Overview

Positive and negative clauses

A clause C is

- positive, if all literals in C are positive,
- negative, if all literals in C are negative,

Property

Every unsatisfiable formula F contains a positive and a negative clause.

Proof: Otherwise the assignments $\alpha \equiv 0$ or $\alpha \equiv 1$ satisfy F.

Horn formulas

A clause C is

- definite, if exactly one literal in C is positive,
- a Horn clause, if at most one literal in C is positive.

Thus, a Horn clause is either negative or definite.

A Horn-formula is a conjunction of Horn clauses.

Corollary

Every unsatisfiable Horn formula contains a positive unit clause.

Algorithm for Horn formulas

Theorem

Horn-SAT can be decided in time $O(nm)$.

Algorithm:

$\alpha := []$

`while positive unit clause` x `in` $F\alpha$

  $\alpha := \alpha \cup [x := 1]$

$\alpha' := \alpha \cup \left[y := 0\,; y \notin \mathrm{dom}\,\alpha\right]$

`if` $\alpha' \models F$

  `then return` α'

  `else return UNSAT`

2-SAT as a graph

For a 2-CNF formula F, define the directed graph $G(F)$:

- vertices are the literals of F
- (a, b) is an edge if $\bar{a} \vee b$ is a clause in F.
- $(\bar{a}, a)$ is an edge if a is a unit clause in F.

Lemma

If $\alpha \models F$, and b is reachable from a in $G(F)$, then $\alpha(a) = 1$ implies $\alpha(b) = 1$.

Let $[a]$ denote the strongly connected component of a in $G(F)$.

Corollary

If $\alpha \models F$, and $[a] = [b]$, then $\alpha(a) = \alpha(b)$.

Algorithm for 2-SAT

Theorem

F is unsatisfiable iff $[x] = [\bar{x}]$ for some $x \in V(F)$.

Algorithm to compute $\alpha \models F$ if $[x] \neq [\bar{x}]$ holds for all $x \in V(F)$:

Let $[a_1], \ldots, [a_r]$ be the SCCs in reverse topological order:

```
for j := 1 to r do
   if the literals in [a_j] are unassigned
      α(b) := 1 for all b ∈ [a_j]
      α(b) := 0 for all b ∈ [ā_j]
```

Corollary

2-SAT can be decided in linear time,
and in nondeterministic logarithmic space.

CNF(2) as a graph

For F in CNF(2), define undirected, marked (multi-)graph $G(F)$:

- vertex v_C for every clause C in F
- there is an edge e_x between v_C and v_D if $x \in C$ and $\bar{x} \in D$.
- v_C is marked if C contains a pure literal.

Assignment $\quad \triangleq \quad$ orientation of the edges

Clause C is satisfied $\quad \triangleq \quad$ v_C is marked or of outdegree > 0

SAT(2) is tractable

Lemma

F is satisfiable iff $G(F)$ can be oriented s.t. every unmarked vertex has non-zero out-degree.

A connected component is marked if it contains a marked vertex.

Theorem

F is satisfiable iff every unmarked connected component in $G(F)$ has a cycle.

Corollary

SAT(2) can be decided in linear time,
and in deterministic logarithmic space.

Horn-renamable formulas

A renaming is a permutation r on literals with $r(a) \in \{a, \bar{a}\}$.

Formula F is Horn-renamable if there is a renaming r such that $r(F)$ is a Horn formula.

Theorem

There is a linear time algorithm to test whether F is Horn-renamable and if yes, computes a renaming r s.t. $r(F)$ is a Horn formula.

Theorem

SAT for Horn-renamable formulas can be solved in linear time.

Cluster formulas

Two clauses C, D in a formula F clash, if $a \in C$ and $\bar{a} \in D$ for some literal a.

F is a hitting formula if any two clauses in F clash.

Theorem

A hitting formula F is satisfiable iff $\sum_{C \in F} 2^{-w(C)} < 1$.

A cluster formula is a union $\bigcup_{i=1}^{t} F_i$, where each F_i is a hitting formula and $V(F_i) \cap V(F_j) = \emptyset$ for $i \neq j$.

Corollary

Satisfiability of cluster formulas can be tested in polynomial time.