variations of SAT

I looked up on the internet, but I could not find any 'big-list' of variants of SAT problem.

Apart from the (common)

- SAT,
- k-SAT,
- MAX-kSAT,
- Half-SAT,
- XOR-SAT,
- NAE-SAT

what else variants are there?

(also it will be really useful if there complexity classes are given (where possible))

cc.complexity-theory sat big-list

asked Sep 7 '13 at 3:59
Subhayan
501 5 17

What would be the purpose of this list? – Tyson Williams Sep 7 '13 at 12:50

2 Firstly because I wanted to present a talk to some undergraduate students. I was planning to talk about the variations of SAT and show some (non-trivial) reductions... they have already had a introductory course in TOC, so i thought this might be a good idea .. AND THE SECOND REASON being the fact there is no such list on the internet, this list will also serve any curious mind who wants to know about the variants. – Subhayan Sep 7 '13 at 13:23

11 I am not sure how this list will help with your talk. Instead of reading an arbitrarily long list of SAT variants, a curious mind should read Schaefer's dichotomy theorem and the generalization by Allender et al. that shows that every possible SAT variant is complete for one of six well-known complexity classes. – Tyson Williams Sep 7 '13 at 13:57

that's a nice suggestion... thanks @TysonWilliams .. you can make that into an answer too, although that's not exactly what I was looking for, but surely this is helpful. – Subhayan Sep 7 '13 at 14:21

6 Answers

On the "NP-complete side" I came across these variants (I asked a similar question on cs.stackexchange, too):

- Planar 3-SAT;
- Planar 1-in-3 SAT and Positive Planar 1-in-3 SAT (see Mulzer and Rote);
- NAE 3-SAT;
- 2/2/4-SAT (see Finding a shortest solution for the NxN extension of the 15-Puzzle is intractable by D. Ratner and M. Warmuth (1986));
- XSAT, NAE-SAT for Linear Monotone Formulas, XSAT for exact linear formulas (see the nice work of Ewald Speckenmeyer on linear XSAT problems: Computational Complexity of SAT, XSAT and NAE-SAT for linear and mixed Horn CNF formulas);
- k-colourable Monotone NAE-3SAT (see P Jain, On a variant of Monotone NAE-3SAT and the Triangle-Free Cut problem, 2010).

edited Sep 7 '13 at 15:09 answered Sep 7 '13 at 14:54
Marzio De Biasi
16.8k 2 34 97

Apart from the list above, there are also:

- #SAT: model counting
- All-SAT: model enumerating

answered Sep 7 '13 at 22:01
qsp
375 2 10

(Making comment an answer as requested and expanding a bit.)

"A curious mind" should read Schaefer's dichotomy theorem and the generalization by Allender et al. that shows that every possible SAT variant is either trivial or in one of six well-known complexity classes:

1. NP-complete
2. P-complete
3. NL-complete
4. L-complete
5. $\oplus$L-complete
6. co-NLOGTIME

edited Sep 8 '13 at 4:10

answered Sep 7 '13 at 22:10
Tyson Williams
3,417 2 16 41

Let $\mathrm{SAT}(k)$ be SAT for CNF formulas with at most k occurrences of every variable. Then $\mathrm{SAT}(2)$ ist complete for L (i.e., deterministic log space, see my paper in STACS 2004), whereas $\mathrm{SAT}(k)$ for $k \geq 3$ is NP-complete.

answered Sep 9 '13 at 8:00
Jan Johannsen
3,547 1 21 43

This list will be very long;) Here are some of my favourite (NP-complete) variants of SAT:

- PLANAR($\leq 3, 3$)-SAT (each clause contains at least two and at most three literals, each variable appears in exactly three clauses; twice in its non-negated form, and once in its negated form, and the bipartite incidence graph is planar.)

  See: Dahlhaus, Johnson, Papadimitriou, Seymour, Yannakakis, The complexity of multiterminal cuts, SIAM Journal of Computing 23 (1994) 864-894
- 4-BOUNDED PLANAR 3-CONNECTED 3SAT (every clause contains exactly 3 distinct variables, every variable appears in at most 4 clauses, the bipatite incident graph is planar and 3-connected)

  See: Kratochvíl, A special planar satisfiability problem and a consequence of its NP-completeness, Discrete Applied Math. 52 (1994) 233-252
- MONOTONE CUBIC 1-IN-3SAT (MONOTONE-1-IN-3SAT in which every variable appears exactly 3 times)

  See: Moore and Robsen, Hard tilings problem with simple tiles, Discrete Compute. Geom. 26 (2001) 573-590
- I find very interesting that PLANAR-NAE-kSAT is in P for every k.

  See this post.

edited Apr 13 at 12:32
Community ♦
1

answered Sep 9 '13 at 16:04
user13136
2,112 1 15 22

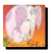
4 If you find the last point interesting, you might also be interested to know that #PLANAR-NAE-3SAT (counting solutions) is tractable as well, whereas other seemingly simple SAT variants like PLANAR-MONOTONE-2SAT are tractable (or even trivial) as a decision problem, but #P-hard for counting. Note that the reduction from the last link above (reducing PLANAR-NAE-kSAT to PLANAR-NAE-3SAT) is not parsimonious, and that #PLANAR-NAE-4SAT is #P-hard. – William Whistler Oct 3 '13 at 14:32

There is a very classic connection between logic and algebra, which goes back to the origin of modern logic and the work of George Boole. A formula in propositional logic can be interpreted as an element of a Boolean algebra. The logical constants *true* and *false* become the algebraic notions of the top and bottom element of a lattice. The logical operations of conjunction, disjunction and negation will become the algebraic operations of meet, join and complementation in the Boolean algebra. This connection is less emphasised in modern treatments of logic, but it is particularly interesting in the context of your question. Algebra allows us to move away from many problem specific details and find generalisations of a problem that will apply to many different situations.

In the specific case of SAT, the algebraic question one may ask is what happens when we interpret formulae in more general lattices than Boolean algebras. On the logical side, you can generalise the satisfiability problem from propositional logic to intuitionistic logic. More generally, you can generalise the propositional satisfiability problem to that of determining if a formula, when interpreted over a bounded lattice (one with top and botto), defines the bottom element of the lattice. This generalisation allows you to treat problems in program analysis as satisfiability problems.

Another generalisation is to quantifier-free first-order logic where you get the question of Satisfiability Modulo a Theory. Meaning, in addition to having Boolean variables, you also have first-order variables and function symbols and you want to know if a formula is satisfiable. At this point you can ask questions about formulae in arithmetic, theories of strings, or arrays, etc. So we get a strict and very useful generalisation of SAT which has lots of applications in systems, computer security, programming languages, program verification, planning, artificial intelligence, etc.

answered Sep 16 '13 at 16:43



Vijay D
10.7k 2 27 55