UBC Department of Computer Science
Undergraduate Events

More details @ https://my.cs.ubc.ca/students/development/events

Salesforce Info Session
Mon., Oct 26
6 – 7 pm
DMP 310

Dynastream Info Session
Thurs., Oct 29
5:30 – 6:30 pm
DMP 110

Visier Info Session
Tues., Nov 3
12 – 1:30 pm
Kaiser 2020/2030

E-Portfolio Competition Info & Training Session
Wed., Nov 4
5:45 – 7:15 pm
DMP 310

Rakuten Info Session
Thurs., Nov 5
5:30 – 6:30 pm
DMP 110
We model *predicate detection* as a *sequence labeling problem* — ... We adopt the *BIO encoding*, a widely-used technique in NLP. Our method, called Meta-CRF, is based on *Conditional Random Fields (CRF)*. CRF is a graphical model that estimates a conditional probability distribution, denoted $p(y|x)$, over label sequence $y$ given the token sequence $x$. 

---

**PhD thesis I was reviewing some months ago...**

University of Alberta

**EXTRACTING INFORMATION NETWORKS FROM TEXT**

---

\[ \text{ARG} \quad \text{O-REL} \quad \text{O-REL} \quad \text{B-REL} \quad \text{I-REL} \quad \text{ARG} \]

\[ \text{U.S.} \quad \text{is} \quad \text{likely} \quad \text{to} \quad \text{punish} \quad \text{Moscow} \]
422 big picture: Where are we?

Deterministic
- Logics
  - First Order Logics
- Ontologies
  - Temporal rep.
- Full Resolution
- SAT

Stochastic
- Belief Nets
  - Approx. : Gibbs
- Markov Chains and HMMs
  - Forward, Viterbi...
  - Approx. : Particle Filtering
- Undirected Graphical Models
  - Markov Networks
  - Conditional Random Fields
- Markov Decision Processes and Partially Observable MDP
  - Value Iteration
  - Approx. Inference

Hybrid: Det + Sto
- Prob CFG
- Prob Relational Models
- Markov Logics

Applications of AI

Representation
- Reasoning Technique
Logics in AI: Similar slide to the one for planning

- Propositional Definite Clause Logics
- Semantics and Proof Theory
- Satisfiability Testing (SAT)
- First-Order Logics
- Production Systems
- Cognitive Architectures
- Video Games
- Summarization
- Tutoring Systems
- Hardware Verification
- Product Configuration

- Ontologies
- Semantic Web
- Information Extraction
- Description Logics
- Datalog

- BU Sound & Complete
- TD complete
- 422
- CPSC 322, Lecture 19
Relationships between different Logics

First Order Logic

$$\forall X \exists Y p(X, Y) \iff \neg q(Y)$$

$$p(a_1, a_2)$$

$$\neg q(a_3)$$

Propositional Logic

$$\neg(p \lor q) \implies (r \land s \lor f)$$

$$p, r$$

Datalog

$$p(X) \leftarrow q(X) \land r(X, Y)$$

$$r(X, Y) \leftarrow s(Y)$$

$$s(a_1), q(a_2)$$

PDL

$$p \leftarrow s \lor f$$

$$r \leftarrow s \land q \land p$$

$$p$$
Lecture Overview

• Basics Recap: Interpretation / Model /..
• Propositional Logics
• Satisfiability, Validity
• Resolution in Propositional logics
Definition (interpretation)
An interpretation $I$ assigns a truth value to each atom.

Definition (truth values of statements cont’): A knowledge base $KB$ is true in $I$ if and only if every clause in $KB$ is true in $I$. 
A knowledge base KB is true in I if and only if every clause in KB is true in I.

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>l₁</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

Which of the three KB below is True in l₁?

A

p
r
s ← q ∧ p

B

p
q
s ← q

C

p
q ← r ∧ s
PDC Semantics: Knowledge Base (KB)

- A **knowledge base KB** is true in I if and only if every clause in KB is true in I.

