CS 228 : Logic in Computer Science

Krishna. S

Polynomial Time Formula Classes

Horn Formulae

- A formula F is a Horn formula if it is in CNF and every disjunction contains at most one positive literal.
- $p \wedge (\neg p \vee \neg q \vee r) \wedge (\neg a \vee \neg b)$ is Horn, but $a \vee b$ is not Horn.
- A basic Horn formula is one which has no $\wedge$. Every Horn formula is a conjunction of basic Horn formulae.

Horn Formulae

- Three types of basic Horn : no positive literals, no negative literals, have both positive and negative literals.
- Basic Horn with both positive and negative literals are written as an implication $p \wedge q \wedge \cdots \wedge r \to s$ involving only positive literals.
- Basic Horn with no negative literals are of the form p and are written as $\top \to p$.
- Basic Horn with no positive literals are written as $p \wedge q \wedge \cdots \wedge r \to \bot$.
- Thus, a Horn formula is written as a conjunction of implications.

The Horn Algorithm

Given a Horn formula H,

- Mark all occurrences of p, whenever $\top \to p$ is a subformula.

The Horn Algorithm

Given a Horn formula H,

- Mark all occurrences of p, whenever $\top \rightarrow p$ is a subformula.
- If there is a subformula of the form $(p_1 \wedge \cdots \wedge p_m) \rightarrow q$, where each p_i is marked, and q is not marked, mark q. Repeat this until there are no subformulae of this form and proceed to the next step.

The Horn Algorithm

Given a Horn formula H,

- Mark all occurrences of p, whenever $\top \rightarrow p$ is a subformula.
- If there is a subformula of the form $(p_1 \wedge \cdots \wedge p_m) \rightarrow q$, where each p_i is marked, and q is not marked, mark q. Repeat this until there are no subformulae of this form and proceed to the next step.
- Consider subformulae of the form $(p_1 \wedge \cdots \wedge p_m) \rightarrow \bot$. If there is one such subformula with all p_i marked, then say Unsat, otherwise say Sat.

An Example

$$(\top \to A) \wedge (C \to D) \wedge ((A \wedge B) \to C) \wedge ((C \wedge D) \to \bot) \wedge (\top \to B).$$

An Example

$(\top \to A) \land (C \to D) \land ((A \land B) \to C) \land ((C \land D) \to \bot) \land (\top \to B).$

- $(\top \to A) \land (C \to D) \land ((A \land B) \to C) \land ((C \land D) \to \bot) \land (\top \to B).$

An Example

$(\top \to A) \land (C \to D) \land ((A \land B) \to C) \land ((C \land D) \to \bot) \land (\top \to B).$

- $(\top \to A) \land (C \to D) \land ((A \land B) \to C) \land ((C \land D) \to \bot) \land (\top \to B).$
- $(\top \to A) \land (C \to D) \land ((A \land B) \to C) \land ((C \land D) \to \bot) \land (\top \to B).$

An Example

$(\top \to A) \wedge (C \to D) \wedge ((A \wedge B) \to C) \wedge ((C \wedge D) \to \bot) \wedge (\top \to B)$.

- $(\top \to A) \wedge (C \to D) \wedge ((A \wedge B) \to C) \wedge ((C \wedge D) \to \bot) \wedge (\top \to B)$.
- $(\top \to A) \wedge (C \to D) \wedge ((A \wedge B) \to C) \wedge ((C \wedge D) \to \bot) \wedge (\top \to B)$.
- $(\top \to A) \wedge (C \to D) \wedge ((A \wedge B) \to C) \wedge ((C \wedge D) \to \bot) \wedge (\top \to B)$.

An Example

$(\top \to A) \wedge (C \to D) \wedge ((A \wedge B) \to C) \wedge ((C \wedge D) \to \bot) \wedge (\top \to B).$

- $(\top \to A) \wedge (C \to D) \wedge ((A \wedge B) \to C) \wedge ((C \wedge D) \to \bot) \wedge (\top \to B).$
- $(\top \to A) \wedge (C \to D) \wedge ((A \wedge B) \to C) \wedge ((C \wedge D) \to \bot) \wedge (\top \to B).$
- $(\top \to A) \wedge (C \to D) \wedge ((A \wedge B) \to C) \wedge ((C \wedge D) \to \bot) \wedge (\top \to B).$
- $(\top \to A) \wedge (C \to D) \wedge ((A \wedge B) \to C) \wedge ((C \wedge D) \to \bot) \wedge (\top \to B).$

The Horn Algorithm

The Horn algorithm concludes Sat iff H is satisfiable.

Complexity of Horn

- Given a Horn formula ψ with *n* propositions, how many times do you have to read ψ?
- Step 1: Read once
- Step 2: Read atmost *n* times
- Step 3: Read once

2-CNF

- 2-CNF : CNF where each clause has at most 2 literals.

Resolution

- Resolution is a technique used to check if a formula in CNF is unsatisfiable.

Resolution

- Resolution is a technique used to check if a formula in CNF is unsatisfiable.
- CNF notation as set of sets : $(p \vee q) \wedge (\neg p \vee q) \wedge p$ represented as $\{\{p, q\}, \{\neg p, q\}, \{p\}\}$

Resolution

- Resolution is a technique used to check if a formula in CNF is unsatisfiable.
- CNF notation as set of sets : $(p \vee q) \wedge (\neg p \vee q) \wedge p$ represented as $\{\{p, q\}, \{\neg p, q\}, \{p\}\}$
- Let C_1, C_2 be two clauses. Assume $p \in C_1$ and $\neg p \in C_2$ for some literal p. Then the clause $R = (C_1 - \{p\}) \cup (C_2 - \{\neg p\})$ is a resolvent of C_1 and C_2.

