CS228 Logic for Computer Science 2023

Lecture 10: k-SAT

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2023-01-24

Topic 10.1

k-sat

k-sat

Definition 10.1

A k-sat formula is a CNF formula and has at most k literals in each of its clauses

Example 10.1

- $(p \wedge q \wedge \neg r)$ *is 1-SAT*
- $(p \vee \neg p) \wedge (p \vee q)$ *is 2-SAT*
- $(p \vee \neg q \vee \neg s) \wedge (p \vee q) \wedge \neg r$ *is 3-SAT*

3-SAT satisfiablity

Theorem 10.1

For each k-SAT formula F there is a 3-SAT formula F' with linear blow up such that F and F' are equisatisfiable.

Proof.

Consider F a k-SAT formula with $k \geq 4$.
Consider a clause $G = (\ell_1 \vee \cdots \vee \ell_k)$ in F, where ℓ_i are literals.

Let $x_2, \ldots, x_{k-2}$ be variables that are not in $Vars(F)$.
Let G' be the following set of clauses

$$(\ell_1 \vee \ell_2 \vee x_2) \wedge \bigwedge_{i \in 2..k-3} (\neg x_i \vee x_{i+1} \vee \ell_{i+1}) \wedge (\neg x_{k-2} \vee \ell_{k-1} \vee \ell_k).$$

We show F is sat iff $(F - \{G\}) \cup G'$ is sat. ...

Exercise 10.1

Convert $(p \vee \neg q \vee s \vee \neg t) \wedge (\neg q \vee x \vee \neg y \vee z)$ into a 3-SAT formula

3-SAT satisfiability(cont. I)

Proof(contd. from last slide).

Recall

$$G' = (\ell_1 \vee \ell_2 \vee x_2) \wedge \bigwedge_{i \in 2..k-3} (\neg x_i \vee x_{i+1} \vee \ell_{i+1}) \wedge (\neg x_{k-2} \vee \ell_{k-1} \vee \ell_k).$$

Assume $m \models (F - \{G\}) \cup G'$:

Assume for each $i \in 1..k$, $m(\ell_i) = 0$.

Due to the first clause $m(x_2) = 1$.

Due to ith clause, if $m(x_i) = 1$ then $m(x_{i+1}) = 1$.

Due to induction, $m(x_{k-2}) = 1$.

Due to the last clause of G', $m(x_{k-2}) = 0$. Contradiction.

Therefore, there is $i \in 1..k$ such that $m(\ell_i) = 1$. Therefore $m \models G$. Therefore, $m \models F$

...

3-SAT satisfiability(cont. II)

Proof(contd. from last slide).

Recall $G' = (\ell_1 \vee \ell_2 \vee x_2) \wedge \bigwedge_{i \in 2..k-3} (\neg x_i \vee x_{i+1} \vee \ell_{i+1}) \wedge (\neg x_{k-2} \vee \ell_{k-1} \vee \ell_k)$.

Assume $m \models F$:

Therefore, $m \models G$.

There is a $m(\ell_i) = 1$.

Let $m' \triangleq m[x_2 \mapsto 1, .., x_{i-1} \mapsto 1, x_i \mapsto 0, \ldots, x_{k-2} \mapsto 0]$.

Therefore, $m' \models F - \{G\}$ and $m' \models G'_{(\text{why?})}$. Therefore, $m' \models (F - \{G\}) \cup G'$.

G' contains $3(k-2)$ literals. In the worst case, the formula size will increase 3 times. □

Commentary: Tseitin is a trick. We can apply it anywhere. Can we can convert 3-SAT to 2-SAT? All the usual tricks including Tseitin do not work. Unfortunately, We do not have a proof that 3-SAT cannot be converted to 2-SAT with in polynomial blow up. This transformation is the heart of the question P=?=NP.

Exercise 10.2

a. Complete the above argument.

b. Show a 3-SAT cannot be converted into a 2-SAT via Tseitin encoding.

c. When is the worst case?

Special classes of formulas

We will discuss the following polynomial time subclasses of SAT problem

- 2-SAT
- Horn clauses
- XOR-SAT

Commentary: For the curious, Schaefer's dichotomy theorem identified subclasses of SAT problem that are polynomial and all others are NP-complete. The above subclasses play an important role in the theorem. The theorem suggests that no other subclass is polynomial.

Topic 10.2

2-SAT

2-SAT

Definition 10.2

A 2-sat formula is a CNF formula that has only binary clauses

We assume that unit clauses are replaced by clauses with repeated literals.