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>I₁</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

**KB₁**

\[
p
r
s \leftarrow q \land p
\]

**KB₂**

\[
p
q
s \leftarrow q
\]

**KB₃**

\[
p
q \leftarrow r \land s
\]

Which of the three KB above is True in I₁?  **KB₃**
# Basic definitions from 322 (Semantics)

<table>
<thead>
<tr>
<th><strong>Definition (interpretation)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>An interpretation $I$ assigns a truth value to each atom.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Definition (truth values of statements cont’):</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>A knowledge base $KB$ is true in $I$ if and only if every clause in $KB$ is true in $I$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Definition (model)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>A model of a set of clauses (a KB) is an interpretation in which all the clauses are true.</td>
</tr>
</tbody>
</table>
Example: Models

\[ KB = \begin{cases} 
  p \leftarrow q. \\
  q. \\
  r \leftarrow s. 
\end{cases} \]

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_1 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>( I_3 )</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>( I_4 )</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>( I_5 )</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>

Which interpretations are models?
### Basic definitions from 322 (Semantics)

**Definition (interpretation)**
An interpretation $I$ assigns a truth value to each atom.

**Definition (truth values of statements cont’):** A knowledge base $KB$ is true in $I$ if and only if every clause in $KB$ is true in $I$.

**Definition (model)**
A model of a set of clauses (a KB) is an interpretation in which all the clauses are *true*.

**Definition (logical consequence)**
If $KB$ is a set of clauses and $G$ is a conjunction of atoms, $G$ is a logical consequence of $KB$, written $KB \vdash G$, if $G$ is *true* in every model of $KB$. 

---

**CPSC 322, Lecture 19**
Is it true that if

\[ M(KB) \] is the set of all models of KB
\[ M(\alpha) \] is the set of all models of \( \alpha \)

Then \( KB \models \alpha \) if and only if \( M(KB) \subseteq M(\alpha) \)

\[ \alpha \] true in all the models of \( KB \)

A. yes
B. no
C. It depends

All interpretations
Definition (soundness)
A proof procedure is **sound** if $KB \vdash G$ implies $KB \vDash G$.

Definition (completeness)
A proof procedure is **complete** if $KB \vDash G$ implies $KB \vdash G$. 
Lecture Overview

• Basics Recap: Interpretation / Model /
• Propositional Logics
• Satisfiability, Validity
• Resolution in Propositional logics
Relationships between different Logics

First Order Logic

\( \forall X \exists Y p(X,Y) \iff \neg q(Y) \)

- \( p(a_1, a_2) \)
- \( \neg q(a_5) \)

Propositional Logic

\( \neg (p \lor q) \rightarrow (r \land s \land t) \)

- \( p, r \)

Datalog

\( p(X) \leftarrow q(X) \land r(X,Y) \)

- \( r(X,Y) \leftarrow s(Y) \)

- \( s(a_1), q(a_2) \)

PDCL

- \( p \leftarrow s \land f \)

- \( r \leftarrow s \land q \land p \)

- \( r \leftarrow p \)
Propositional logic: Syntax

Atomic sentences = single proposition symbols

- E.g., P, Q, R
- Special cases: True = always true, False = always false

Complex sentences:

- If S is a sentence, $\neg S$ is a sentence (negation)
- If $S_1$ and $S_2$ are sentences, $S_1 \land S_2$ is a sentence (conjunction)
- If $S_1$ and $S_2$ are sentences, $S_1 \lor S_2$ is a sentence (disjunction)
- If $S_1$ and $S_2$ are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)
- If $S_1$ and $S_2$ are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)
Propositional logic: Semantics

Each interpretation specifies true or false for each proposition symbol

E.g.  

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

Rules for evaluating truth with respect to an interpretation I:

\[-S\text{ is true iff } S\text{ is false}\]

\[S_1 \land S_2 \text{ is true iff } S_1 \text{ is true and } S_2 \text{ is true}\]

\[S_1 \lor S_2 \text{ is true iff } S_1 \text{ is true or } S_2 \text{ is true}\]

\[S_1 \implies S_2 \text{ is true iff } S_1 \text{ is false or } S_2 \text{ is true}\]

\ i.e., \[S_1 \implies S_2 \text{ is false iff } S_1 \text{ is true and } S_2 \text{ is false}\]

\[S_1 \iff S_2 \text{ is true iff } S_1 \implies S_2 \text{ is true and } S_2 \implies S_1 \text{ is true}\]

