

Clause Normal Form

Clause Normal Form (CNF) is a sub-language of 1st order logic. A clause is an expression of the form $\text{L}_1 \mid \dots \mid \text{L}_m$ where each L_i is a literal. Clauses are denoted by uppercase letters with a superscript \mid , e.g., C^1 .

There are satisfiability preserving transformations from 1st order logic to CNF, i.e., if a set of (1st order) formulae are satisfiable, then their CNF is satisfiable. Conversely, if the CNF of a set of formulae is unsatisfiable, then the formulae are unsatisfiable. This is then useful for showing logical consequence. The benefit of converting to CNF is that more is possible using just the Herbrand interpretations. Computationally, CNF is easier to work with, and is the form used by the resolution inference rule.

Transforming to Clause Normal Form

There are several algorithms for this conversion, but they all have some parts in common, and all have similarities. The best algorithm is the one which produces the CNF which is most easily shown to be unsatisfiable. In general such a determination is impossible to make, but a common measure is to aim to produce a CNF with the fewest symbols (where a symbol is a variable, predicate, or constant). The key feature of all the algorithms is that they are satisfiability preserving.

The starting point is a set of closed formulae. As a preprocessing step, all variables must be renamed apart.

Rename variables apart

The first operation is to *simplify* the formulae so that they contain only the \forall , \exists , $\&$, \mid and \sim connectives. This is done by using known logical identities. The identities are easily verified using truth tables, to show that the original formula has the same value as the simplified formula for all possible interpretations of the component formulae, denoted by uppercase variable letters P and Q .

Simplify	
Formula	Rewrites to
$P \text{ <}\sim\text{> } Q$	$(P \mid Q) \ \& \ (\sim P \mid \sim Q)$
$P \text{ <=}\Rightarrow\text{> } Q$	$(P \Rightarrow Q) \ \& \ (Q \Rightarrow P)$
$P \Rightarrow Q$	$\sim P \mid Q$

The second operation is to *move negations in* so that they apply to only atoms. This is done by using known logical identities (including De Morgan's laws), which are easily verified as above. After moving negations in the formulae are in *literal normal form*.

Move negations in	
Formula	Rewrites to
$\sim\forall X \ P$	$\exists X \ \sim P$
$\sim\exists X \ P$	$\forall X \ \sim P$
$\sim(P \ \& \ Q)$	$(\sim P \mid \sim Q)$
$\sim(P \mid Q)$	$(\sim P \ \& \ \sim Q)$
$\sim\sim P$	P

The third operation is to *move quantifiers out* so that each formula is in the scope of all the quantifiers in that formula. This is done by using known logical identities, which are easily verified as above. In these identities, if the variable x already exists in Q , it is renamed to a new variable name that does not exist in the formula. After moving quantifiers out the formulae are in *prenex normal form*.

Move quantifiers out	
Formula	Rewrites to
$\forall x \ P \ \& \ Q$	$\forall x \ (P \ \& \ Q)$
$\exists x \ P \ \& \ Q$	$\exists x \ (P \ \& \ Q)$
$Q \ \& \ \forall x \ P$	$\forall x \ (Q \ \& \ P)$
$Q \ \& \ \exists x \ P$	$\exists x \ (Q \ \& \ P)$
$\forall x \ P \ \ Q$	$\forall x \ (P \ \ Q)$
$\exists x \ P \ \ Q$	$\exists x \ (P \ \ Q)$
$Q \ \ \forall x \ P$	$\forall x \ (Q \ \ P)$
$Q \ \ \exists x P$	$\exists x \ (Q \ \ P)$

The fourth operation is to *Skolemize* to remove existential quantifiers. This step replaces existentially quantified variables by *Skolem functions* and removes all quantifiers. The variables remaining after Skolemization are all implicitly universally quantified. Whereas the previous three operations maintain equivalence between the original and transformed formulae, Skolemization does not. However, Skolemization does maintain the satisfiability of the formulae, which is what is required for the overall goal of establishing logical consequence. After Skolemization the formulae are in *Skolem normal form*.

