1 Competition: Autarkies (7(+7) Points)

Write an anytime algorithm that computes a maximal autarky for a given formula, i.e., an autarky that satisfies as many clauses as possible. The tool should accept a DIMACS file path as input and, as soon as possible, output a set of literals representing an autarky. Then, it should output improved sets iteratively. For each autarky, use a single line that begins with "v " and ends with a zero, with space-separated literals of the autarky in between. This format is similar to the SAT Competition Solution Output Format [1].

We will run your submission on DIMACS files and evaluate the quality of the solutions found within a set time limit. We will stop each run after k minutes. Then, we will check the last output line to see if it contains a valid autarky and compute the number of clauses satisfied by the autarky. The solution that finds the best autarkies on average will win the competition.

[1] **Output Format:** `https://satcompetition.github.io/2025/output.html`

2 Variable Elimination (3 Points)

For a gate encoding E with output x in a formula $F = E \cup R$, we simplified the resolvents $(E_x \cup R_x) \otimes (E_{\overline{x}} \cup R_{\overline{x}})$ by $S := (E_x \otimes R_{\overline{x}}) \cup (R_x \otimes E_{\overline{x}})$, dropping both $R_x \otimes R_{\overline{x}}$ and $E_x \otimes E_{\overline{x}}$. Show that the clauses in $R_x \otimes R_{\overline{x}}$ can be derived from S by resolution. You can assume that E encodes a binary AND gate.

3 Variable Elimination (2 × 2 Points)

Let the formula S with gate encodings E_1 and E_2 be given. Apply variable elimination for gates for variables a and r. Give the clause sets after each elimination step. Try the following two strategies.

1. Eliminate variable a first, and then r if possible.
2. Eliminate variable r first, and then a if possible.

$$S = \{\underbrace{\{\neg x, \neg y, a\}, \{x, \neg a\}, \{y, \neg a\}}_{E_1}, \underbrace{\{\neg a, r\}, \{\neg z, r\}, \{a, z, \neg r\}}_{E_2}, \{a, z, r\}, \{\neg a, \neg r\}\}$$

4 Blocked Clauses (3 × 3 Points)

If Blocked Clause Elimination (BCE) reduces a formula F to the empty formula then F is called a blocked set. Prove the following statements.

1. Any formula F can be partitioned into two blocked sets S and L such that $F = S \cup L$. Design a linear algorithm that produces L and S from F.
2. Blocked sets are not closed unter resolution. If F is a blocked set then $F \cup C_1 \otimes C_2$, where $C_1, C_2 \in F$ may not be a blocked set anymore.
3. Blocked sets are not closed unter partially assigning variables. If F is a blocked set then $F_{x=v}$ (the result of assigning v to x and subsequent simplification) may not be a blocked set anymore.