Simple recursive process evaluates an arbitrary sentence, e.g.,

\[\neg p \land (q \lor r) \iff \neg p = (T \land F) \iff T\]

\[= (F \lor (T \land F)) \iff T\]

\[= (T \land T) \iff T\]

\[= \neg p \iff \neg p = T\]
Logical equivalence

Two sentences are logically equivalent iff true in same interpretations
\( \alpha \equiv \beta \) if and only if \( \alpha \models \beta \) and \( \beta \models \alpha \)

\[
\begin{align*}
(\alpha \land \beta) & \equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) & \equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) & \equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) & \equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\neg(\neg \alpha) & \equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) & \equiv (\neg \beta \Rightarrow \neg \alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) & \equiv (\neg \alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) & \equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \land \beta) & \equiv (\neg \alpha \lor \neg \beta) \quad \text{De Morgan} \\
\neg(\alpha \lor \beta) & \equiv (\neg \alpha \land \neg \beta) \quad \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) & \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) & \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{align*}
\]
\[
\begin{align*}
(\alpha \land \beta) & \equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) & \equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) & \equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) & \equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\neg(\neg \alpha) & \equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) & \equiv (\neg \beta \Rightarrow \neg \alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) & \equiv (\neg \alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \iff \beta) & \equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \land \beta) & \equiv (\neg \alpha \lor \neg \beta) \quad \text{De Morgan} \\
\neg(\alpha \lor \beta) & \equiv (\neg \alpha \land \neg \beta) \quad \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) & \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) & \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{align*}
\]

Can be used to rewrite formulas…. 

\[
(\neg p \lor \neg (q \land r)) \\
\leftarrow \neg p \lor \neg (q \land r) \\
\rightarrow \neg p \lor (q \lor r)
\]
Can be used to rewrite formulas…

\( (\alpha \land (\beta \lor \gamma)) \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \) distributivity of \( \land \) over \( \lor \)

\( (\alpha \lor (\beta \land \gamma)) \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \) distributivity of \( \lor \) over \( \land \)
Validity and satisfiability

A sentence is **valid** if it is true in **all** interpretations

\[ \text{e.g., } \text{True, } A \lor \neg A, \quad A \Rightarrow A, \quad (A \land (A \Rightarrow B)) \Rightarrow B \]\n
Validity is connected to inference via the **Deduction Theorem**:

\[ KB \models \alpha \text{ if and only if } (KB \Rightarrow \alpha) \text{ is valid} \]

A sentence is **satisfiable** if it is true in **some** interpretation

\[ \text{e.g., } A \lor B, \quad C \]

A sentence is **unsatisfiable** if it is true in **no** interpretations

\[ \text{e.g., } A \land \neg A \]

Satisfiability is connected to inference via the following:

\[ KB \models \alpha \text{ if and only if } (KB \land \neg \alpha) \text{ is unsatisfiable} \]

i.e., prove \( \alpha \) by **reductio ad absurdum**
Validity and Satisfiability

\( \alpha \text{ is valid iff } \neg \alpha \text{ is unsatisfiable} \)

\( \alpha \text{ is satisfiable iff } \neg \alpha \text{ is valid} \)

The statements above are:

A: All false
B: Some true, Some false
C: All true
Validity and Satisfiability

\(< \alpha \text{ is valid } \iff \text{ it is unsatisfiable}>\)

\(< \alpha \text{ is satisfiable } \iff \text{ it is valid} >\)

The statements above are:

A: All false
B: Some true Some false
C: All true
Lecture Overview

• Basics Recap: Interpretation / Model /
• Propositional Logics
• Satisfiability, Validity
• Resolution in Propositional logics
Proof by resolution

Key ideas

- Simple Representation for
- Simple Rule of Derivation

\[ KB \models \alpha \]

Equivalent to: \( KB \land \neg \alpha \) unsatisfiable

 Conjunctive Normal Form

Resolution
Conjunctive Normal Form (CNF)

Rewrite $KB \land \neg \alpha$ into conjunction of disjunctions

\[(A \lor \neg B) \land (B \lor \neg C \lor \neg D)\]

- Any KB can be converted into CNF!
Example: Conversion to CNF

\[ A \iff (B \lor C) \]

1. Eliminate \( \iff \), replacing \( \alpha \iff \beta \) with \( (\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha) \).
   \[(A \Rightarrow (B \lor C)) \land ((B \lor C) \Rightarrow A)\]

2. Eliminate \( \Rightarrow \), replacing \( \alpha \Rightarrow \beta \) with \( \neg \alpha \lor \beta \).
   \[(\neg A \lor B \lor C) \land (\neg (B \lor C) \lor A)\]

3. Using de Morgan's rule replace \( \neg (\alpha \lor \beta) \) with \( (\neg \alpha \land \neg \beta) \):
   \[(\neg A \lor B \lor C) \land (\neg B \land \neg C) \lor A\]

4. Apply distributive law (\( \lor \) over \( \land \)) and flatten:
   \[(\neg A \lor B \lor C) \land (\neg B \lor A) \land (\neg C \lor A)\]
Example: Conversion to CNF

