Solver name	akmaxsat, akmaxsat_ls
Author	Adrian Kügel

Description The solver uses a branch-and-bound approach. At each node of the search tree a lower bound on the minimum number (or minimum sum of weights for weighted formulas) of unsatisfied clauses is calculated. The method used to calculate the lower bounds is to extract inconsistent subformulas with generalized unit propagation [1]. Generalized unit propagation can be summarized as follows: Do unit propagation until no unit literal is left. Then search for a failed literal in the simplified formula. If found, continue unit propagation with the newly derived unit literal. These two steps are repeated until either no more failed literal can be found, or an inconsistent subformula is found. As a byproduct of the lower bound calculation the formula can be transformed into an equivalent simpler formula. The solver uses the same transformation rules as the WMaxsatz solver [2], and in addition to that the transformation rule $(x \vee y \vee z) \wedge (x \vee y \vee \overline{z}) \rightarrow (x \vee y)$. The solver uses a lazy deletion data structure which deletes lazily clause pointers from the literal occurrence lists to clauses which are currently fulfilled or deleted. For details how this works, see [1]. This data structure leads to speedups for clauses with a high clauses-to-variables ratio.

akmaxsat_ls uses ubcsat in order to calculate an initial upper bound on the optimal solution, whereas akmaxsat assumes an initial upper bound of infinity.

References

[1] A. Kügel. Improved exact solver for the weighted max-sat problem. To appear in Easychair electronic proceedings.

[2] C. M. Li, F. Manyà, N. O. Mohamedou, and J. Planes. Exploiting cycle structures in max-sat. In O. Kullmann, editor, *SAT*, volume 5584 of *Lecture Notes in Computer Science*, pages 467–480. Springer, 2009.