(X), \text{ownsdog}(X)\}\}

Conjunctive Normal Form

• Convert all clause into conjunctive normal form
  - Can view the literals as a set
  - Duplicates are removed

Our KB is now a set of clauses,
where each clause is a set of literals.
Overview

• Integrity Constraints
• Disjunctive & Negative Knowledge
• Resolution Rule
  ⇒ Bottom-Up
  ⇒ Proof by Refutation (Horne)
• Top-Down (Disj. and Neg. Knowledge)

Examples

\{a \lor c, \neg b, \neg c, d\} with \{\neg b, c, b\}

\{a(X) \lor b(Y), \neg c(Y)\} with \{\neg b(Z, Z), c(a)\}

Resolution Rule

- \text{Resolution Rule - Resolvent A includes} \neg a
  - \text{Resolvent B includes} b
  - a \text{ and } b \text{ can be unified}
  - \sigma \text{ is the } \text{MGU of } a \text{ and } b
  - \text{Let } A' \text{ be } A \text{ with } \neg a \text{ removed and } \sigma \text{ applied}
  - \text{Let } B' \text{ be } B \text{ with } b \text{ removed and } \sigma \text{ applied}
  - \text{Resolvent is } A' \cup B'

Resolution Rule
Example I

\[ \neg a \wedge \neg b \Rightarrow \neg c \]

\[ a \wedge b \Rightarrow c \]

\[ \neg c \Rightarrow a \lor b \]

---

Queries to Bottom-Up Proof Procedure

- A query can be a disjunction of positive or negative literals.
- For previous bottom-up procedure, was just a single positive literal.
- Write query as a set of literals.
- If query contains \( A \) and \( \neg A \), obviously true.
- Since one of them is true in any model.
- If query contains \( \neg A \) and \( A \) obviously true.
- Write query as a set of literals.
- If query contains \( A \) and \( \neg A \), obviously true.
- Since one of them is true in any model.
- If there is a member \( e \) of \( C \) and a substitution \( \sigma \) such that \( e\sigma \) is a subset of the query, then the query is true.

---

Bottom-Up Proof Procedure

- Find set of 'minimal' truths.
- Set \( C \) to \( KB \).
- Repeat:
  - Pull two clauses from \( C \).
  - Apply resolution rule if you can, giving \( R \).
  - If \( R \) contains \( A \) and \( \neg A \), skip \( R \), since trivially true.
  - If there is \( R' \) such that \( R \subset R' \), skip \( R \), since already implied.
  - If there is \( R' \) such that \( R \subset R' \), remove \( R' \), since now implied.
- If \( R \) contains \( A \) and \( \neg A \), apply resolution rule on \( A \) and \( \neg A \)
- Update \( C \) with new clauses.
- Repeat.

---

Consensus set no longer just has atoms in it.
- But can have any arbitrary clause in Conjunctive normal form.
- If there are any \( R, C \) such that \( C \subset R \), remove \( C \) since now implied by \( R \).
- If there is \( R, C \) such that \( C \subset R \), remove \( C \), since already implied.
- If there is \( R, C \) such that \( C \subset R \), remove \( C \) since now implied.
- If there is \( R, C \) such that \( C \subset R \), remove \( C \) since now implied.

---

Find set of minimal truths.

Bottom-Up Proof Procedure (Section 7.5)
Horne Clauses and Resolution (not in textbook)

- What is so special about Horne clauses?
  - More powerful than Datalog
  - And an efficient search solution for false

- If \( \text{KB} | t = \{\} \)
  - \( \{\} \) represents the empty clause, which is the same as false
  - No disjuncts in the clause means nothing can make clause true
  - \( \text{KB} \) has no model, which means it is inconsistent

- Why is being able to prove a KB is inconsistent useful?
  - Say we want to prove \( \text{KB} | t = q \), where \( q \) is a literal
  - What would happen if we add \( \neg q \) into KB?

- Overview
  - Integrity Constraints
  - Disjunctive & Negative Knowledge
  - Resolution Rule
  - Bottom-Up
    - Proof by Refutation (Horne)
  - Top-Down (Disj. and Neg. Knowledge)

Example II

\[ (a \lor \neg b) \leftarrow c \]

\[ \neg e \leftarrow \neg c \]

\[ b \lor d \]

\[ (a \lor b) \leftarrow d \]

\[ e \leftarrow \neg a \]

\[ a \lor c \]
Overview

- Integrity Constraints
- Disjunctive & Negative Knowledge
- Resolution Rule
- Bottom-Up
- Proof by Refutation (Horne)

⇒ Top-Down (Disj. and Neg. Knowledge)

Unit Resolution

- Pick two resolvents where one of them is a unit clause
- This is a restricted bottom-up proof procedure
- Unit Resolution always halts
- Unit Resolution is refutation complete
- We say it is refutation complete

Inconsistencies

- Given a KB, there is an efficient way to see if it is inconsistent

\[ \{ b \} \cap \emptyset = \text{false} \]
**Negative Ancestor Rule**

- Can view proof as adding original answer clause KB and trying to prove yes by itself.
- So, should be sound to resolve answer clause with a previous answer clause.
- Didn't need to do this for Datalog as it did not need this to make proof procedure complete.
- But we do need this ability here.

---

**A problem**

- Consider KB $\neg a \lor b$.
- Proof:

Now, rewrite:

- $\neg a \lor b$
- $a \lor b$

In disjunctive normal form:

- $a \land \neg b$
- $\neg a \land b$

Proof:

- Stop when just "yes" atom in answer clause.
- Check if atom is literal in answer clause.
- Use resolution rule to derive new answer clauses.
- Turn into disjunctive normal form.
- For positive top-down procedure it was conjunction of positive literals.
- Start with query, which is a conjunction of literals.
Example

KB

\[(X)d \land (X)\neg b \quad (\neg d \lor (q)b \land (v)b) \quad (X)b \land (X)d \quad (\alpha)b \land (q)b \land (v)b \quad (X)b \rightarrow (X)d\]

Proof

\[(X)d \land (X)\neg b \quad \neg b \rightarrow (X)\neg b \quad (\neg b \lor (q)b \land (v)b) \quad (X)b \land (X)d \quad (\alpha)b \land (q)b \land (v)b \quad (X)b \rightarrow (X)d\]

Rule applied

Yes

\[(X)d \land (X)\neg b \quad (\neg d \lor (q)b \land (v)b) \quad (X)b \land (X)d \quad (\alpha)b \land (q)b \land (v)b \quad (X)b \rightarrow (X)d\]

Answer Clause
Disjunctive Answers

- Dis conjunctive & unknown literals
- Dis conjunctive & known literals
- Dis conjunctive & satisfied literals

We would like proof procedure to find yes

\((q) \wedge (\neg q))(X)\)

\((q) b \land (q) b\) vs \((q) b \land q)(b)\)

Very different from answer that it has two solutions, one with \(X = a\) and another with \(X = b\).

\(\neg X\) \land \((q) b\)

\(\neg X\) \land \((q) b\) with \(q(a)\) and \(q(b)\)

\(\neg X\) \land \((q) b\) vs \(\neg X\) \land \((q) b\)

\((X) b \land (X) d\)

\((X) b \land (X) d\) vs \((X) b \land (X) d\)

\(\neg X\) \land \((X) b\)

\(\neg X\) \land \((X) b\) vs \(\neg X\) \land \((X) b\)

\((q) b \land (q) b\)

\((q) b \land (q) b\) vs \(\neg X\) \land \((X) b\)