

# Investigation of Binary Representations of SAT especially 2-Literal Representation<sup>3</sup>

Gábor Kusper

The problem of propositional SATisfiability for formulae in conjunctive normal form is the first known NP-complete problem. Many practical NP-hard problems may be transformed efficiently to SAT. Thus, a good SAT solver would likely have considerable utility. Since the complexity of SAT solvers depends also on the representation we shall carefully study this aspect. A formula is conjunction of clauses and a clause is disjunction of literals. There are three kind of literals: positive (+), negative (-) and joker or no occurrence (x) literal. The positive and negative literals are concrete literals. The joker literal means that the corresponding variable does not occur in the clause. There are many possibilities how to represent a formula. For example we may represent a formula as a list of clauses or matrix of literals. We discuss advantages and disadvantages of several SAT representations. First we discuss the natural binary representation, where a literal is represented by 2 bits (-:01, +:10, x:11). Then we introduce a new binary representation, called 2-literal representation, which uses 4 bits to represent all the 15 possible basic (basic means cannot be simplified) formulae with 2 variables ([--]:0001, [-+]:0010, [-x]:0011, [++]:0100, [--,+]:0101, [x+]:0110, [-x,x+]:0111, [+ -:1000, [x-]:1001, [-+,-]:1010, [-x,x-]:1011, [+x]:1100, [+x,x-]:1101, [+x,x+]:1110, [xx]:1111). The idea is that every bit corresponds to a 2-clause (a clause with two concrete literals). For example, the least significant bit corresponds to --. If a bit is 1 then the corresponding 2-clause is subsumed by the represented formula. The 2-literal representation is better than the natural one because with 4 bits it can represent 2 literals as the natural one and it can represent also any combinations of 2-clauses and many operations are easier with it. We can generalize the 2-literal representation, called  $n$ -literal representation, such that it uses  $2^n$  bits to represent all basic formulae with  $n$  variable. This representation is rather costly but we give arguments for using 5-literal representation. Finally we give a comparative table with 5 different SAT representations with time complexity information of operations like unit-propagation or formula negation.

---

<sup>3</sup>Sponsored by Upper Austrian Government (scholarship), FWF SFB F013 (P1302) and Austro-Hungarian Action Foundation