Example 10.2

- $(\neg p \vee q) \wedge (\neg q \vee r) \wedge (\neg r \vee p) \wedge (r \vee q)$ *is a 2-SAT formula*
- $(p \vee p) \wedge (\neg p \vee \neg p)$ *is a 2-SAT formula*

Implication graph

Definition 10.3

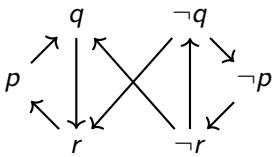
Let F be a 2-SAT formula such that $\mathbf{Vars}(F) = \{p_1, \ldots, p_n\}$. *The implication graph (V, E) for F is defined as follows.*

$$V \triangleq \{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\} \qquad E \triangleq \{(\bar{\ell_1}, \ell_2), (\bar{\ell_2}, \ell_1) | (\ell_1 \vee \ell_2) \in F\},$$

where $\bar{p} = \neg p$ and $\overline{\neg p} = p$.

Example 10.3

Consider $(\neg p \vee q) \wedge (\neg q \vee r) \wedge (\neg r \vee p) \wedge (r \vee q)$.



Exercise : implication graph

Exercise 10.3

Draw implication graphs of the following

1. $(p \vee q) \wedge (\neg p \vee \neg q)$
2. $(p \vee \neg q) \wedge (q \vee p) \wedge (\neg p \vee \neg r) \wedge (r \vee \neg p)$
3. $(p \vee p) \wedge (\neg p \vee \neg p)$
4. $(p \vee \neg p) \wedge (p \vee \neg p)$

Properties of implication graph

Consider a formula F and its implication graph (V, E).

Theorem 10.2

If there is a path from ℓ_1 to ℓ_2 in (V, E) then there is a path from $\bar{\ell_2}$ to $\overline{\ell_1}$.

Exercise 10.4

a. Prove the above theorem.

b. Does the above theorem imply
if there is a path from p to $\neg p$ in (V, E) then there is a path from $\neg p$ to p?

Theorem 10.3

For every strongly connected component(scc) $S \subseteq V$ in (V, E), there is another scc S^c, called complementary component, that has exactly the literals that are negation of the literals in S.

Proof.

Due to theorem 10.2. □

Properties of implication graph (contd.)

Theorem 10.4

For each $m \models F$, if there is a path from ℓ_1 to ℓ_2 in (V, E) then if $m(\ell_1) = 1$ then $m(\ell_2) = 1$.

Theorem 10.5

For each $m \models F$ and each scc S in (V, E), either

- $m(\ell) = 1$ *for each* $\ell \in S$, *or*
- $m(\ell) = 0$ *for each* $\ell \in S$.

Exercise 10.5

Prove the above theorems.

(use theorem in the previous slide)

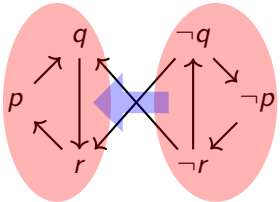
Reduced implication graph

Definition 10.4

For an implication graph (V, E), the reduced implication DAG (V^R, E^R) is defined as follows.

$V^R \triangleq \{S | S \text{ is a scc in } (V,E)\}$ $\quad E^R \triangleq \{(S, S') | \text{there are } \ell \in S \text{ and } \ell' \in S' \text{ such that } (\ell, \ell') \in E\}$

Example 10.4



Theorem 10.6

If $(S, S') \in E^R$ then $(S'^c, S^c) \in E^R$.

Exercise 10.6

Prove the above theorem.

Commentary: (V^R, E^R) is a graph over scc's of (V, E). Please notice that (V^R, E^R) will always be a directed acyclic graph (DAG).

2-SAT satisfiablity

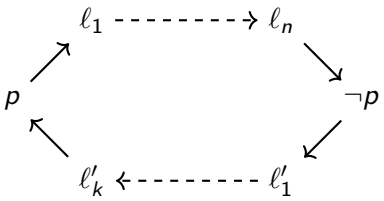
Theorem 10.7

A 2-SAT formula F is unsat iff there is a scc S in its implication graph (V, E) such that $\{p, \neg p\} \subseteq S$ for some p.

Proof.

Reverse direction

We assume $\{p, \neg p\} \subseteq S$.



There is a path that goes from p to $\neg p$.
Therefore, if p is true then $\neg p$ is true.
Therefore, p must be false.

