Learning with unitary propagation

Olivier Bailleux, https://orcid.org/0000-0002-3433-7344
Research idea sheet n°1, january 2018
If you are interested in using the idea presented in this sheet, or if you think it has already been published or exploited, please contact me.

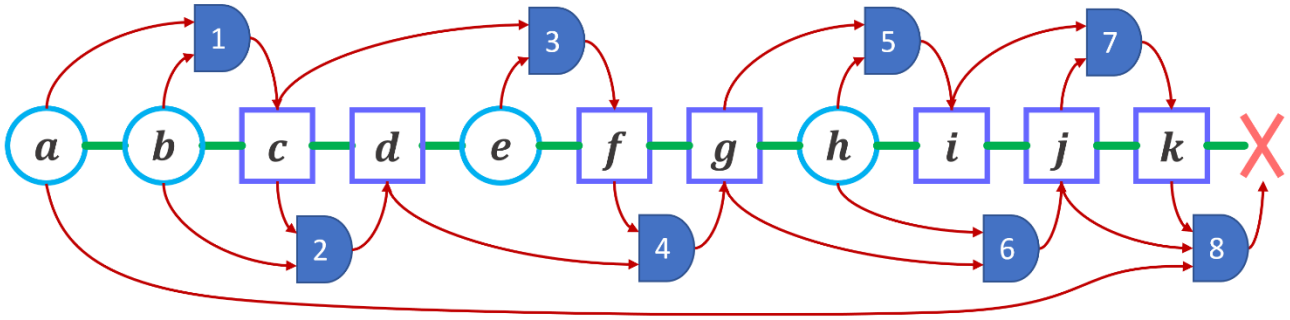
Abstract

DPLL and CDCL SAT solvers produce sequences of assumptions and deductions thanks to unitary propagation. In CDCL solvers, a bottom-up analysis of these sequences produces clauses that explain the conflicts. The idea proposed in this note is to analyze such sequences in chronological order of assumptions and deductions to produce other clauses than those produced by CDCL solvers.

AUP-sequences

DPLL and CDCL SAT solvers learn clauses thanks to unitary propagation, more precisely by producing sequences of assumptions and deductions leading to a contradiction. In this post, such sequences will be called AUP-contradictory sequences (AUP for Assumption and Unitary Propagation). Assumptions and deductions are literals.

Here is an example of a contradictory AUP-sequence that could be produced by a CDCL solver or a DPLL solver.



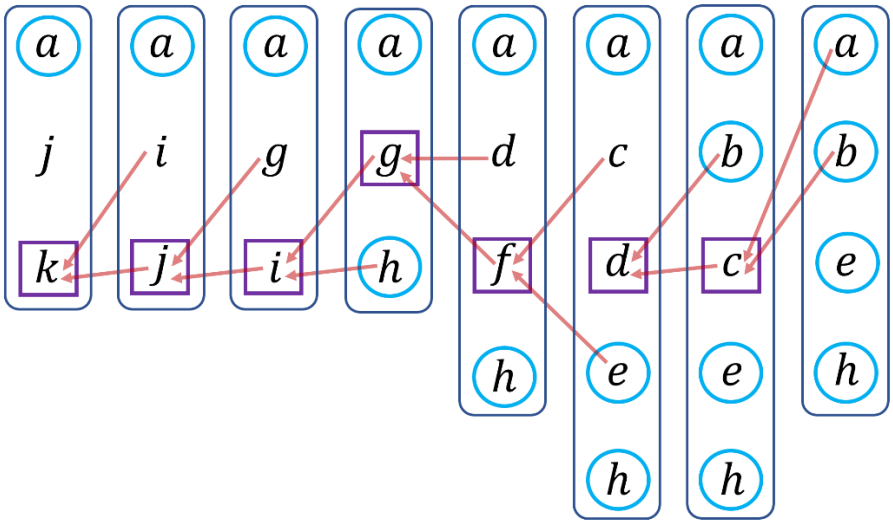
Circles represent assumptions and squares deductions. The clauses used for the deductions are numbered from 1 to 8.