Skolemize	
Formula	Rewrites to
Initially	Set $\gamma = \{\}$
$\forall x \ P$	P and set $\gamma = \gamma \cup \{x\}$
$\exists x \ P$	$P[x/\gamma(x)]$ where x is a new Skolem functor.

The fifth operation is to *distribute disjunctions* so that the formulae are conjunctions of disjunctions. This is done by using known logical identities, which are easily verified as above. After distributing disjunctions the formulae are in *conjunctive normal form*.

Distribute disjunctions	
Formula	Rewrites to
$P \ \ (Q \ \& \ R)$	$(P \ \ Q) \ \& \ (P \ \ R)$
$(Q \ \& \ R) \ \ P$	$(Q \ \ P) \ \& \ (R \ \ P)$

The last operation is to *convert to Clause Normal Form*. This is done by removing the conjunctions, to form a set of clauses.

Convert to CNF	
Formula	Rewrites to
$P_1 \ \& \ \dots \ \& \ P_n$	$\{P_1, \dots, P_n\}$

Example

1st order formula
$\forall Y (\forall X (\text{taller}(Y,X) \mid \text{wise}(X)) \Rightarrow \text{wise}(Y))$
Simplify
$\forall Y (\sim \forall X (\text{taller}(Y,X) \mid \text{wise}(X)) \mid \text{wise}(Y))$
Move negations in
$\forall Y (\exists X (\sim \text{taller}(Y,X) \ \& \ \sim \text{wise}(X)) \mid \text{wise}(Y))$
Move quantifiers out
$\forall Y (\exists X ((\sim \text{taller}(Y,X) \ \& \ \sim \text{wise}(X)) \mid \text{wise}(Y)))$
Skolemize
$\exists X ((\sim \text{taller}(Y,X) \ \& \ \sim \text{wise}(X)) \mid \text{wise}(Y)) \ \forall = \{Y\}$ $(\sim \text{taller}(Y,x(Y)) \ \& \ \sim \text{wise}(x(Y))) \mid \text{wise}(Y)$
Distribute disjunctions
$(\sim \text{taller}(Y,x(Y)) \mid \text{wise}(Y)) \ \& \ (\sim \text{wise}(x(Y)) \mid \text{wise}(Y))$
Convert to CNF
$\{ \sim \text{taller}(Y,x(Y)) \mid \text{wise}(Y),$ $\sim \text{wise}(x(Y)) \mid \text{wise}(Y) \}$

Special Forms

Clauses may be expressed in Kowalski form. Kowalski form forms the antecedent of an implication by conjoining the atoms of the negative literals in a clause, and forms the consequent from the disjunction of the positive literals.

Example
$\sim \text{taller}(Y,x(Y)) \mid \text{wise}(Y) \mid \sim \text{wise}(x(Y)) \mid \text{taller}(x(Y),Y)$
is written as
$\text{taller}(Y,x(Y)) \ \& \ \text{wise}(x(Y)) \Rightarrow \text{wise}(Y) \mid \text{taller}(x(Y),Y)$
If the antecedent is empty it is set to TRUE, and if the consequent is empty it is set to FALSE.

Horn clauses are clauses that have one or zero positive literals. Sets of Horn clauses are also called Horn.

Example
$\sim \text{taller}(Y,x(Y)) \mid \text{wise}(Y) \mid \sim \text{wise}(x(Y))$ $\sim \text{wise}(\text{geoff}) \mid \sim \text{taller}(\text{Person},\text{brother_of}(\text{jim}))$
are Horn clauses. Written in Kowalski form, these Horn clauses are :
$\text{taller}(Y,x(Y)) \ \& \ \text{wise}(x(Y)) \Rightarrow \text{wise}(Y)$ $\text{wise}(\text{geoff}) \ \& \ \text{taller}(\text{Person},\text{brother_of}(\text{jim})) \Rightarrow \text{FALSE}$

Non-Horn clauses are clauses that have two or more positive literals. Sets of clauses that contain one or more non-Horn clauses are also called non-Horn.