A \iff (B \lor C)

5. KB is the conjunction of all of its sentences (all are true), so write each clause (disjunct) as a sentence in KB:

\[
\begin{align*}
\ldots \\
(\neg A \lor B \lor C) \\
(\neg B \lor A) \\
(\neg C \lor A) \\
\ldots 
\end{align*}
\]
Resolution Deduction step

Resolution: inference rule for CNF: sound and complete! *

\[(A \lor B \lor C)\]
\[\neg A\]

\[\therefore (B \lor C)\]

“If A or B or C is true, but not A, then B or C must be true.”

\[(A \lor B \lor C)\]
\[\neg A \lor D \lor E\]

\[\therefore (B \lor C \lor D \lor E)\]

“If A is false then B or C must be true, or if A is true then D or E must be true, hence since A is either true or false, B or C or D or E must be true.”

\[(A \lor B)\]
\[\neg A \lor B\]

\[\therefore (B \lor B) \equiv B\]

Simplification

CPSC 322, Lecture 19
Learning Goals for today’s class

You can:

• Describe relationships between different logics
• Apply the definitions of Interpretation, model, logical entailment, soundness and completeness
• Define and apply satisfiability and validity
• Convert any formula to CNF
• Justify and apply the resolution step
We model *predicate detection* as a *sequence labeling problem* — …. We adopt the *BIO encoding*, a widely-used technique in NLP.

Our method, called Meta-CRF, is based on *Conditional Random Fields (CRF)*. CRF is a graphical model that estimates a conditional probability distribution, denoted $p(y|\mathbf{x})$, over label sequence $y$ given the token sequence $\mathbf{x}$.

---

**EXTRACTING INFORMATION NETWORKS FROM TEXT**

---

PhD thesis I was reviewing some months ago…

University of Alberta
Next class Fri

• Finish Resolution

• Another proof method for Prop. Logic
  Model checking - Searching through truth assignments. Walksat.

• First Order Logics
Preview: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.

That is, the procedure will answer any question whose answer follows from what is known by the $KB$. 
Try it Yourselves

- 7.9 page 238: (Adapted from Barwise and Etchemendy (1993).) If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

- Derive the KB in normal form.
- Prove: Horned, Prove: Magical.
Exposes useful constraints

- “You can’t learn what you can’t represent.” --- G. Sussman

- **In logic:** If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.
  
  Prove that the unicorn is both magical and horned.

- A good representation makes this problem easy:

\[
(\neg Y \lor \neg R) \land (Y \lor R) \land (Y \lor M) \land (R \lor H) \land (\neg M \lor H) \land (\neg H \lor G)
\]
Resolution

Conjunctive Normal Form (CNF—universal)

conjunction of \( \text{disjunctions of literals} \)

clauses

E.g., \((A \lor \neg B) \land (B \lor \neg C \lor \neg D)\)

Resolution inference rule (for CNF): complete for propositional logic

\[
\ell_1 \lor \cdots \lor \ell_k, \quad m_1 \lor \cdots \lor m_n
\]

\[
\ell_1 \lor \cdots \lor \ell_{i-1} \lor \ell_{i+1} \lor \cdots \lor \ell_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n
\]

where \(\ell_i\) and \(m_j\) are complementary literals. E.g.,

\[
P_{1,3} \lor P_{2,2}, \quad \neg P_{2,2}
\]

\[
P_{1,3}
\]

Resolution is sound and complete for propositional logic
Conversion to CNF

\[ B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \]

1. Eliminate \( \iff \), replacing \( \alpha \iff \beta \) with \( (\alpha \implies \beta) \land (\beta \implies \alpha) \).
   \[ (B_{1,1} \implies (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \implies B_{1,1}) \]

2. Eliminate \( \implies \), replacing \( \alpha \implies \beta \) with \( \neg \alpha \lor \beta \).
   \[ (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg (P_{1,2} \lor P_{2,1}) \lor B_{1,1}) \]

3. Move \( \neg \) inwards using de Morgan’s rules and double-negation:
   \[ (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land ((\neg P_{1,2} \land \neg P_{2,1}) \lor B_{1,1}) \]

4. Apply distributivity law (\( \lor \) over \( \land \)) and flatten:
   \[ (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg P_{1,2} \lor B_{1,1}) \land (\neg P_{2,1} \lor B_{1,1}) \]
Resolution example

\[ KB = (B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})) \land \neg B_{1,1} \land \alpha = \neg P_{1,2} \]
Forward, backward chaining are linear-time, complete for Horn clauses.
Resolution is complete for propositional logic.
Propositional logic lacks expressive power.
Logical equivalence

