Literal Counting Scheme

- Each variable keeps a list of all its occurrence in the clauses, both in positive and negative form.
- Each clause maintains counters to indicate its status (number of 1/0/- assignments).
- When a variable is assigned, it will visit all the clauses that contain it and modify the status counters.
- When a variable is unassigned, it will also need to reverse the modification it did to the clauses.

Microsoft Research

Literal Counting as in GRASP

- Each clause maintains two counters:
  - Num_1_Lits
  - Num_0_Lits
  - Num_all_Lits (This is not a counter, just a constant)
- A Clause is unit if
  - (Num_0_Lits == Num_all_Lits – 1) && Num_1_Lits == 0
  - If this is true, solver needs to search through all the literals of the clause to find out the remaining free literal
- A Clause is a conflict clause if
  - Num_0_Lits == Num_all_Lits
- A SAT instance with n variables, m clauses, each clause has l literals on the average:
  - A variable assignment/unassignment takes *l m / n* operations on the average
- A Clause is SAT if
  - Num_1_Literals > 0
  - This is a constant time operation

A Better Literal Counting Scheme

- Each clause keeps one counter
  - Num_Non_Zero_Literals
- A Clause is Unit if
  - Num_Non_Zero_Literals == 1
  - And, the solver will search all the literals in the clause to find out the remaining literal with value other than 0, if it’s unassigned, then it is implied. Otherwise, skip this clause.
- A Clause is Conflict if
  - Num_Non_Zero_Literals == 0
- A SAT instance with n variables, m clauses, each clause has l literals on the average:
  - A variable assignment/unassignment takes $l\,m / 2n$ operations on the average
- A Clause is SAT if
  - Search all the literals in the clause to find if it has at least one of them with value 1.
  - This operation complexity is linear wrt the length of the clause