Example

```
~taller(Y,x(Y)) | ~wise(Y) | wise(x(Y)) | taller(x(Y),x(x(Y)))
wise(X) | wise(brother_of(X))
```

are non-Horn clauses. Written in Kowalski form, these non-Horn clauses are :

```
taller(Y,x(Y)) & wise(Y) => wise(x(Y)) | taller(x(Y),x(x(Y)))
TRUE => wise(X) | wise(brother_of(X))
```

In terms of ATP, the disjunction of positive literals in non-Horn clauses corresponds to a choice point in any goal directed search. This is a source of search. Thus, in general, Horn problems are easier than non-Horn problems.

Exercises

Convert the following to CNF:

- $\sim \exists X (s(X) \ \& \ q(X))$
Answer: $\{ \sim s(X) \mid \sim q(X) \}$
- $\forall X (p(X) \Rightarrow (q(X) \mid r(X)))$
Answer: $\{ \sim p(X) \mid q(X) \mid r(X) \}$
- $\sim \exists X (p(X) \Rightarrow \exists X \ q(X))$
Answer: $\{ p(X), \sim q(X1) \}$
- $\forall X (q(X) \mid r(X) \Rightarrow s(X))$
Answer: $\{ \sim q(X) \mid s(X), \sim r(X) \mid s(X) \}$
- $\exists X (p \Rightarrow f(X))$
Answer: $\{ \sim p \mid f(skX) \}$
- $\exists X (p \Leftrightarrow f(X))$
Answer: $\{ \sim p \mid f(skX), p \mid \sim f(skX) \}$
- $\forall Z \ \exists Y \ \forall X (f(X,Y) \Leftrightarrow (f(X,Z) \ \& \ \sim f(X,X)))$
Answer: $\{ \sim f(A,skA(B)) \mid f(A,B), \sim f(A,skA(B)) \mid \sim f(A,A), \sim f(A,B) \mid f(A,A) \mid f(A,skA(B)) \}$
- $\forall X \ \forall Y (q(X,Y) \Leftrightarrow \forall Z (f(Z,X) \Leftrightarrow f(Z,Y)))$
Answer: $\{ \sim q(A,B) \mid \sim f(C,A) \mid f(C,B), \sim q(A,B) \mid \sim f(C,B) \mid f(C,A), f(sk1(A,B),B) \mid f(sk1(A,B),A) \mid q(B,A), f(sk1(A,B),B) \mid \sim f(sk1(A,B),B) \mid q(B,A), \sim f(sk1(A,B),A) \mid f(sk1(A,B),A) \mid q(B,A), \sim f(sk1(A,B),A) \mid \sim f(sk1(A,B),B) \mid q(B,A) \}$
- $\exists X (\ \exists Y (p(X,Y) \ \& \ q(Y)) \Rightarrow \exists Z (p(Z,X) \ \& \ q(Z)) \)$
Answer: TBA
- $\forall X \ \exists Y ((p(X,Y) \ \Leftarrow \ \forall X \ \exists T \ q(Y,X,T) \) \Rightarrow r(Y))$
Answer: TBA
- $\forall X \ \forall Z (p(X,Z) \Rightarrow \exists Y \sim (q(X,Y) \mid \sim r(Y,Z)))$
Answer: TBA

The Quest for Logical Consequence

To show that C is a logical consequence of A, it is necessary to:

- Show that $S' = A \cup \{\sim C\}$ is unsatisfiable.

S' is unsatisfiable iff S = CNF(S') is unsatisfiable

- Show that S is unsatisfiable.

Exam Style Questions

1. Convert the following to CNF. Notation: & for AND, | for OR, ! for universal quantifier, ? for existential quantifier.

```
( ? [Y] :  
  ! [X] :  
    ( element(X,Y)  
      <=> element(X,X) )  
=> ~ ( ! [X1] :  
      ? [Y1] :  
      ! [Z] :  
        ( element(Z,Y1)  
          <=> ~ element(Z,X1) ) ) )
```

2. Explain, with examples, the difference between Horn and non-Horn clauses.

3. Explain, with examples, what is meant by Kowalski form for clauses. Why is Kowalski form important?
