Knowledge Representation & Reasoning

Shyamanta M Hazarika
CSE, SoE
Tezpur University

Horn clauses

Clauses are used two ways:

- as disjunctions: (rain $\vee$ sleet)
- as implications: ($\neg$child $\vee$ $\neg$male $\vee$ boy)

Here focus on 2nd use

Horn clause = at most one +ve literal in clause

- positive / definite clause = exactly one +ve literal

  e.g. $[\neg p_1, \neg p_2, ..., \neg p_n, q]$

- negative clause = no +ve literals

  e.g. $[\neg p_1, \neg p_2, ..., \neg p_n]$ and also [ ]

Note: $[\neg p_1, \neg p_2, ..., \neg p_n, q]$ is a representation for
$(\neg p_1 \vee \neg p_2 \vee ... \vee \neg p_n \vee q)$ or $[(p_1 \wedge p_2 \wedge ... \wedge p_n) \supset q]$

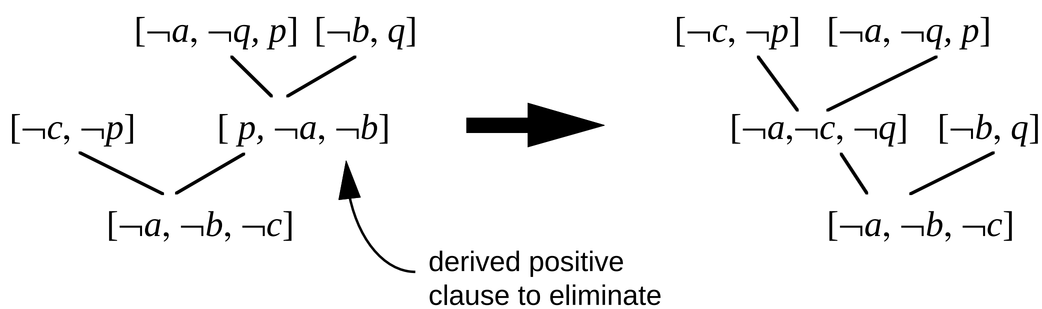
so can read as: If p_1 and p_2 and ... and p_n then q

and write as: $p_1 \wedge p_2 \wedge ... \wedge p_n \Rightarrow q$ or $q \Leftarrow p_1 \wedge p_2 \wedge ... \wedge p_n$

Resolution with Horn clauses

Only two possibilities:

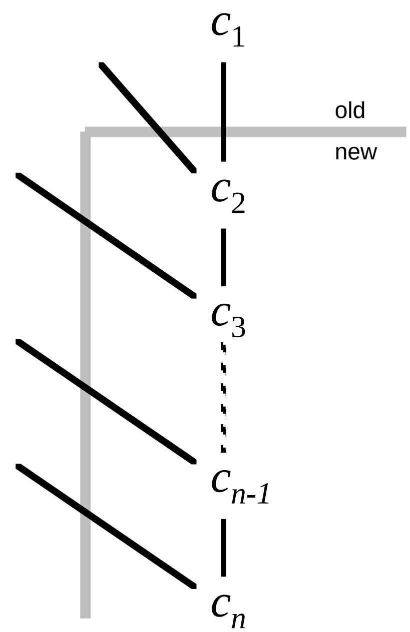
It is possible to rearrange derivations of negative clauses so that all new derived clauses are negative



Further restricting resolution

Can also change derivations such that each derived clause is a resolvent of the previous derived one (negative) and some positive clause in the original set of clauses

- Since each derived clause is negative, one parent must be positive (and so from original set) and one parent must be negative.
- Chain backwards from the final negative clause until both parents are from the original set of clauses
- Eliminate all other clauses not on this direct path



This is a recurring pattern in derivations

- See previously:
  - example 1, example 3, arithmetic example
- But not:
  - example 2, the 3 block example

SLD Resolution

An <u>SLD-derivation</u> of a clause c from a set of clauses *S* is a sequence of clause $c_1, c_2, \ldots c_n$ such that $c_n = c$, and

1. $c_1 \in S$
2. c_{i+1} is a resolvent of c_i and a clause in *S*

Write: $S \overset{\text{SLD}}{\rightarrow} c$

SLD means S(elected) literals
L(inear) form
D(efinite) clauses

Note: SLD derivation is just a special form of derivation and where we leave out the elements of *S* (except c_1)

In general, cannot restrict ourselves to just using SLD-Resolution

Proof: $S = \{[p, q], [p, \neg q], [\neg p, q]\ [\neg p, \neg q]\}$. Then $S \rightarrow []$.

Need to resolve some $[\,\rho\,]$ and $[\,\overline{\rho}\,]$ to get [].
But *S* does not contain any unit clauses.

So will need to derive both $[\,\rho\,]$ and $[\,\overline{\rho}\,]$ and then resolve them together.

Completeness of SLD

However, for Horn clauses, we can restrict ourselves to SLD-Resolution

Theorem: SLD-Resolution is refutation complete for Horn clauses: $H \rightarrow []$ iff $H \overset{\text{SLD}}{\rightarrow} []$

So: H is unsatisfiable iff $H \overset{\text{SLD}}{\rightarrow} []$

This will considerably simplify the search for derivations

Note: in Horn version of SLD-Resolution, each clause in the $c_1, c_2, ..., c_n$, will be negative

So clauses H must contain at least one negative clause, c_1 and this will be the only negative clause of H used.

Typical case:

- KB is a collection of positive Horn clauses
- Negation of query is the negative clause

Example 1 (again)

KB

FirstGrade

FirstGrade $\supset$ Child

Child $\wedge$ Male $\supset$ Boy

Kindergarten $\supset$ Child

Child $\wedge$ Female $\supset$ Girl

Female

SLD derivation

[$\neg$Girl]

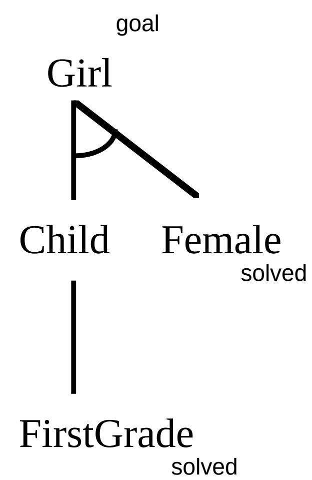
[$\neg$Child, $\neg$Female]

[$\neg$Child]

[$\neg$FirstGrade]

[]

alternate representation



Show KB $\cup$ {$\neg$Girl} unsatisfiable

A goal tree whose nodes are atoms, whose root is the atom to prove, and whose leaves are in the KB