Combinatorics and SAT

Oliver Kullmann

Computer Science Department
University of Wales Swansea
Swansea, SA2 8PP, UK

e-mail: O.Kullmann@Swansea.ac.uk
http://cs-svr1.swan.ac.uk/~csoliver/

March 29, 2005

CNF's as mediators between Logic,
Constraint Satisfaction and Combinatorics.

"Conjunctive Normal Forms"

Some say, it must be "propositional formulas **in** conjunctive normal form". An example is

$$F = (a \vee \neg b \vee c) \wedge (\neg a \vee \neg c) \wedge (b \vee d).$$

Embedded into the richer logic context, the theory of "propositional formulas in conjunctive normal form" inherits

- semantics
- proof theory
- complexity theory.

In the SAT community, people just speak of "CNF's" or "conjunctive normal forms":

- If you are using these terms often, a shorter expression is needed.
- And CNF's become emancipated — they want their own living.

What are CNF's ?!

For many SAT people (escpecially if they are coming from an engineering background) CNF's are merely convenient **data structures**.

CNF's are represented as **clause-sets**; the above example becomes

$$F = \{\, \{a, \overline{b}, c\}, \{\overline{a}, \overline{c}\}, \{b, d\} \,\}.$$

We got rid off the logic — a good thing and a bad thing.

Let's go further:

$$\{\, \{a, b, c\}, \{a, c\}, \{b, d\} \,\}.$$

Sure, something essential was lost, but we see some structure emerging: **hypergraphs**.

This is the path I want to follow in this talk: CNF's aka clause-sets as generalised hypergraphs.

Generalising clause-sets

Actually, I don't want to speak about *boolean* clause-sets, but about *generalised* clause-sets.

This shall make the connection to combinatorics (and constraint satisfaction problems) more direct.

What are clause-sets *essentially* ?!?!

In order not to loose contact with hypergraphs, we assume that we are having **variables**.

Let the set of variables be $\boldsymbol{\mathcal{VA}}$. Each variable $v \in \mathcal{VA}$ has a **domain** $\boldsymbol{D_v}$, a non-empty set.

A **total assignment** is a choice function for $(D_v)_{v \in \mathcal{VA}}$, that is, a map f with $\mathrm{dom}(f) = \mathcal{VA}$ such that $\forall\, v \in \mathcal{VA} : f(v) \in D_v$.

To each "set of clauses" F we assign a set $\boldsymbol{\mathfrak{F}(F)} \subseteq \mathcal{TASS}(\mathcal{VA})$ of **falsifying assignments**. F is satisfiable if $\mathfrak{F}(F) \neq \mathcal{TASS}(\mathcal{VA})$, otherwise **unsatisfiable**.

(Every total assignments either **satisfies** or **falsifies** F. We search for satisfying assignments, but therefore falsifying assignments are much more "common".)

From sets of clauses to literals

F is a set of "clauses". To each clause C we also assign a set $\mathfrak{F}(C) \subseteq \mathcal{TASS}(\mathcal{VA})$ of falsifying assignments (all other assignments are satisfying), and we set

$$\mathfrak{F}(F) = \bigcup_{C \in F} \mathfrak{F}(C),$$

since we think of set of clauses as **conjunctions** (or as sets of "constraints").

A clause is a set of "literals". To each literal x we assign a set $\mathfrak{F}(x) \subseteq \mathcal{TASS}(\mathcal{VA})$ of falsifying assignments (all other assignments are satisfying), and we set

$$\mathfrak{F}(C) = \bigcap_{x \in C} \mathfrak{F}(x),$$

since we think of clauses as **disjunctions**.

So what is left is to specify what are "literals" x, and what is $\mathfrak{F}(x)$.

Literals

We want a literal to depend exactly on one variable. That is, for every literal x there should be a (unique) variable $\mathrm{var}(x) \in \mathcal{VA}$ such that $\mathfrak{F}(x)$ restricts only assignments for $\mathrm{var}(x)$.

So we consider literals as pairs (v, N) with $N \subseteq D_v$ denoting the set of **forbidden values**; thus

$$\mathfrak{F}(x) = \{f \in \mathcal{TASS}(\mathcal{VA}) : f(v) \in N\}.$$

What shall we allow for N ?! Alan Frisch (1999, "NB-resolution") considered (essentially) all possible $N \in \mathbb{P}(D_v)$, but here we don't go that far.

For us the **correspondence between partial assignments and clauses** is essential: Given a partial assignment φ (a restriction of some total assignment to some domain $V \subseteq \mathcal{VA}$) we want, that there exists exactly one maximal clause C such that C is falsified by all total assignments extending φ.

It follows that we restrict our attention to the case $|N| = 1$.

Some further conditions

So literals are pairs (v, ε) with $v \in \mathcal{VA}$, while $\varepsilon \in D_v$ is the "forbidden" value. We use $\mathrm{val}((v, \varepsilon)) = \varepsilon$.

For clauses C we do not allow literals $x, y \in C$ with $\mathrm{var}(x) = \mathrm{var}(y)$ but $\mathrm{val}(x) \neq \mathrm{val}(y)$, since this would break the 1-1-correspondence to partial assignment (a clause like this would be "tautological", i.e., would be satisfied by every total assignment).

In order to have compactness (i.e., if a set of clause F is unsatisfiable then there exists a finite subset $F' \subseteq F$ which is already unsatisfiable) we allow only **finite domains** and **finite clauses**.

A finite set of clauses is called **clause-set**.

A boolean variable is a variable $v \in \mathcal{VA}$ with $D_v = \{0, 1\}$. A **boolean clause-set** is a clause-set F where every $v \in \mathrm{var}(F)$ is boolean.

