Propositional Resolution Part 3

Short Review

Professor Anita Wasilewska

CSE 352 Artificial Intelligence

Resolution Strategies

- **We present here some Deletion Strategies and discuss their Completeness.**

Deletion Strategies are restriction techniques in which clauses with specified properties are eliminated from set of clauses **CL** before they are used.

Pure Literals

Pure literal definition

A literal is **pure** in **CL** iff it has no complementary literal in any other clause in **CL**

Example: **CL** = { {a,b},{¬ c, d},{c,b}, {¬ d}}

a, b are **pure**, c, d, ¬ c, ¬ d are **not pure**.

c has complement literal ¬ c in {¬ c, d} and
vice versa, ¬ c has complement literal c in {c,b}.
d has a complement literal ¬d in the clause {¬ d} and
vice versa ¬d has a complement literal d in {¬ c, d}.

1. Pure Literals Deletion Strategy

Strategy: Remove all clauses that contain Pure Literals.

Clauses that contain pure literals are useless for retention process. One pure literal in a clause is enough for the clause removal.

This Strategy is complete, i.e.

CL $\vdash$ {} iff CL' $\vdash$ {}

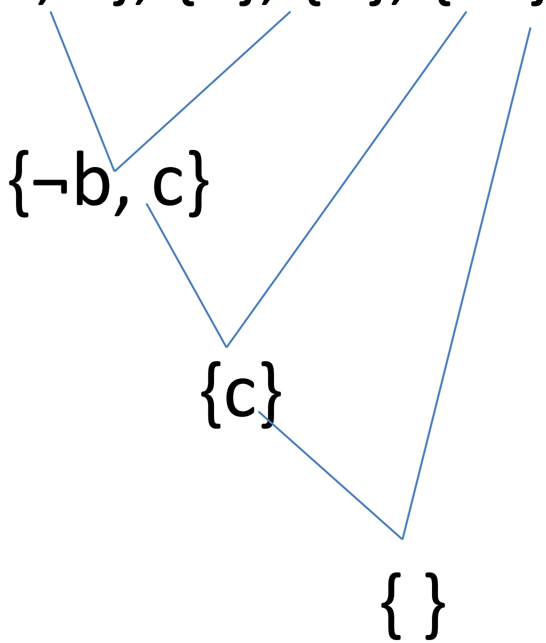
where **CL'** is obtained from **CL** by pure literal deletion

Example

- **CL** = {{¬a, ¬b, c}, {¬p, d}, {¬b, d}, {a}, {b}, {¬c}}

  d, ¬p are pure,

CL' = {{¬a, ¬b, c}, {a}, {b}, {¬c}}



2. Tautology Deletion Strategy

- **Tautology – A clause containing a pair of Complementary Literals** (a and ¬a)
- **Tautology Deletion:**
- **CL'** = Remove all Tautologies from **CL**
- Example:
- **CL** = {{ a, b, ¬a}, {b, ¬b, c}, {a}}
- **CL'** = {{a}}
- Tautology Strategy is **COMPLETE**.

CL is satisfiable **≡** **CL'** **is** satisfiable

CL unsatisfiable **≡** **CL'** unsatisfiable

Exercise

- Example:
- **CL** = {{ a, ¬a, b}, {b, ¬b, c}} - remove tautologies;
- **CL'** has no elements, i.e. **CL'** = ϕ,

CL is always satisfiable and so is **CL'** as Φ is always satisfiable!

Exercise: Prove Correctness of Tautology delete strategy.

Case 1: **CL** contains only tautologies

In this case **CL'** = ϕ because Φ is always satisfiable!

Case 2:

3. Unit Resolution Strategy

- **A unit resolvent** – resolvent in which at least one of the parent clauses is **a unit clause** i.e. is a clause containing a single literal.
- **A unit deduction** – all derived clauses are unit resolvents.
- **A unit Refutation** – **unit deduction of the empty clause { }.**
- Example: {{a, b}, {¬a, c}, {¬b, c}, {¬c}}