Due to the path from $\neg p$ to p, if p is false then $\neg p$ is false.
Therefore, p must is true.

Therefore, F is unsat. ...

2-SAT satisfiablity(contd.)

Proof(contd.)

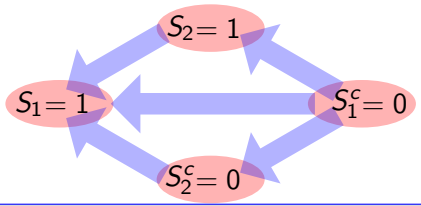
Fwd direction: Let us assume there is no such S.

We will construct a model of F as follows.

1. Initially all literals are unassigned.
2. While(some scc in V^R is unassigned)
   2.1 Let $S \in V^R$ be an unassigned scc such that all children of S are already assigned 1.
   2.2 Assign literals of S to 1. Consequently, S^c is assigned 0.

...

Example 10.5



2-SAT satisfiablity(contd.)

Proof(contd.)

We need to show that step 2.1 always finds S with all children assigned 1.

claim: at step 2.1, there is an unassigned node whose all children are assigned

Choose an unassigned node.

Descend down if there is an unassigned child.

Since the DAG is finite, the process will terminate.

claim: an unassigned node can not have a child that is assigned 0.

If S is assigned 1, all its children are already 1.

Therefore, all the parents of S^c are already assigned 0(due to theorem 10.6).

Therefore, no node with 0 model has an unassigned parent. □

Exercise 10.7

a. Show that the procedure produces a satisfying model.

b. Where did we use the fact that $\{p, \neg p\} \not\subseteq S$?

2-SAT is polynomial

Theorem 10.8

A 2-SAT satisfiability problem can be solved in linear time.

Proof.

Due to the previous theorem. □

Topic 10.3

Problems

Exercise: 2-SAT solving

Exercise 10.8

Find a satisfying model of the following formula

1. $(\neg x \vee \neg y) \wedge (\neg y \vee \neg z) \wedge (\neg z \vee \neg x) \wedge (x \vee \neg w) \wedge (y \vee \neg w) \wedge (z \vee \neg w)$
2. $(p_0 \vee p_2) \wedge (p_0 \vee \neg p_3) \wedge (p_1 \vee \neg p_3) \wedge (p_1 \vee \neg p_4) \wedge (p_2 \vee \neg p_4) \wedge$
   $(p_0 \vee \neg p_5) \wedge (p_1 \vee \neg p_5) \wedge (p_2 \vee \neg p_5) \wedge (p_3 \vee p_6) \wedge (p_4 \vee p_6) \wedge (p_5 \vee p_6)$

Unsat 2-SAT**

Exercise 10.9

Let us suppose we have n variables in a 2-CNF problem. What is the maximum number of clauses in the formula such that the formula is satisfiable?

Unsatisfiable core of 2-SAT

Exercise 10.10

An unsatisfiable core of an unsatisfiable CNF formula is a (preferably minimal) subset of the formula that is also unsatisfiable. Give an algorithm to compute a minimal unsatisfiable core of 2-SAT formula.

Topic 10.4

Extra slides: Horn clauses

Horn clauses

Definition 10.5

A Horn clause is a clause that has the following form

$$\neg p_1 \vee \cdots \vee \neg p_n \vee q,$$

where $p_1, \ldots, p_n \in \mathbf{Vars}$*, and* $q \in \mathbf{Vars} \cup \{\bot\}$.
A Horn formula is a set of Horn clauses, which is a conjunction of the Horn clauses.
The clauses with $\bot$ *literals are called goal clauses and others are called implication clauses.*

Example 10.6

The following set is a Horn formula

$$\{p, \quad \neg q \vee \neg r \vee \neg t \vee p, \quad \neg p \vee q, \quad \neg p \vee \neg r \vee t, \quad \neg p \vee \neg q \vee t, \quad \neg r \vee \bot, \quad \neg p \vee \neg q \vee \neg t \vee \bot\}$$

Implication view of the Horn clauses

We may view a Horn clause

$$\neg p_1 \vee \cdots \vee \neg p_n \vee q$$

as

$$p_1 \wedge \cdots \wedge p_n \Rightarrow q.$$

Example 10.7

The following is an implication view of the Horn formula from previous slide.