Hypergraph colouring

A **hypergraph** is a pair (V, E), where V is the set of vertices, and E is the set of **hyperedges**, which are finite sets of vertices.

A C-colouring of a hypergraph G is a map $f : V \to C$ such that for all (hyper)edges $H \in E(G)$ there are $v, w \in H$ with $f(v) \neq f(w)$ (that is, there is no monochromatic edge). G is called k-**colourable** for $k \in \mathbb{N}_0$ if G is $\{1, \ldots, k\}$-colourable.

The hypergraph colouring problems generalises the graph colouring problem.

For example the **Ramsey-number problem** as well as the **Van-der-Waerden-number problem** are most naturally cast as hypergraph colouring problems.

Translating hypergraph colouring problems

Consider a hypergraph G and $k \in \mathbb{N}$.

We want to define a set of clauses $\boldsymbol{F_k(G)}$ such that $F_k(G)$ is satisfiable iff G is k-colourable, and furthermore the satisfying assignments for $F_k(G)$ correspond to the k-colourings of G.

We consider the vertices $v \in V(G)$ as the variables of $F_k(G)$ with $D_v = \{1, \dots, k\}$. For every edge $H \in E(G)$ and $\varepsilon \in \{1, \dots, k\}$ let the clause H_ε be defined as

$$\boldsymbol{H_\varepsilon} := \{(v, \varepsilon) : v \in H\}.$$

We see that clause H_ε expresses the condition that not all vertices in H get colour ε. So let

$$\boldsymbol{F_k(G)} := \left\{\, H_\varepsilon : H \in E(G) \wedge \varepsilon \in \{1, \dots, k\} \,\right\}$$

We see that **hypergraph 2-colouring problems** are *directly* translated into satisfiability problems for *boolean* clause-sets.

A theorem of Seymour

Seymour (1974) showed:

For a minimally non-2-colourable hypergraph G we have $|E(G)| \geq |V(G)|$.

It is natural to expect, that this property should hold for every non-k-colourable hypergraph G ($k \geq 2$). How to prove this? We are facing two difficulties:

1. The proof is based on linear algebra, and works only for $k = 2$.
2. From G being minimally non-k-colourable there seems no way to get down to being minimally non-k'-colourable for some $k' < k$.

Perhaps considering $F_k(G)$ helps?

There is a similar-sounding assertion for boolean clause-sets: For a minimally unsatisfiable boolean clause-set F we have $c(F) \geq n(F) + 1$, where $c(F)$ is the number of clauses of F, and $n(F)$ is the number of variables.

Autarkies

There are several proofs of

For a minimally unsatisfiable boolean clause-set F we have $c(F) \geq n(F) + 1$.

The earliest proof (Aharoni and Linial 1986) is based on **matching theory**. I think **autarky theory** gives the most natural environment for this approach:

An autarky for a (generalised) clause-set F is a partial assignment φ such that every clause of F "touched" by φ is satisfied by φ.

If φ is an autarky for F then the clauses of F touched by φ can be removed satisfiability-equivalently.

Deciding whether a clause-set has a non-trivial autarky is NP-complete.

However for the special class of **matching autarkies** the (confluent) reduction process by applying matching autarkies can be performed in polynomial time, and for a **matching-lean** clause-set F (having no non-trivial matching autarky) we have

$$\forall\, F' \subset F : \delta(F') < \delta(F),$$

where the **deficiency $\boldsymbol{\delta(F)}$** of a (generalised) clause-set F is zero for the empty clause-set, and thus for matching-lean clause-sets it follows $\delta(F) \geq 1$.
For a boolean clause-set F we have

$$\delta(F) = c(F) - n(F).$$

Since every minimally unsatisfiable clause-set is (matching) lean, it follows $c(F) \geq n(F) + 1$ for minimally unsatisfiable boolean clause-sets.

Using a different type of autarky, **balanced linear autarky**, we can show for boolean clause-sets F which are lean w.r.t. balanced linear autarkies

$$\forall\, F' \subset F : \delta(F') \leq \delta(F),$$

and thus $\delta(F) \geq 0$. From this we easily get Seymour's theorem (the case $k = 2$). But what about $k > 2$?!

How to generalise Seymour's theorem

Consider a hypergraph G and the clause-sets $F_k(G)$ for $k = 1, 2, 3, \ldots$.

For all $k < \chi(G)$, where $\chi(G)$ is the chromatic number of G, the $F_k(G)$ are unsatisfiable, while for all $k \geq \chi(G)$ the $F_k(G)$ are satisfiable.

"Minimal non-k-colourable" means that $F_k(G)$ is minimally unsatisfiable, while $F_{k+1}(G)$ is satisfiable. This is not transportable to some $k' < k$.

Now, as we have seen, fortunately assertions like Seymour's theorem or the theorem about the deficiency on minimally unsatisfiable clause-sets do not really depend on the condition of being minimally unsatisfiable, but on some **leanness condition**.

And if $F_k(G)$ is lean, then $F_{k'}(G)$ is lean for all $k' < k$!

So the generalisation of Seymour's theorem direclty follows. (Likely you noticed, that I left out some details, but I told you the main ideas.)

Conclusion

We presented a canonical embedding of hypergraph colouring problems into the space of generalised satisfiability problems.

This embedding into a richer space enables operations which are external to the hypergraph environment, and can be expressed at this level only in clumsy ways. (However, it seems essential that this richer space is still "close enough" to the original domain.)

One of these operations is the operation of autarkies. Autarky-freeness ("leanness") yields a natural and "smooth" extension of the notion of "minimallity".

We showed how to get some results in hypergraph theory. On the other hand, we can gain much from the rich intuition provided by hypergraph theory (and combinatorics in general).

END