To manipulate logical sentences we need some rewrite rules. Two sentences are logically equivalent iff they are true in same models: $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

- $(\alpha \land \beta) \equiv (\beta \land \alpha)$ commutativity of $\land$
- $(\alpha \lor \beta) \equiv (\beta \lor \alpha)$ commutativity of $\lor$
- $((\alpha \land \beta) \land \gamma) \equiv (\alpha \land (\beta \land \gamma))$ associativity of $\land$
- $((\alpha \lor \beta) \lor \gamma) \equiv (\alpha \lor (\beta \lor \gamma))$ associativity of $\lor$
- $\neg(\neg \alpha) \equiv \alpha$ double-negation elimination
- $(\alpha \Rightarrow \beta) \equiv (\neg \beta \Rightarrow \neg \alpha)$ contraposition
- $(\alpha \Rightarrow \beta) \equiv (\neg \alpha \lor \beta)$ implication elimination
- $(\alpha \leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha))$ biconditional elimination
- $\neg(\alpha \land \beta) \equiv (\neg \alpha \lor \neg \beta)$ de Morgan
- $\neg(\alpha \lor \beta) \equiv (\neg \alpha \land \neg \beta)$ de Morgan
- $(\alpha \land (\beta \lor \gamma)) \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma))$ distributivity of $\land$ over $\lor$
- $(\alpha \lor (\beta \land \gamma)) \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma))$ distributivity of $\lor$ over $\land$
Validity and satisfiability

A sentence is valid if it is true in all models,
e.g., True, $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$
(tautologies)

Validity is connected to inference via the Deduction Theorem:
$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is satisfiable if it is true in some model
e.g., $A \lor B$, $C$
(determining satisfiability of sentences is NP-complete)

A sentence is unsatisfiable if it is false in all models
Proof methods

Proof methods divide into (roughly) two kinds:

**Application of inference rules:**
- Legitimate (sound) generation of new sentences from old.
  - Resolution
  - Forward & Backward chaining

**Model checking**
- Searching through truth assignments.
  - Improved backtracking: Davis--Putnam-Logemann-Loveland (DPLL)
  - Heuristic search in model space: Walksat.
Normal Form

We want to prove: \( KB \models \alpha \)

\( equivalent \ to: \ KB \land \neg \alpha \) unsatisfiable

We first rewrite \( KB \land \neg \alpha \) into conjunctive normal form (CNF).

A “conjunction of disjunctions”

\((A \lor \neg B) \land (B \lor \neg C \lor \neg D)\)

- Any KB can be converted into CNF
- k-CNФ: exactly k literals per clause
Example: Conversion to CNF

\[ B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \]

1. Eliminate \( \iff \), replacing \( a \iff \beta \) with \( (a \implies \beta) \land (\beta \implies a) \).
   \[ (B_{1,1} \implies (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \implies B_{1,1}) \]

2. Eliminate \( \implies \), replacing \( a \implies \beta \) with \( \neg a \lor \beta \).
   \[ (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg (P_{1,2} \lor P_{2,1}) \lor B_{1,1}) \]

3. Move \( \neg \) inwards using de Morgan's rules and double-negation:
   \[ (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land ((\neg P_{1,2} \land \neg P_{2,1}) \lor B_{1,1}) \]

4. Apply distributive law (\( \land \) over \( \lor \)) and flatten:
   \[ (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg P_{1,2} \lor B_{1,1}) \land (\neg P_{2,1} \lor B_{1,1}) \]
Resolution Inference Rule for CNF

\[(A \lor B \lor C)\]
\[\neg A\]
\[\therefore (B \lor C)\]

“If A or B or C is true, but not A, then B or C must be true.”

\[(A \lor B \lor C)\]
\[\neg A \lor D \lor E\]
\[\therefore (B \lor C \lor D \lor E)\]

“If A is false then B or C must be true, or if A is true then D or E must be true, hence since A is either true or false, B or C or D or E must be true.”

\[(A \lor B)\]
\[\neg A \lor B\]
\[\therefore (B \lor B) \equiv B\]

Simplification
Resolution Algorithm

The resolution algorithm tries to prove: $KB \models \alpha$ equivalent to $KB \wedge \neg \alpha$ unsatisfiable

Generate all new sentences from KB and the query.
One of two things can happen:

1. We find $P \wedge \neg P$ which is unsatisfiable, i.e. we can entail the query.

2. We find no contradiction: there is a model that satisfies the Sentence (non-trivial) and hence we cannot entail the query.