Resolution

- Resolution is a technique used to check if a formula in CNF is unsatisfiable.
- CNF notation as set of sets : $(p \vee q) \wedge (\neg p \vee q) \wedge p$ represented as $\{\{p, q\}, \{\neg p, q\}, \{p\}\}$
- Let C_1, C_2 be two clauses. Assume $p \in C_1$ and $\neg p \in C_2$ for some literal p. Then the clause $R = (C_1 - \{p\}) \cup (C_2 - \{\neg p\})$ is a resolvent of C_1 and C_2.
- Let $C_1 = \{p_1, \neg p_2, p_3\}$ and $C_2 = \{p_2, \neg p_3, p_4\}$. As $p_3 \in C_1$ and $\neg p_3 \in C_2$, we can find the resolvent. The resolvent is $\{p_1, p_2, \neg p_2, p_4\}$.

Resolution

- Resolution is a technique used to check if a formula in CNF is unsatisfiable.
- CNF notation as set of sets : $(p \vee q) \wedge (\neg p \vee q) \wedge p$ represented as $\{\{p, q\}, \{\neg p, q\}, \{p\}\}$
- Let C_1, C_2 be two clauses. Assume $p \in C_1$ and $\neg p \in C_2$ for some literal p. Then the clause $R = (C_1 - \{p\}) \cup (C_2 - \{\neg p\})$ is a resolvent of C_1 and C_2.
- Let $C_1 = \{p_1, \neg p_2, p_3\}$ and $C_2 = \{p_2, \neg p_3, p_4\}$. As $p_3 \in C_1$ and $\neg p_3 \in C_2$, we can find the resolvent. The resolvent is $\{p_1, p_2, \neg p_2, p_4\}$.
- Resolvent not unique : $\{p_1, p_3, \neg p_3, p_4\}$ is also a resolvent.

3 rules in Resolution

- Let G be any formula. Let F be the CNF formula resulting from the CNF algorithm applied to G. Then $G \vdash F$ (Prove!)

3 rules in Resolution

- Let G be any formula. Let F be the CNF formula resulting from the CNF algorithm applied to G. Then $G \vdash F$ (Prove!)
- Let F be a formula in CNF, and let C be a clause in F. Then $F \vdash C$ (Prove!)

3 rules in Resolution

- Let G be any formula. Let F be the CNF formula resulting from the CNF algorithm applied to G. Then $G \vdash F$ (Prove!)
- Let F be a formula in CNF, and let C be a clause in F. Then $F \vdash C$ (Prove!)
- Let F be a formula in CNF. Let R be a resolvent of two clauses of F. Then $F \vdash R$ (Prove!)

Completeness of Resolution

Show that resolution can be used to determine whether any given formula is unsatisfiable.

- Given F in CNF, let $Res^0(F) = \{C \mid C \text{ is a clause in } F\}$.

Completeness of Resolution

Show that resolution can be used to determine whether any given formula is unsatisfiable.

- Given F in CNF, let $Res^0(F) = \{C \mid C \text{ is a clause in } F\}$.
- $Res^n(F) = Res^{n-1}(F) \cup \{R \mid R \text{ is a resolvent of two clauses in } Res^{n-1}(F)\}$

Completeness of Resolution

Show that resolution can be used to determine whether any given formula is unsatisfiable.

- Given F in CNF, let $Res^0(F) = \{C \mid C \text{ is a clause in } F\}$.
- $Res^n(F) = Res^{n-1}(F) \cup \{R \mid R \text{ is a resolvent of two clauses in } Res^{n-1}(F)\}$
- $Res^0(F) = F$, there are finitely many clauses that can be derived from F.

Completeness of Resolution

Show that resolution can be used to determine whether any given formula is unsatisfiable.

- Given F in CNF, let $Res^0(F) = \{C \mid C \text{ is a clause in } F\}$.
- $Res^n(F) = Res^{n-1}(F) \cup \{R \mid R \text{ is a resolvent of two clauses in } Res^{n-1}(F)\}$
- $Res^0(F) = F$, there are finitely many clauses that can be derived from F.
- There is some $m \geqslant 0$ such that $Res^m(F) = Res^{m+1}(F)$. Denote it by $Res^*(F)$.

Example

Let $F = \{\{p_1, p_2, \neg p_3\}, \{\neg p_2, p_3\}\}$.

- $Res^0(F) = F$

Example

Let $F = \{\{p_1, p_2, \neg p_3\}, \{\neg p_2, p_3\}\}$.

- $Res^0(F) = F$
- $Res^1(F) = F \cup \{p_1, p_2, \neg p_2\} \cup \{p_1, \neg p_3, p_3\}$.

Example

Let $F = \{\{p_1, p_2, \neg p_3\}, \{\neg p_2, p_3\}\}$.

- $Res^0(F) = F$
- $Res^1(F) = F \cup \{p_1, p_2, \neg p_2\} \cup \{p_1, \neg p_3, p_3\}$.
- $Res^2(F) = Res^1(F) \cup \{p_1, p_2, \neg p_3\} \cup \{p_1, p_3, \neg p_2\}$