$$\{\top \Rightarrow p, \quad q \wedge r \wedge t \Rightarrow p, \quad p \Rightarrow q, \quad p \wedge r \Rightarrow t, \quad p \wedge q \Rightarrow t, \quad r \Rightarrow \bot, \quad p \wedge q \wedge t \Rightarrow \bot\}$$

Note $\top \Rightarrow p$ means p, which is a Horn clause without negative literals

Horn satisfiability

Algorithm 10.1: HORNSAT(Hs,Gs)

Input: *Hs*: implication clauses, *Gs* : goal clauses
Output: model/unsat
$m := \lambda x.0$;
while $m \not\models (p_1 \wedge .. \wedge p_n \Rightarrow p) \in Hs$ **do**
  $m := m[p \mapsto 1]$;
if $m \not\models (q_1 \wedge .. \wedge q_k \Rightarrow \bot) \in Gs$ **then return** *unsat* ;
return m

Exercise 10.11

Solve $\{\top \Rightarrow p, \quad q \wedge r \wedge t \Rightarrow p, \quad p \Rightarrow q, \quad p \wedge r \Rightarrow t, \quad p \wedge q \Rightarrow t, \quad r \Rightarrow \bot, \quad p \wedge q \wedge t \Rightarrow \bot\}$

Exercise 10.12

What is the maximum number of times the truth value of a clause in Hs changes during the algorithm? Give a supporting argument for the answer and an example that exhibits the situation.

Exercise: Horn SAT true to false

Exercise 10.13

In the Horn solving algorithm, we started with all false model and incrementally turned the variables true.

a. Is it possible to modify the algorithm such that it starts with all true model and finds satisfying model for the Horn clauses?

b. Can we also start with any initial model?

Exercise : wrong Horn satisfiability

Exercise 10.14

Here is another slightly different version of HornSAT. *Give an input for which the following program will give a wrong answer.*

Algorithm 10.2: HornSAT(Hs,Gs)

Input: *Hs*: implication clauses, *Gs* : goal clauses
Output: model/unsat
$m := \lambda x.0$;
while $m \not\models (p_1 \wedge .. \wedge p_n \Rightarrow p) \in Hs$ **do**
    $m := m[p \mapsto 1]$;
    if $m \not\models (q_1 \wedge .. \wedge q_k \Rightarrow \bot) \in Gs$ **then return** *unsat* ;
return m

Topic 10.5

Extra lecture slides : recognizable Horn clauses

Recognizing Horn clauses

Sometimes a set of clauses are not immediately recognizable as Horn clause.

We may convert a CNF into a Horn formula by flipping the negation signs for some variables. Such CNF are called Horn clause renamable.

Definition 10.6

Let F be a CNF formula and m be a model. Let $flip(F, m)$ denote the formula obtained by flipping the variables that are assigned 1 in m.

Example 10.8

$flip((p \vee \neg q \vee \neg s), \{p \mapsto 1, q \mapsto 0, s \mapsto 1, ..\}) = (\neg p \vee \neg q \vee s)$

Exercise 10.15

Calculate $flip((\neg p \vee q \vee \neg s), \{p \mapsto 1, q \mapsto 1, s \mapsto 0, ..\})$

Renaming Horn clauses

Theorem 10.9

A CNF formula $F = \{C_1, \ldots, C_n\}$, where $C_i = \{\ell_{i1}, \ldots, \ell_{i|C_i|}\}$ is Horn clause renamable iff the following 2-SAT formula is satisfiable.

$$G = \{\ell_{ij} \vee \ell_{ik} | i \in 1..n \text{ and } 1 \leq j < k \leq |C_i|\}$$

Proof.

Forward direction: there is a model m such that $flip(F, m)$ is a Horn formula

claim: $m \models G$

consider a clause $\ell_{ij} \vee \ell_{ik} \in G$

case $\ell_{ij} = p, \ell_{ik} = q$: one of them must flip,i.e.,$m(p) = 1$ or $m(q) = 1$

case $\ell_{ij} = \neg p, \ell_{ik} = \neg q$: at least one must not flip, i.e., not $m(p) = m(q) = 1$

case $\ell_{ij} = \neg p, \ell_{ik} = q$: if p flips then q must,i.e., if $m(p) = 1$ then $m(q) = 1$

In all the three cases $m \models \ell_{ij} \vee \ell_{ik}$.

Commentary: Proof source : H. Lewis. Renaming a Set of Clauses as a Horn Set. J. of the ACM, 25:134-135, 1978.

Renaming Horn clauses(contd.)

Proof(contd.)

