Horn clauses and disjunction

Written with implication	**Written with disjunction**
$A \rightarrow C$	$\neg A \vee C$
$A \wedge B \rightarrow C$	$\neg A \vee \neg B \vee C$

- **Literal**: either p or $\neg p$, where p is a propositional symbol
- **Clause**: disjunction of literals $A \vee B$ $A \vee B \vee \neg C$
- **Horn clauses**: at most one positive literal

Horn clauses and disjunction

Written with implication	**Written with disjunction**
$A \rightarrow C$	$\neg A \vee C$
$A \wedge B \rightarrow C$	$\neg A \vee \neg B \vee C$

- **Literal**: either p or $\neg p$, where p is a propositional symbol
- **Clause**: disjunction of literals
- **Horn clauses**: at most one positive literal

Modus ponens (rewritten):

$$\frac{A, \quad \neg A \vee C}{C}$$

$$\frac{A \quad B \quad \neg A \vee \neg B \vee C}{C}$$

- Intuition: cancel out A and $\neg A$

Conjunctive normal form

So far: resolution only works on clauses...but that's enough!

Definition: conjunctive normal form (CNF)

A **CNF formula** is a conjunction of clauses.

Example: $(A \vee B \vee \neg C) \wedge (\neg B \vee D)$

Equivalent: knowledge base where each formula is a clause

$KB = \{f_1, f_2, \dots\}$

$KB \models f$

$f_1 \wedge f_2 \wedge \dots \models f$

Conversion to CNF: example

$A \leftrightarrow B$ $A \to B \wedge B \to A$

Initial formula:

$$(\text{Summer} \to \text{Snow}) \to \text{Bizzare}$$

Conversion to CNF: example

Initial formula:

$$(\text{Summer} \to \text{Snow}) \to \text{Bizzare}$$

Remove implication ($\to$):

$$\neg(\neg\text{Summer} \lor \text{Snow}) \lor \text{Bizzare}$$

Push negation ($\neg$) inwards (de Morgan):

$$(\neg\neg\text{Summer} \land \neg\text{Snow}) \lor \text{Bizzare}$$

$\neg(f \land g) \equiv \neg f \lor \neg g$

$\neg(f \lor g) \equiv \neg f \land \neg g$

Conversion to CNF: example

Initial formula:

$$(\text{Summer} \to \text{Snow}) \to \text{Bizzare}$$

Remove implication ($\to$):

$$\neg(\neg\text{Summer} \vee \text{Snow}) \vee \text{Bizzare}$$

Push negation ($\neg$) inwards (de Morgan):

$$(\neg\neg\text{Summer} \wedge \neg\text{Snow}) \vee \text{Bizzare}$$

Remove double negation:

$$(\text{Summer} \wedge \neg\text{Snow}) \vee \text{Bizzare}$$

$$f \vee (g \wedge h) \equiv (f \vee g) \wedge (f \vee h)$$

Resolution: example

$$\mathrm{KB}' = \{\underline{A \to (B \lor C)}, A, \neg B, \neg C\}$$

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

Convert to CNF:

$$\mathrm{KB}' = \{\neg A \lor B \lor C, A, \neg B, \neg C\}$$

Summary

Horn clauses	**any clauses**
modus ponens	resolution
linear time	exponential time
less expressive	more expressive

$A \vee B$

Syntax of first-order logic

Terms (refer to objects):

- Constant symbol (e.g., $\mathbf{arithmetic}$)
- Variable (e.g., x)
- Function of terms (e.g., $\mathrm{Sum}(3, x)$)

Sum(Sum(x, y), 3)

Syntax of first-order logic

Terms (refer to objects):

- Constant symbol (e.g., $\mathrm{arithmetic}$)
- Variable (e.g., x)
- Function of terms (e.g., $\mathrm{Sum}(3, x)$)

Formulas (refer to truth values):

- Atomic formulas (atoms): predicate applied to terms (e.g., $\mathrm{Knows}(x, \mathrm{arithmetic})$)
- Connectives applied to formulas (e.g., $\mathrm{Student}(x) \rightarrow \mathrm{Knows}(x, \mathrm{arithmetic})$)
- Quantifiers applied to formulas (e.g., $\forall x\, \mathrm{Student}(x) \rightarrow \mathrm{Knows}(x, \mathrm{arithmetic})$)

Given a formula f:
- Free variables
- Bound "

Some examples of first-order logic

There is some course that every student has taken.