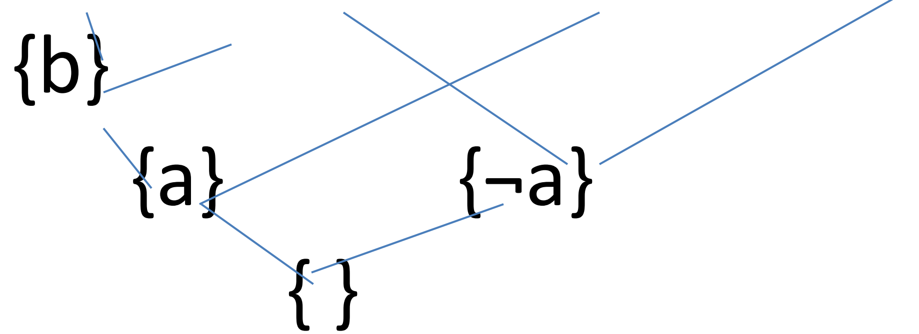
{¬a} {¬b}

{b}

{ } **Efficient but not Complete!**

Unit Resolution not complete Example

- **CL** = {{a, b}, {¬a, b}, {a, ¬b}, {¬a, ¬b}}



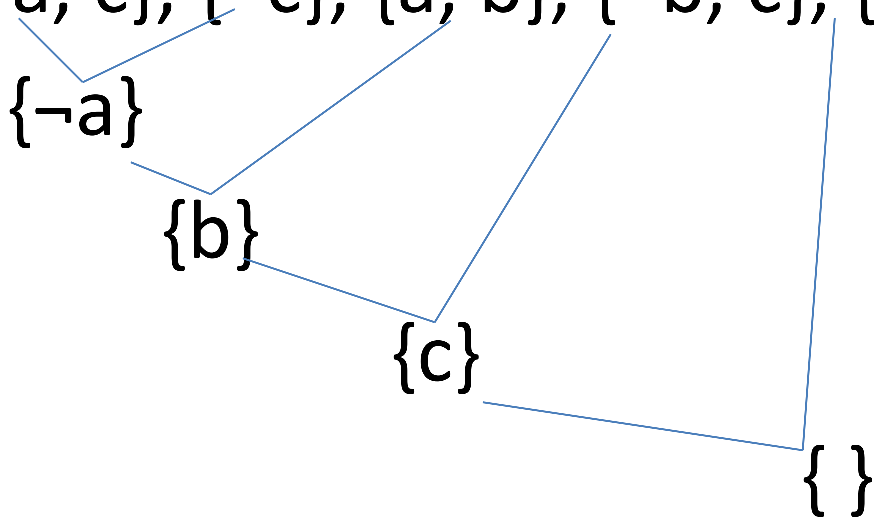
CL is **unsatisfiable**, but does not have unit deduction.

Horn Clause: a clause with at most one positive literal.

Theorem: Unit Resolution is complete on Horn Clauses.

Example of Unit Resolution Deduction

- **CL** = {{¬a, c}, {¬c}, {a, b}, {¬b, c}, {¬c}}



CL is not Horn but **CL⊢ {}** by unit deduction.

Remark: if we get { } by unit deduction we are OK but if we don’t get { } by unit deduction it does not mean that **CL** is satisfiable, because unit strategy is not a Complete Strategy on non- Horn clauses.

4. Input Resolution

- **Input Resolution- At least one of the two parent clauses is in the initial database.**
- **Input Deduction-** all derived clauses are input resolvents.
- **Input Refutation-** Input deduction of { }.

THM 1: Unit and Input Resolution are equivalent.

THM 2: Input Resolution is complete only on Horn Clauses.

Input Resolution Deduction

Example: **CL** = {{a, b}, {¬a, c}, {¬b, c}, {¬c}}

{b, c}

{c}

{}

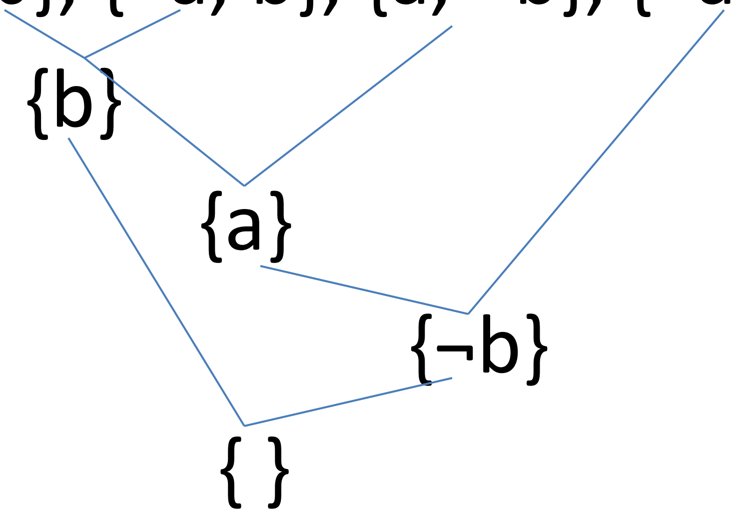
NOT Complete!

5. Linear Resolution

- **Linear Resolution** also called **Ancestry-Filtered** resolution is a slight generalization of Input Resolution.
- **A Linear Resolution: At least one of the parents is either in the initial DB or is in an Ancestor of the other parent.**
- **A Linear Deduction:** Uses only linear resolvents : each derived clauses is a linear resolvent
- **A Linear Refutation:** Linear deduction of { }.
- **Linear Resolution is complete**

Example

- Δ = {{a, b}, {¬a, b}, {a, ¬b}, {¬a, ¬ b}}



Here :

{a} is a parent of {¬b}

{b} is the ancestor of {¬b} (other parent of {¬b})

Linear Resolution

Linear Resolution is complete

There are also more modifications of the LR that are **complete**

Our Strategies work also for Predicate Logic Resolution.

Kowalski 1974, 1976 “Logic for problem solving” “Predicate Logic as a programming language”.

Robinson 1965 “A Machinery Oriented logic based on the resolution principle” J Assoc. for Computing Machinery 12(1)