Reverse direction: Let $m \models G$. Let $F' = flip(F, m)$.

claim: F' is a Horn formula

Suppose F' is not a Horn formula.

Then, there are positive literals ℓ'_{ij} and ℓ'_{ik} in clause C_i in F'.

Therefore, $m \not\models \ell_{ij} \vee \ell_{ik}$ (why?). Contradiction. □

Exercise 10.16

What is the complexity of checking if a formula is Horn clause renameable?

Exercise 10.17

Can you improve the above complexity?

Topic 10.6

Extra lecture slides: XOR SAT

XOR-SAT

Definition 10.7

A formula is XOR-SAT if it is a conjunction of xors of literals.

Example 10.9

$(p \oplus r \oplus s) \wedge (q \oplus \neg r \oplus s) \wedge (p \oplus q \oplus \neg s) \wedge (p \oplus \neg q \oplus \neg r)$ *is a XOR-SAT formula.*

Solving XOR-SAT

Since xors are negation of equality, we may eliminate variables via substitution.

Theorem 10.10
For a variable p, formula G, and formula F, if $p \notin Vars(G)$ then $(p \oplus G) \wedge F$ and $F[\neg G/p]$ are equivsatisfiable.

Exercise 10.18
Prove the above theorem.

Commentary: The substitution process is similar to the Gaussian elimination used for solving linear equations.

Example : solving XOR-SAT

Example 10.10

Consider $(p \oplus r \oplus s) \wedge (q \oplus \neg r \oplus s) \wedge (p \oplus q \oplus \neg s) \wedge (p \oplus \neg q \oplus \neg r)$

Eliminate p*:*
Due to the first xor: $p \Leftrightarrow \neg r \oplus s$
Substitution: $(q \oplus \neg r \oplus s) \wedge (\neg r \oplus s \oplus q \oplus \neg s) \wedge (\neg r \oplus s \oplus \neg q \oplus \neg r)$
Simplification: $(q \oplus \neg r \oplus s) \wedge (\neg r \oplus \neg q) \wedge (s \oplus \neg q)$

Eliminate r*:*
Due to the second xor: $r \Leftrightarrow \neg q$
Substitution: $(q \oplus \neg\neg q \oplus s) \wedge (s \oplus \neg q)$
Simplification: $s \wedge (s \oplus \neg q)$

Example: solving XOR-SAT (contd.)

Eliminate q:
Due to the second xor: $q \Leftrightarrow s$
After substitution: s

Solution: $m(s) = 1$ $m(q) = m(s) = 1$ $m(r) = m(\neg q) = 0$ $m(p) = m(\neg r \oplus s) = 0$

Exercise 10.19

Find a satisfying model of the following formula

- $(p \oplus r \oplus s) \wedge (q \oplus r \oplus s) \wedge (\neg p \oplus q \oplus \neg s) \wedge (p \oplus \neg q \oplus \neg r)$

Commentary: We construct the model by following the substitution expressions backwards, when we assign a value to q, r, and p.

Unsat XOR-sat

Exercise 10.20

Give an unsat XOR-sat formula that has only xors with more than three arguments.

Attendance quiz

Exercise 10.21

Which of the following are unsatisfiable formulas?

$(p \oplus q) \wedge (p \oplus \neg r) \wedge (q \oplus \neg r)$
$(p \oplus q) \wedge (\neg p \oplus \neg r) \wedge (q \oplus r)$
$(p \oplus q) \wedge (p \oplus \neg r) \wedge (q \oplus r)$
$(p \oplus q) \wedge (\neg p \oplus \neg r) \wedge (q \oplus \neg r)$

$(p \oplus \neg r) \wedge (p \oplus q) \wedge (q \oplus \neg r)$
$(p \oplus q) \wedge (q \oplus r) \wedge (\neg p \oplus \neg r)$
$(p \oplus q) \wedge (q \oplus r) \wedge (p \oplus \neg r)$
$(q \oplus \neg r) \wedge (p \oplus q) \wedge (\neg p \oplus \neg r)$

$(q \oplus \neg r) \wedge (p \oplus q) \wedge (p \oplus \neg r)$
$(\neg p \oplus \neg r) \wedge (q \oplus r) \wedge (p \oplus q)$
$(p \oplus \neg r) \wedge (q \oplus r) \wedge (p \oplus q)$
$(\neg p \oplus \neg r) \wedge (q \oplus \neg r) \wedge (p \oplus q)$

End of Lecture 10