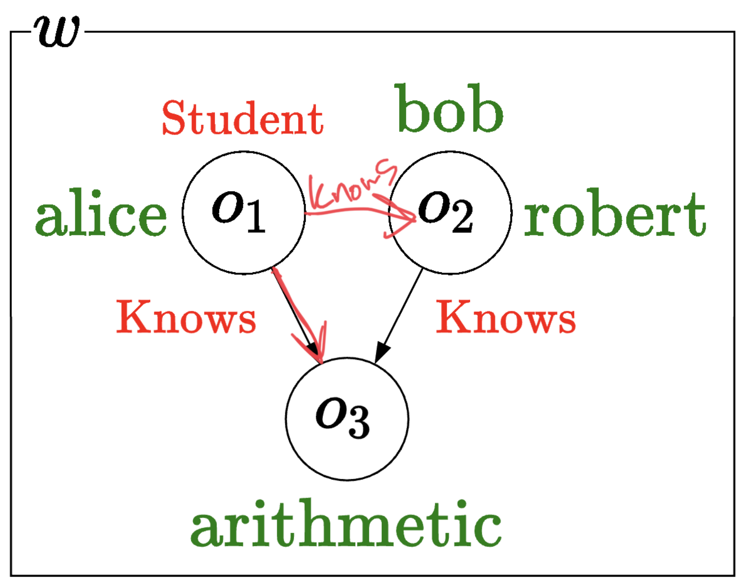
$$\exists y\, \text{Course}(y) \wedge [\forall x\, \text{Student}(x) \rightarrow \text{Takes}(x, y)]$$

Every even integer greater than 2 is the sum of two primes.

$$\forall x\, \text{EvenInt}(x) \wedge \text{Greater}(x, 2) \rightarrow \exists y\, \exists z\, \text{Equals}(x, \text{Sum}(y, z)) \wedge \text{Prime}(y) \wedge \text{Prime}(z)$$

Graph representation of a model

If only have unary and binary predicates, a model w can be represented as a directed graph:



Knows(·, ·)

Student(·)

- Nodes are objects, labeled with constant symbols
- Directed edges are binary predicates, labeled with predicate symbols; unary predicates are additional node labels

Definite clauses

$$\forall x\, \forall y\, \forall z\, (\mathrm{Takes}(x, y) \wedge \mathrm{Covers}(y, z)) \rightarrow \mathrm{Knows}(x, z)$$

Note: if propositionalize, get one formula for each value to (x, y, z), e.g., $(\mathrm{alice}, \mathrm{cs221}, \mathrm{mdp})$

Definition: definite clause (first-order logic)

Knows(x_1, x_2)

Knows(F(x_1), x_2)

A definite clause has the following form:

$$\forall x_1 \cdots \forall x_n\, (a_1 \wedge \cdots \wedge a_k) \rightarrow b$$

for variables $x_1, \ldots, x_n$ and atomic formulas $a_1, \ldots, a_k, b$ (which contain those variables).

Unification

$$\text{Unify}[\text{Knows}(\text{alice}, \text{arithmetic}), \text{Knows}(x, \text{arithmetic})] = \{x/\text{alice}\}$$

$$\text{Unify}[\text{Knows}(\text{alice}, y), \text{Knows}(x, z)] = \{x/\text{alice}, y/z\}$$

{x/alice, y/bob, z/bob}

Conversion to CNF (part 2)

$\forall w$ $\forall x\, (\exists y\, \mathbf{Animal}(y) \wedge \neg \mathbf{Loves}(x, y)) \vee \exists z\, \mathbf{Loves}(z, x)$

Replace existentially quantified variables with Skolem functions (**new**): $Y(w, x)$

$$\forall x\, [\mathbf{Animal}(\cancel{Y}(x)) \wedge \neg \mathbf{Loves}(x, Y(x))] \vee \mathbf{Loves}(Z(x), x)$$