$KB \wedge \neg \alpha$
Resolution example

- $KB = (B_{1,1} \leftrightarrow (P_{1,2} \lor P_{2,1})) \land \lnot B_{1,1}$
- $\alpha = \lnot P_{1,2}$

$KB \land \lnot \alpha$

True

False in all worlds
Horn Clauses

• Resolution in general can be exponential in space and time.

• If we can reduce all clauses to “Horn clauses” resolution is linear in space and time

A clause with at most 1 positive literal.

- e.g. $A \lor \neg B \lor \neg C$

- Every Horn clause can be rewritten as an implication with a conjunction of positive literals in the premises and a single positive literal as a conclusion.

- e.g. $B \land C \Rightarrow A$

- 1 positive literal: definite clause

- 0 positive literals: Fact or integrity constraint:

  - e.g. $(\neg A \lor \neg B) \equiv (A \land B \Rightarrow False)$
Normal Form

We want to prove:

\[ KB \models \alpha \]

\[ \text{equivalent to: } KB \land \neg \alpha \text{ unsatisfiable} \]

We first rewrite

\[ KB \land \neg \alpha \]

into conjunctive normal form (CNF).

A “conjunction of disjunctions”

\[ (A \lor \neg B) \land (B \lor \neg C \lor \neg D) \]

Clause     Clause

• Any KB can be converted into CNF
• k-CNF: exactly k literals per clause
Example: Conversion to CNF

\[ B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \]

1. Eliminate \( \iff \), replacing \( \alpha \iff \beta \) with \( (\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha) \).
   \[ (B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1}) \]

2. Eliminate \( \Rightarrow \), replacing \( \alpha \Rightarrow \beta \) with \( \neg \alpha \lor \beta \).
   \[ (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg (P_{1,2} \lor P_{2,1}) \lor B_{1,1}) \]

3. Move \( \neg \) inwards using de Morgan's rules and double-negation:
   \[ (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land ((\neg P_{1,2} \land \neg P_{2,1}) \lor B_{1,1}) \]

4. Apply distributive law (\( \land \) over \( \lor \)) and flatten:
   \[ (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg P_{1,2} \lor B_{1,1}) \land (\neg P_{2,1} \lor B_{1,1}) \]

CPSC 322, Lecture 19
Resolution Inference Rule for CNF

\[(A ∨ B ∨ C)\]
\[\neg A\]
\[\therefore (B ∨ C)\]

"If A or B or C is true, but not A, then B or C must be true."

\[(A ∨ B ∨ C)\]
\[\neg A ∨ D ∨ E\]
\[\therefore (B ∨ C ∨ D ∨ E)\]

"If A is false then B or C must be true, or if A is true then D or E must be true, hence since A is either true or false, B or C or D or E must be true."

\[(A ∨ B)\]
\[\neg A ∨ B\]
\[\therefore (B ∨ B) ≡ B\]

Simplification

CPSC 322, Lecture 19
Resolution Algorithm

- The resolution algorithm tries to prove: $\text{KB} |\models \alpha$ equivalent to $\text{KB} \land \neg\alpha$ unsatisfiable

- Generate all new sentences from KB and the query.
- One of two things can happen:

1. We find $P \land \neg P$ which is unsatisfiable, i.e. we can entail the query.

2. We find no contradiction: there is a model that satisfies the Sentence (non-trivial) and hence we cannot entail the query.

$\text{KB} \land \neg\alpha$
Resolution example

\[ KB = (B_{1,1} \iff (P_{1,2} \lor P_{2,1})) \land \neg B_{1,1} \]

\[ \alpha = \neg P_{1,2} \]

\[ KB \land \neg \alpha \]

True

False in all worlds

CPSC 322, Lecture 19
Horn Clauses

• Resolution in general can be exponential in space and time.

• If we can reduce all clauses to “Horn clauses” resolution is linear.

A clause with at most 1 positive literal.

\( A \lor \neg B \lor \neg C \)

e.g.

• Every Horn clause can be rewritten as an implication with a conjunction of positive literals in the premises and a single positive literal as a conclusion.

\( B \land C \Rightarrow A \)

e.g.

\( (\neg A \lor \neg B) \equiv (A \land B \Rightarrow False) \)

• 1 positive literal: definite clause