DPLL and CDCL do not extract the same information from such a sequence. DPLL implicitly deducts the negation of the assumptions, while CDCL performs a bottom-up analysis of the sequence from the so-called *conflict clause*, which is the clause 8 in the example.

Bottom-up and to-down analysis

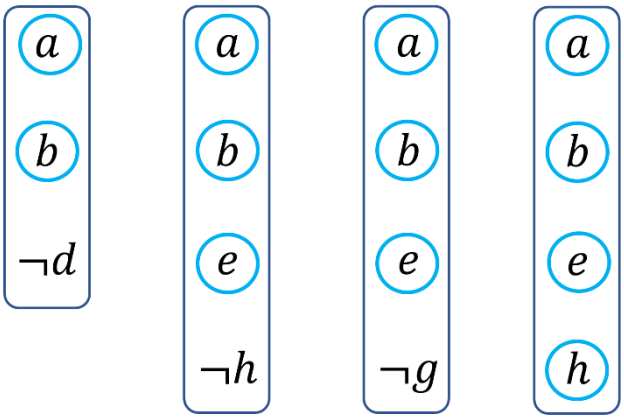
From the example sequence, DPLL *implicitly* produces the clause (¬ a ∨ ¬ b ∨ ¬ e ∨ ¬ h) while CDCL with first UIP learning scheme explicitly produces the clause (¬ a ∨ ¬ g ∨ ¬ h). But other clauses can be deducted. Each of these clauses is the negation of an *UP-nogood*, i.e. a set of literals allowing unit propagation to produce the empty clause.

The following nogoods can be produced by a bottom-up analysis "à la CDCL":



Each framed literal is replaced in the next nogood by those that allowed it to be deduced. The surrounded literals are assumptions of the AUP-sequence. The fourth nogood is the one that will be retained by a CDCL solver with first UIP learning scheme. The principle of bottom-up analysis has been the subject of numerous publications related to CDCL solvers and their learning schemes such as first UIP or last UIP, for example.

But it is possible to produce the following nogoods by a top-down analysis of the assumptions.



Top-down analysis produces UP-nogoods that are generally different from those produced by bottom-up analysis. To the best of my knowledge, it is rarely, if ever, mentioned in the literature. I will use the above example of AUP-sequence to illustrate the principle.

Since a and b allow unit propagation to deduce c, the set {a, b, ¬c} is an UP-nogood. But it is of no interest since its negation is clause 1, already known.

But a and b also allow to deduce d in two steps thanks to the clauses 1 and 2, so the set {a, b, ¬ d} is a UP-nogood whose negation is the clause (¬ a ∨ ¬ b ∨ d). Incidentally, this clause is the resolvent of clauses 1 and 2 as defined in General Resolution.

Similarly, assumptions a, b, and e allow the literals f and g to be deduced, so {a, b, e, ¬f} and {a, b, e, ¬g} are UP-nogoods that establish that the clauses (¬a ∨ ¬b ∨ ¬e ∨ f) and (¬a ∨ ¬b ∨ ¬e ∨ g) are logical consequences of the clauses that were used to achieve unit resolutions in the AUP- sequence.

The last nogood is the set of assumptions, and its negation therefore represents the result of an ad absurdum reasoning, which deduces the negation of the contradictory assumptions.

Exploiting top-down analysis

I propose to develop SAT solvers and other inference systems using top-down analysis of AUP-sequences. The theoretical interest of such a deductive system is that it immediately captures the power of general Resolution. For example, the Resolution of (a ∨ b ∨ x) and (c ∨ d ∨ ¬ x) produces the resolvent (a ∨ b ∨ c ∨ d). This resolvent can be produced by a top-down analysis of the AUP-sequence from the assumptions ¬ a, ¬ b and ¬ c, that will propagate d.

On the other hand, this system has the advantage of allowing deductions to be made from AUP-sequences that do not result in a contradiction or that are not extended to a contradiction. One could imagine a SAT solver with learning, in which AUP-sequences can be stopped before a conflict occurs, in a to way limit the size of the produced clauses.